

DEVELOP 10 TIMES FASTER

21



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: **freesupport@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 2016: This publication may not be reproduced in part or in whole in any form without the express permission of PC SOFT.



## SUMMARY

### Introduction

Preliminary points .....	15
Overview of the tutorial .....	15
How to access the online help? .....	16
Legend of symbols.....	18
If you are familiar with WEBDEV 20.....	19
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
WLanguage.....	27

### Lesson 1.2. The Web development and WEBDEV

Principle of Browser/Server .....	29
How does it work? .....	29
And in WEBDEV? .....	29
Practical example.....	30
Intranet/Extranet/Internet.....	34
Principles .....	34
And in WEBDEV? (Classic/AWP, PHP, Static).....	34
Practical example.....	35
The method for creating a site.....	37

### Lesson 1.3. My first pages

Overview .....	39
Opening the project.....	39
Creating a "form" page.....	40
Creating the page .....	40
Creating the controls.....	41
Page improvements .....	45
Modifying the style of the "Login" and "Password" controls .....	47
Test of the page .....	49
Saving the data in a data file .....	50
Creating the data file.....	50
Saving the data.....	52
Test of the page .....	54
Viewing the data entered .....	54

<b>Adding input checks</b> .....	<b>55</b>
Mandatory input.....	55
Checking the password.....	56
<b>Simple search in a database</b> .....	<b>58</b>
Creating the page.....	58
Creating the controls.....	58
Test of the page.....	61
<b>Conclusion</b> .....	<b>62</b>

## PART 2 - MY FIRST DATABASE

---

### Lesson 2.1. Overview

Overview of the 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 the items .....	77
Importing a CSV file .....	82
Direct import of existing data files.....	85
Creating the links.....	86
Generating the analysis .....	92

## PART 3 - INTRANET SITE WITH DATA

---

### Lesson 3.1. Overview

Overview of the 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 template.....	105
Creating the controls.....	107
Test of the page .....	113
Modifying the products via a dynamic "Product form" page .....	114
Creating the page .....	114
What should this page do? .....	114
Creating the edit controls.....	115
Displaying data in the page.....	118
Creating the 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 addition of a product .....	129
Viewing the records .....	131

## PART 4 - INTERNET SITE WITH DATA

### Lesson 4.1. Overview

Overview of the 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 the controls.....	153
Displaying data in the page.....	157
Displaying the form from the list of products.....	158
Managing the styles .....	159
Importing a CSS style sheet.....	160
Apply the CSS styles .....	161



Linking the Internet site and the Intranet site.....	163
Creating a page used to display the general sales conditions.....	166
Creating the page.....	166
Creating the control.....	166
Conclusion.....	168

## PART 5 - LET'S CONTINUE THE DEVELOPMENT

---

### Lesson 5.1. Multicriteria search

Overview.....	172
Creating the query used to find the orders.....	173
Creating the query.....	173
Test of the query.....	177
Using parameters in the query.....	178
Test of the 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 (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
Test of the print.....	217

### Lesson 5.3. Sending an email

Overview.....	220
A popup page to send emails.....	221
Creating the popup page.....	221
Creating the controls used to enter the characteristics of the email.....	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**

<b>Overview .....</b>	<b>229</b>
<b>Integrating the user groupware .....</b>	<b>230</b>
<b>Configuring the user groupware .....</b>	<b>231</b>
Including the customer area in the site.....	231
Test of the site .....	233
Creating the users and the groups .....	234
Defining the rights .....	236
Test of the site .....	238
<b>Disabling the management of user groupware .....</b>	<b>239</b>

**Lesson 5.5. Managing the multilingual feature**

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

**Lesson 5.6. The referencing**

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

## PART 6 - SITE DEPLOYMENT

---

### 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 .....	265
Creating a deployment account.....	267
The different deployment modes.....	269
A detailed example: deployment by FTP.....	269
Correspondence between the directories of the development computer and the deployment .....	275

### Lesson 6.2. Managing a site

Overview .....	277
Local WEBDEV administrator .....	277
Remote WEBDEV administrator.....	279

## PART 7 - SPECIFIC WEB FEATURES

---

### Lesson 7.1. Styles

Overview .....	284
Discovering the 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 transaction and payment**

<b>Securing the information and the pages via TSL/SSL</b> .....	<b>297</b>
Overview .....	297
Implementing secure transactions via the TLS/SSL protocol.....	297
Transactions secured by TLS\SSL in a WEBDEV site .....	297
<b>Secure payment</b> .....	<b>297</b>
Overview .....	297
System for secure payment in a WEBDEV site .....	299

**Lesson 7.4. Anchoring and zoning**

<b>Overview</b> .....	<b>302</b>
<b>Creating a page in zoning mode</b> .....	<b>302</b>
<b>Implementing the anchoring</b> .....	<b>305</b>

**PART 8 - USEFUL FEATURES OF WEBDEV****Lesson 8.1. The internal components**

<b>Overview</b> .....	<b>310</b>
<b>Step by step</b> .....	<b>310</b>
Step 1: Creating an internal component .....	310
Step 2: Using the internal component.....	314

**Lesson 8.2. Automatic management of errors**

<b>Overview</b> .....	<b>318</b>
Operating mode.....	318
Implementation.....	318
Types of affected errors .....	319
<b>Automatic management of errors: a training example</b> .....	<b>319</b>

**Lesson 8.3. Import/Export**

<b>Importing elements</b> .....	<b>324</b>
<b>Exporting elements</b> .....	<b>325</b>
<b>Specific import operations</b> .....	<b>326</b>
Importing a WINDEV project .....	326
Importing an HTML page.....	327

## PART 9 - OPTIMIZING AND DEBUGGING A PROJECT

---

### Lesson 9.1. Overview

Overview .....	332
----------------	-----

### Lesson 9.2. Project audits

What is an audit? .....	334
Static audit .....	334
Procedure not run.....	336
Orphan element .....	337
Query to optimize .....	338
Cleaning the project .....	339
Dynamic audit.....	339

### Lesson 9.3. Performance profiler

Overview .....	343
Starting the performance profiler .....	343
Analyzing the result .....	344

### Lesson 9.4. Debugging a project

Overview .....	350
Using the debugger.....	350

## PART 10 - MANAGING A HFSQL CLIENT/SERVER DATABASE

---

### Lesson 10.1. Introduction

Overview .....	358
Why switch a site to HFSQL Client/Server mode? .....	359

### Lesson 10.2. Implementation of a Client/Server database

Overview .....	361
Installing a local HFSQL server .....	361
Creating a site that is using a HFSQL Client/Server database .....	362
Adapting a site to use a HFSQL Client/Server database .....	362
Overview .....	362
Adapting the example .....	362
Features available in HFSQL Client/Server mode .....	366

### Lesson 10.3. Managing a Client/Server database

Overview .....	368
Configuring the computers .....	368
The HFSQL Control Center .....	368
Creating a user account in the HFSQL Control Center .....	370
Saving the database.....	375
Conclusion.....	375

## PART 11 - SOURCE CODE MANAGER

---

### Lesson 11.1. SCM

Introduction.....	380
SCM (Source Code Manager).....	380
Principle of SCM .....	380
Creating the SCM database.....	381

### Lesson 11.2. Including a project in the SCM

Including a project in the SCM.....	383
Adding the project into the SCM .....	383
Opening a project from the SCM.....	387
Configuring the SCM.....	388

### Lesson 11.3. Using a project in the SCM

Overview .....	390
Modifying a project parameter .....	390
Modifying a project page.....	392
Modifying the checked-out element .....	393
Checking the checked-out element back in .....	394
Synchronizing the project .....	395
Off-line mode (or mobile mode).....	396
SCM administrator .....	397
Disconnecting from the SCM.....	397

## PART 12 - APPENDICES

---

### Appendice 1. Vocabulary

Main terms used .....	402
-----------------------	-----

### Appendice 2. Programming concepts

Introduction.....	411
Declaring the different types of variables .....	411
The different types of variables .....	411
Declaring the variables and their scope.....	411
Main statements of WLanguage.....	413
Conditional statements (server code and browser code) .....	413
Loop statements (server code and browser code) .....	414
Comments .....	416
Main WLanguage operators .....	416
The logical operators.....	416
The comparison operators.....	417
The indirection operators.....	417



<b>Procedures and functions.....</b>	<b>418</b>
Definition .....	418
Local procedure.....	418
Global procedure and set of procedures.....	419
How to decide whether a procedure is global or local? .....	420
About passing parameters .....	420
Calling a procedure .....	420
Creating a procedure.....	420
Practical example.....	421
A practical exercise? It's your turn! .....	421
<b>Processing strings (server code and browser code) .....</b>	<b>422</b>
Practical example.....	422
Main operations performed on a character string.....	422
A practical exercise ? It's your turn ! .....	424
<b>Processing numeric values (server code and browser code).....</b>	<b>425</b>
Practical example.....	425
Main operations performed on the numeric values .....	425
A practical exercise? It's your turn! .....	426
<b>Processing currencies (server code only) .....</b>	<b>427</b>
Practical example.....	427
Details.....	427
Mixing strings and numeric values .....	428
<b>Processing dates and times (server code and browser code) .....</b>	<b>429</b>
<b>The dates (server code and browser code).....</b>	<b>429</b>
Practical example.....	429
Input mask and returned value .....	429
What is today's date? .....	431
What is today's date, with the day and the month in letters?.....	432
How many days between 2 dates? .....	432
What is the day of the week corresponding to a given date? .....	432
Practical exercise .....	433
Handling Date variables.....	433
A practical exercise? It's your turn! .....	434
<b>The times (server code and browser code).....</b>	<b>434</b>
Practical example.....	434
What time is it?.....	435
How much time has passed between two given times? .....	435
A practical exercise? It's your turn! .....	436
<b>Calculations with dates and times.....</b>	<b>436</b>

### Appendice 3. Code editor: Questions/Answers

---

<b>Questions/Answers.....</b>	<b>438</b>
How to view the element to which the current process belongs ? .....	438
How to print the source code ? .....	438
How to perform a "find and/or replace" ? .....	438
What is the meaning of the "+" and "-" signs in the code editor ? .....	438
Is it possible to identify the person who wrote a code line ? .....	439
Is it possible to find out the number of a code line ? .....	439
Is there a method to easily display the syntax or the help about a function ? .....	440
What are the useful shortcuts in the code editor ? .....	440

### 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 features of WEBDEV.

You should plan on spending a few hours to follow this course and to learn WEBDEV: you'll find it well worth it!

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 to be followed 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 through 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 the tutorial

---

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

- 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 !**



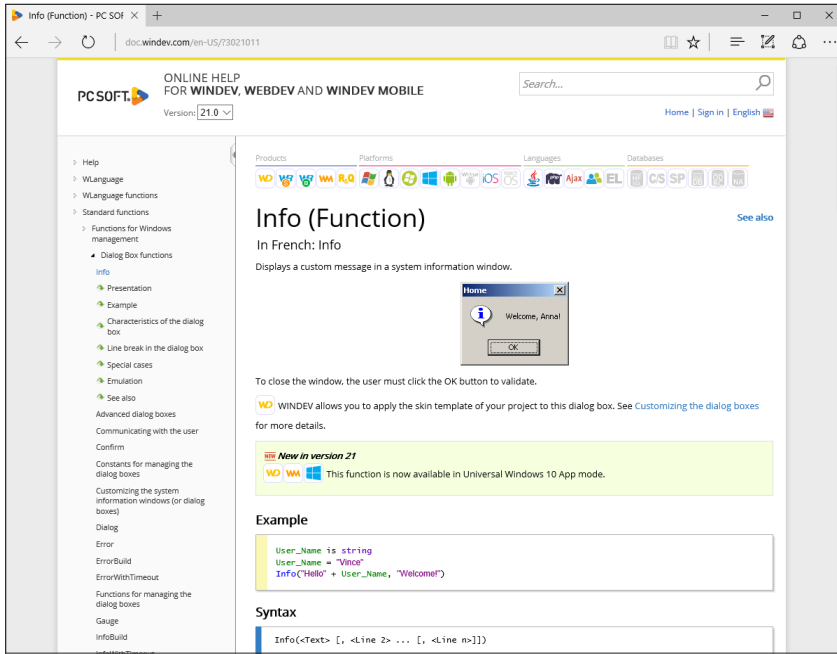
Notes

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

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



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

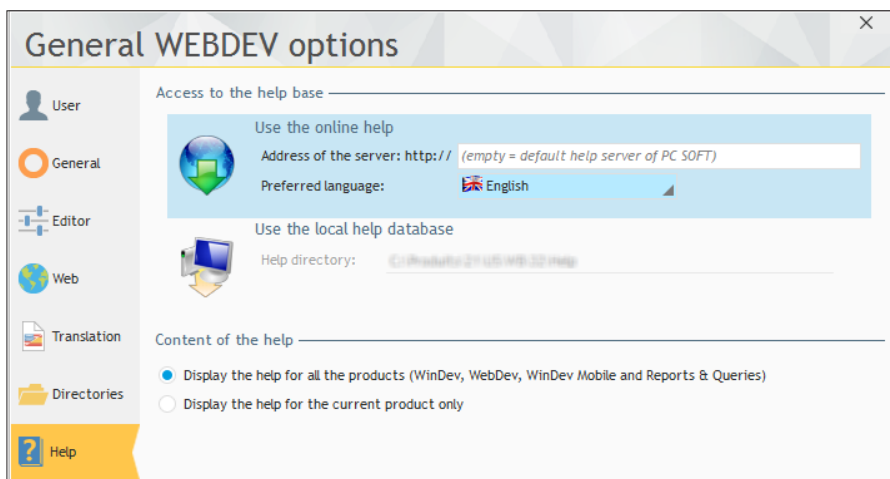


### Notes

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

**Legend of symbols**

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": we strongly advise you to read the associated text.



This symbol introduces a "Warning": reading the associated text is extremely important.



This symbol introduces a "Note": we recommend that you read the associated text.



This symbol presents a feature specific to Internet: we strongly advise you to read the associated text.

## If you are familiar with WEBDEV 20...

---

If you are familiar with WEBDEV 20, following this tutorial will do no harm: it's a good opportunity to "review" the features of WEBDEV!

## 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 deploy 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!



DEVELOP 10 TIMES FASTER

## PART 1

# Discovering WEBDEV

# 21







# LESSON 1.1. DISCOVER WEBDEV

**This lesson will teach you the following concepts...**

---

- Starting WEBDEV



Estimated time: 5 min

## 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 21 (if not already done).
- ▶ **If WEBDEV 21 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 21 was already started**, identify yourself if necessary. The development environment is started. 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.

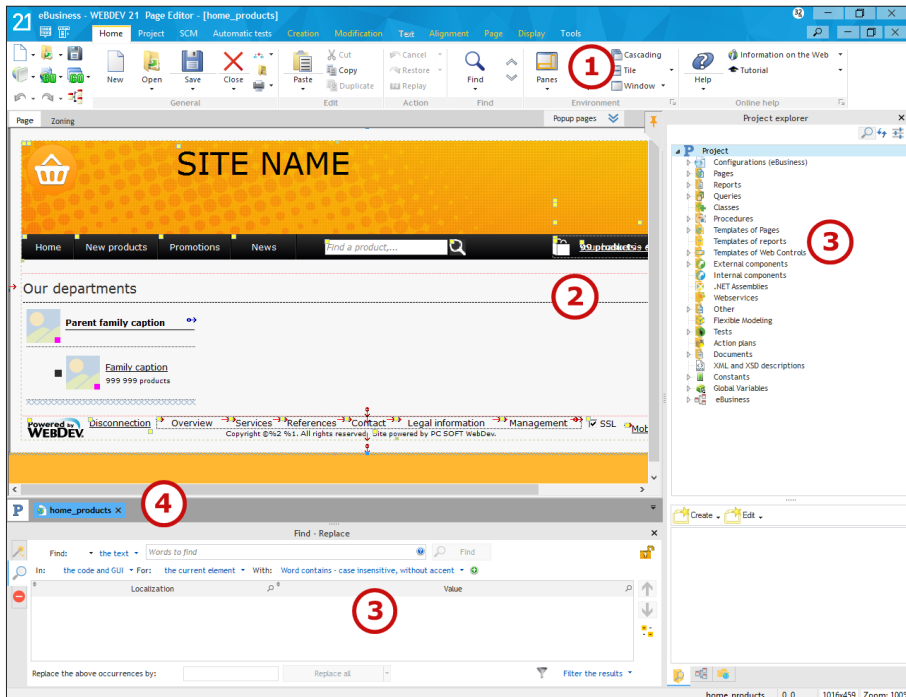
Let's take a look at the development environment of WEBDEV.

## 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.

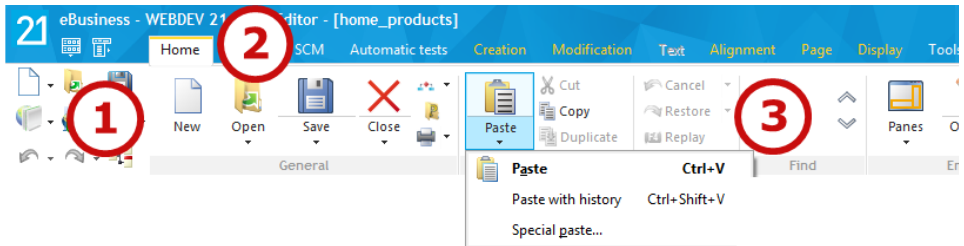
These panes can be hidden by pressing [CTRL] + [W] if necessary.

**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 3 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



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 the editors (version 17 and earlier).
- The 2 other logos are used to restore the toolbars and the drop-down menus found in the former interface of the editors (version 17 and earlier).

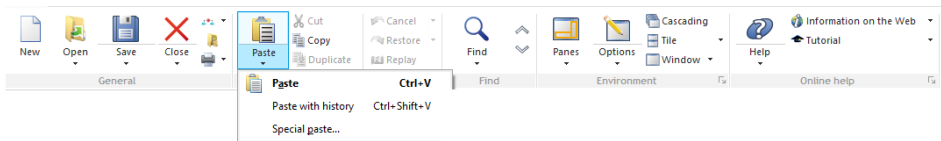
## The pane area



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

## The option area



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: displaying the description of the current element, displaying 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".

## 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 "Programming concepts" lesson, page 410. This lesson presents the different types of variables, the main statements of WLanguage, the processes of strings, numerics, dates, times, ...

# 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 min

## 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, there will be two types of processes:

- Processes run at browser level, on the computer of the Web user.
- Processes run at server level.

The code run at browser level is called JavaScript code. The browsers can only 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 run in WLanguage.

The browser code is converted into JavaScript.

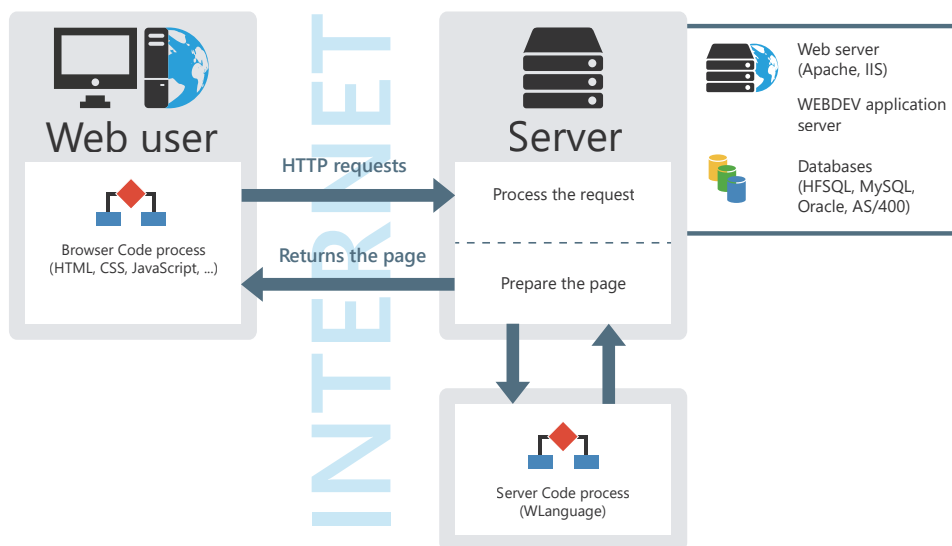
To create a site with WEBDEV, a single language is required: WLanguage.

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 on the browser directly, without having to go back to 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 21 (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 WD210Admin.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 the 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 the 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 values of controls 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.

```

WL Click (onclick) of BTN_Subscribe (browser)
IF 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)
  END
IF 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)
  END
IF 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)
  END
IF 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)
  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 If Error: by program When Exception: by program
IF // Reset the customer structure

```



## Notes

The code editor allows you to write the code of 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:

- Initialization.
- Browser click.
- Server click.

Let's see 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 the Web user, is displayed in green.
- The server, that will be run on the server, is displayed in yellow.

In our example, all the input checks are performed in the green code (browser 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
// 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)
END
```

This check is performed in browser code because there is no need to go back to the server to check that 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 entered 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 directly
ScreenToFile()

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

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

- ▶ Close the code window (click the cross in the top right corner).
- ▶ Close the page displayed in the editor (click the x 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 intended for "public" consumption (standard users or business users).

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

To bring 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 directly accessible at any time. But this constraints 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. Therefore, the address of pages depends on this identifier that changes at 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 mode.
- The static mode if your site contains preset pages only (no database).

## Practical example

► First, let's take a look at how 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.

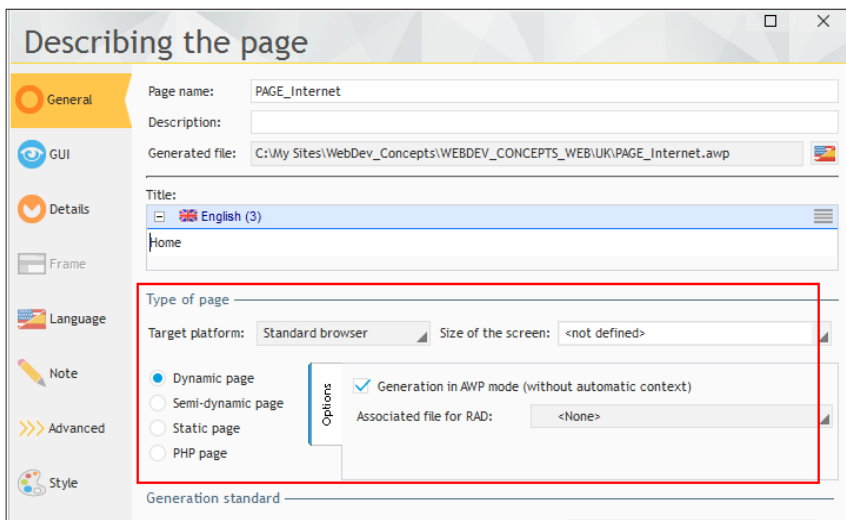



Tip

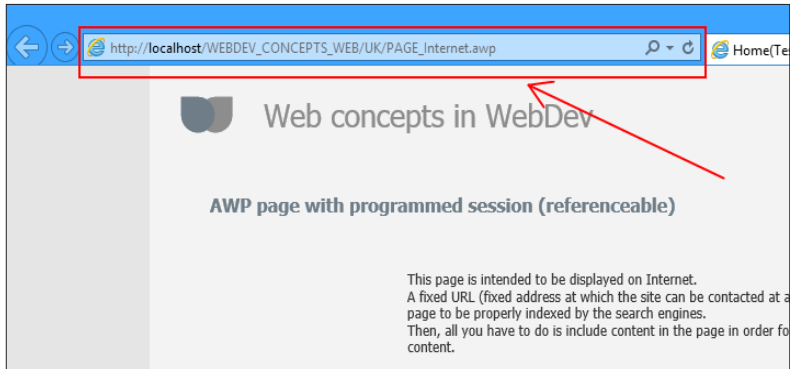
To quickly find a page in the current project, press [CTRL] + [E]. A window is displayed, allowing you to perform a search on all the pages containing the set of letters typed in the search area. All you have to do is select the requested page and validate in order for this page to be opened in the editor.

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 a generation in AWP mode.




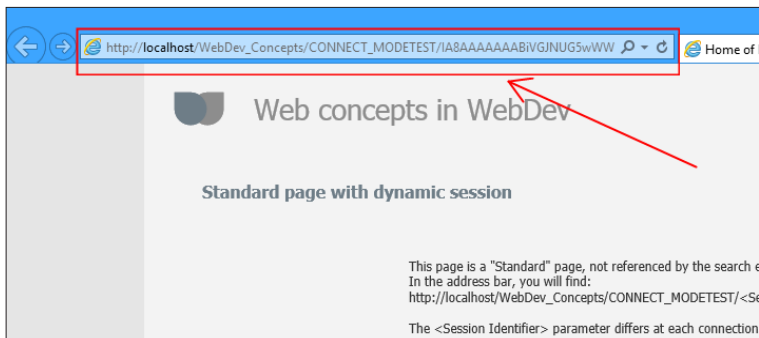
- Validate the description window of the page.
3. Run the test of this page: click  among the quick access buttons.
  4. The page is displayed in your default browser.
  5. See the address displayed in the browser: this page has a fixed address. If you copy this address and if you paste it in a new browser,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



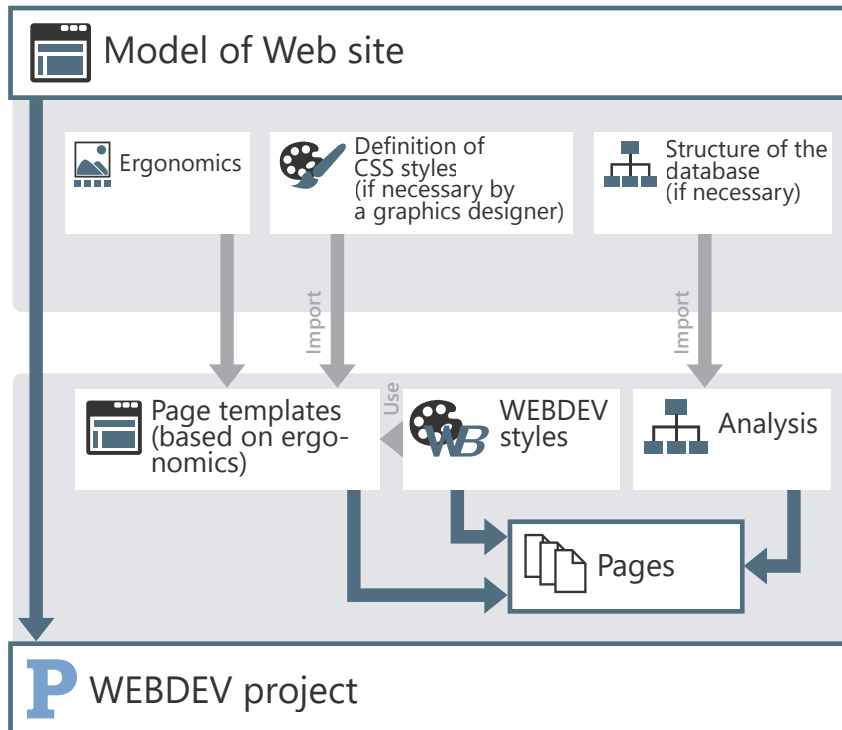
6. Close the browser.

## 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 structure of the database
- Development in WEBDEV:
  - Creating a project that groups all the site elements.
  - Creating the page templates that will define the look of the site. These templates are based on the model of pages, the CSS styles and WEBDEV.
  - Defining the database (Analysis), 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 the data
- Checking the input
- Simple search in a database



Estimated time: 30 min



## Overview

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



Notes

The pages are used to display or enter 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 21 (if not already done). Close (if necessary) the current project in order 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)".

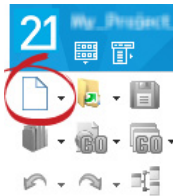
## Creating a "form" page

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.




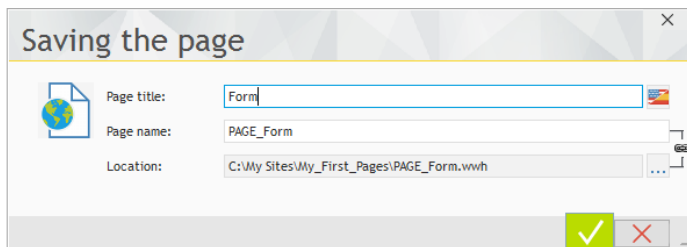
2. The window for creating a new element is displayed. This window is used to create all the elements that can be associated with a project.

3. Click "Page" then "Page". The wizard for page creation starts.

4. Select "Blank - Simple layout" and validate (green button at the bottom of the window). The page is automatically created in the editor.

► To save the page that was just created:

1. Click  among the quick access buttons (you can also use the CTRL S shortcut).
2. The backup window is displayed.
3. Enter the title of the page: "Form".



4. The name of the page (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 name of the page.

5. Validate (green button).



Notes

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

- A "WWH" file that contains the description of the page for the page editor. This file is saved in the project directory.
- An "AWL" file that corresponds to the description of the page for the WEBDEV engine in the EXE directory of the 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 the project directory (one sub-directory per language, for example "FR" for French, "UK" for English, ...).

## Creating the controls


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

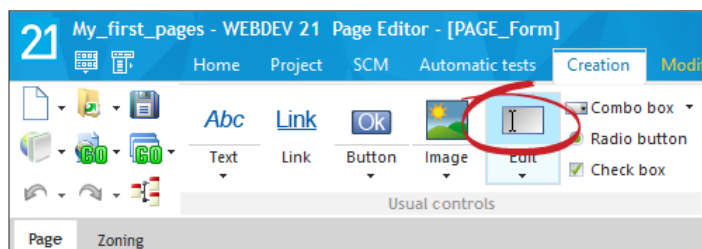
The form includes the following controls:

- Name
- First name
- Address
- Email
- Login
- Password

All these controls are edit controls, allowing the user to enter data.

► To create an edit control:

1. On the "Creation" pane, in the "Usual controls" group, click  (above "Edit").



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



Tip

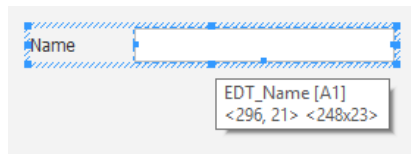
You must create a control? Display the "Creation" pane of WEBDEV: all the available controls are accessible in this pane.

► To modify the caption of the control:

1. Click the control (simple click): the caption becomes editable. If you prefer to use the keyboard, all you have to do is press the [SPACE] key or the [ENTER] key to switch the caption to edit.



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



#### Notes

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

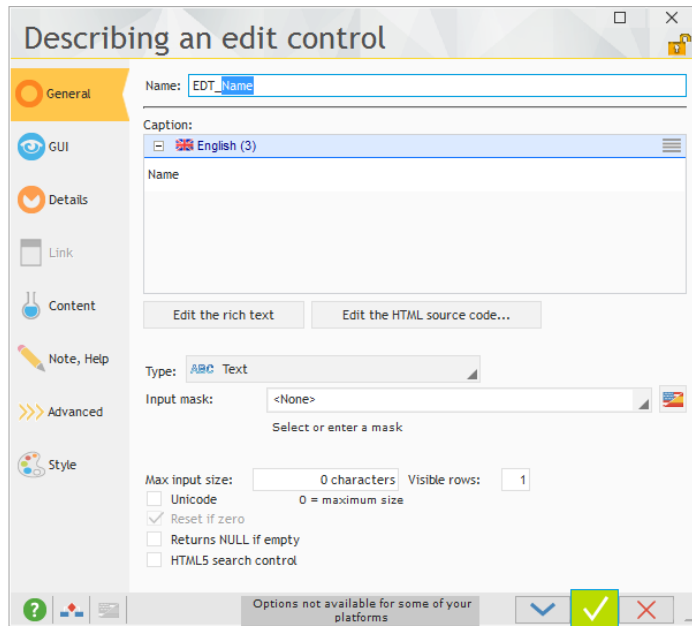
The programming charter is used to define a prefix for each type of object, allowing you to quickly identify the element:

- 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.

If you do not want to use this charter, all you have to do is disable it: in the ribbon, on the "Project" pane, in the "Other actions" group, expand "Charter" and uncheck "Use the charter".

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

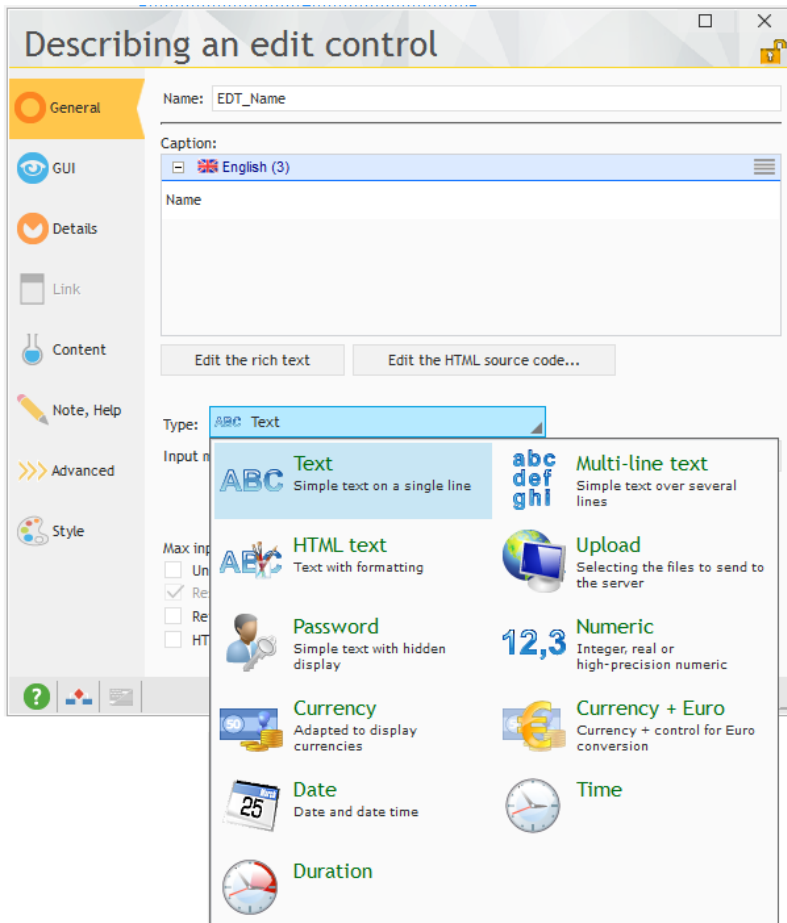
- To modify the detailed characteristics of the control, all you have to do is double-click the control. A description window of the control is displayed:



- 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:

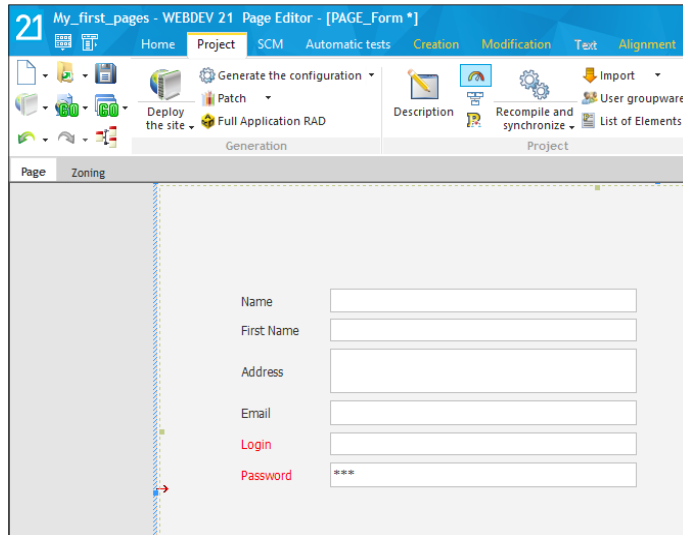
Caption	Type	Size
First name	Text	30
Address	Multi-line text	
Email	Text	255
Login	Text	10
Password	Password	10

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 the control.



The basic form is created. We are going to 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, let's take a look at the "Address" control. This control must display several lines on the screen. We are going to enlarge it.

Then, we are going to align the controls in 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 appear.
  3. With the mouse, grab the handle at the bottom of the input area and enlarge the area.
- ▶ This allows you to increase the width of the different page controls.

- 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)

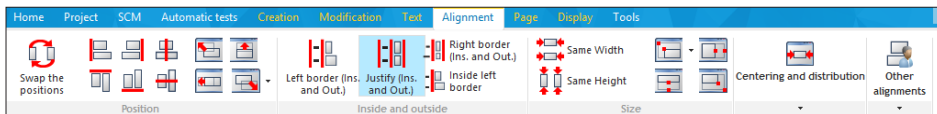


## Notes

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

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

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



3. We want the left and right sides of the controls to be aligned. Click "Justified (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 in the page: the controls are moved back to their initial position.

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

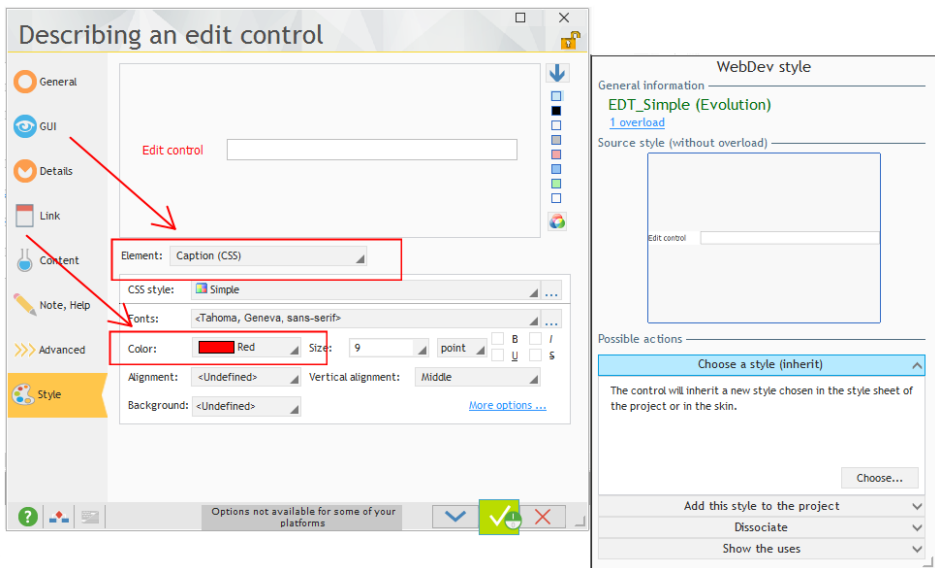


## Modifying the style of the "Login" and "Password" controls

To display the caption of the "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 the 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.





Notes

This style can be added to the style sheet of the project in order to be re-used. To do so, you must:

1. Click the "Add this style to the project" button in the "WEBDEV style" pane displayed on the right of the 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 the control (right mouse click) and select "Choose the WEBDEV style".
3. In the window that is displayed, select the requested style and validate.

Note: The choice of an existing style will be performed in lesson 5, page 188.

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

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

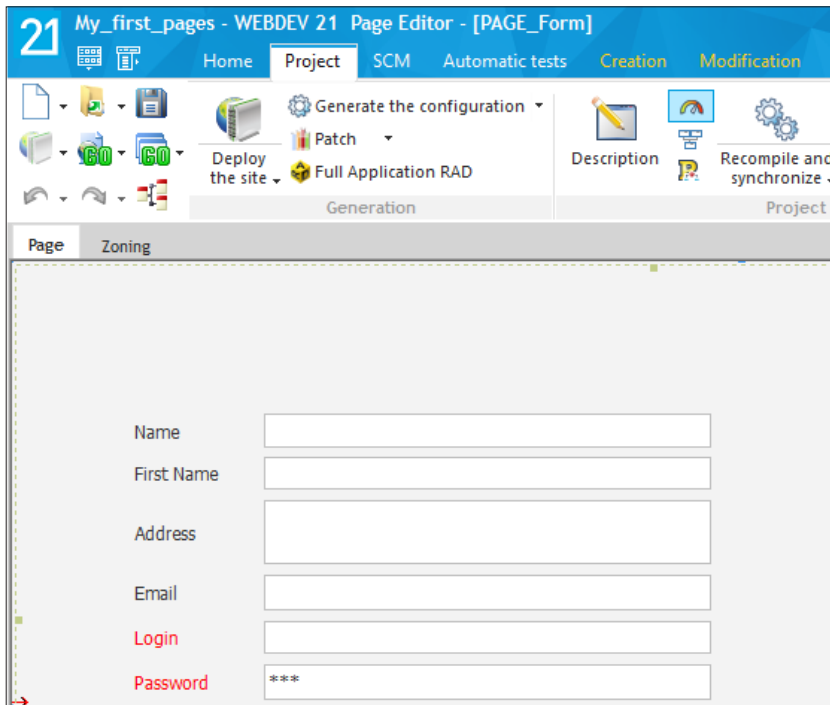
1. Select the "Password" control.
2. Press the [F4] key.




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:



### Test of the page

- ▶ Run the test of this page (click  among the quick access buttons). You have the ability to enter data but this data will be neither processed, nor saved.
- ▶ 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 entered. 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 description of the database is performed in a specific editor, the data model editor. This editor will be presented in details 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 the data files that will be used by the project.
  3. The wizard for generating a data file is started.

Generating a file description

Describing the data file

This wizard helps you create a description of data file based on the controls found in the window or in the page.

Specify a name for the new data file:

Name:

Caption:

A record represents:

< ✓ >

4. The name of the data file is automatically filled with the name of the page. We are now going to modify some elements:
  - Type "Registration" in the name.
  - Type "Registration" in the caption.
  - Type "a registration" in the last control.

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

The screenshot shows a wizard window titled "Generating a file description". The current step is "Selecting the items", with the instruction: "In the list below, choose the controls that must correspond to items." Below this is a table with three columns: "Name of the control", "Item name", and "Caption of the control". The table contains five rows, each with a checked checkbox in the first column. At the bottom right of the table are two buttons: "All" and "None". At the bottom of the wizard are three navigation buttons: a left arrow, a checkmark, and a right arrow.

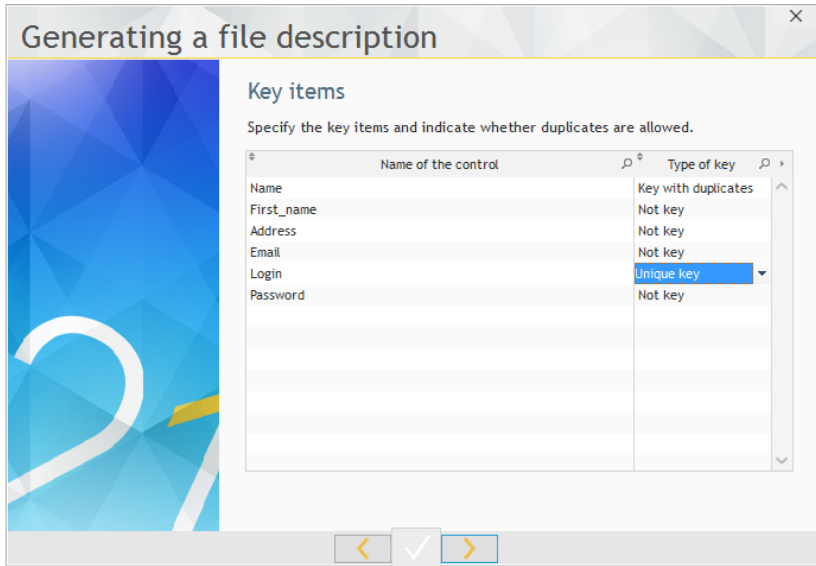
<input type="checkbox"/>	Name of the control	Item name	Caption of the control
<input checked="" type="checkbox"/>	EDT_Name	Name	Name
<input checked="" type="checkbox"/>	EDT_First_name	First_name	First name
<input checked="" type="checkbox"/>	EDT_Address	Address	Address
<input checked="" type="checkbox"/>	EDT_Email	Email	Email
<input checked="" type="checkbox"/>	EDT_Login	Login	Login
<input checked="" type="checkbox"/>	EDT_Password	Password	Password

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

**Notes**

A key provides faster access during the searches (key with duplicates) and/or insures 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. Therefore, the file can contain two people with the same name but not with the same login.



10. Go to the next step.

11. Validate the summary. The file is displayed in the data model editor.

12. Save the analysis: click  among the quick access buttons (or use the CTRL S shortcut).

13. Close the data model editor.

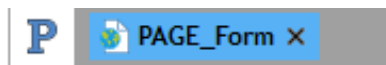
14. A window is displayed, proposing to synchronize the project. It is used to check the differences between the created pages and the data defined in the analysis. This step is performed whenever the analysis is modified, when going back to the project pages.

15. Click "Yes".

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

## Saving the data

- Let's go back to the registration form in order to add the data entered by the Web user into the data file. The document bar is found at the bottom of the 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 entered in the page and to save the data in the Registration file.
- ▶ To create a Button control:
  1. On the "Creation" pane, click **Ok** above "Button".
  2. The control currently created follows the movement of the mouse.
  3. Click in 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 enter the code associated with the button:
  1. Display the popup menu of the control and select "Code" (or press the [F2] key).
  2. The processes associated with the button are displayed. The yellow codes correspond to the processes performed on the server and the green codes correspond to the processes performed on the browser.
  3. We want to save the data in the database. The access to the database can be performed in Server code only: therefore, the code must be entered in the Server click code (yellow).
  4. Enter the following code:

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

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

## Test of the page

► Run the test of this page:

1. Click  among the quick access buttons.
2. Enter the following data:
  - Name: Doe
  - First name: John
  - Address: 5th avenue, New York
  - Email: John.doe@atandt.com
  - Login: john
  - Password: John
3. Click the "Save" button. The browser displays a message indicating that the record is added!
4. Close the browser.


## Viewing the data entered

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

This tool is named WDMAP. We are going to use it in order to check whether the entered 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 content of the data file 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 find the data entered.
4. Close WDMAP ("Close" button).




## Adding input checks

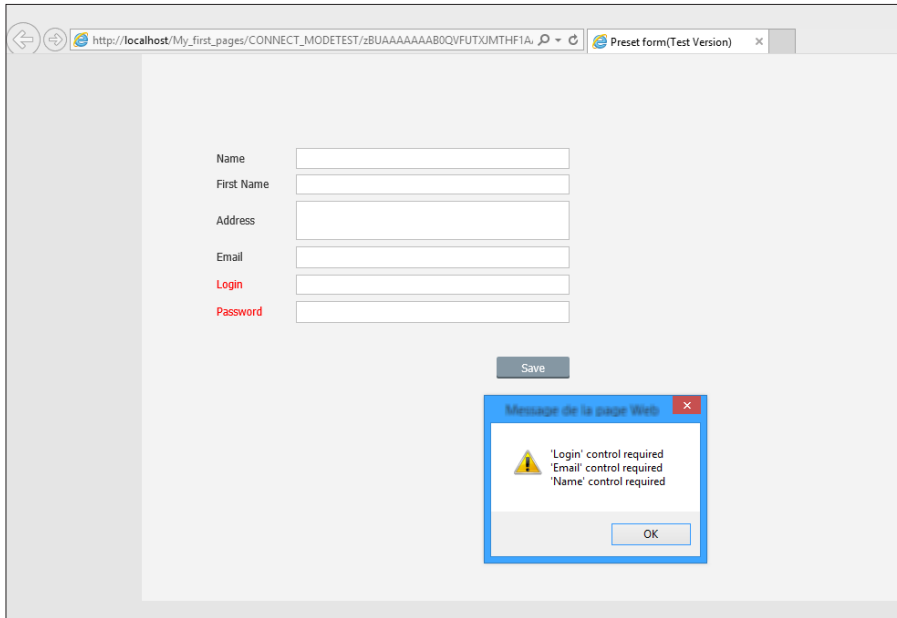
---

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

- force the Web user to enter the name, email and login.
- force the password to be typed twice in order 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 the control is displayed.
  2. Display the "Details" pane.
  3. Check "Mandatory input". If this option is checked, WEBDEV will automatically check that 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.
- ▶ Apply this modification to the "Login" control.
- ▶ Run the test of this page:
  1. Click  among the quick access buttons.
  2. 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 are going to enter the code required for the check.

- Create a new edit control: you can for example copy/paste the "Password" control found in the page. The caption of this control is "Check" and its type is "Password".
- The code for checking the password must be entered 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 enter 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 entered passwords 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. Enter the following information:
  - Name: "Doe"
  - First name: "Alan"
  - Email: "Alan.Doe@gmail.com"
  - Login: "Alan"
  - Password: "Alan"
  - Check: "Luis"
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 entered. Then, we will check whether the information entered 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" and validate.
- ▶ To save the page that was just created:
  1. Click  among the quick access buttons (you can also use the CTRL S shortcut).
  2. The backup window is displayed.
  3. Enter the title of the page: "Login".
  4. Validate.

### Creating the controls

The login page will contain the following controls:

- Two edit controls:
  - "Login" to enter the login.
  - "Password" to enter the password.
- Two buttons:
  - "Connect" to check the login and the 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 steps to perform:

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



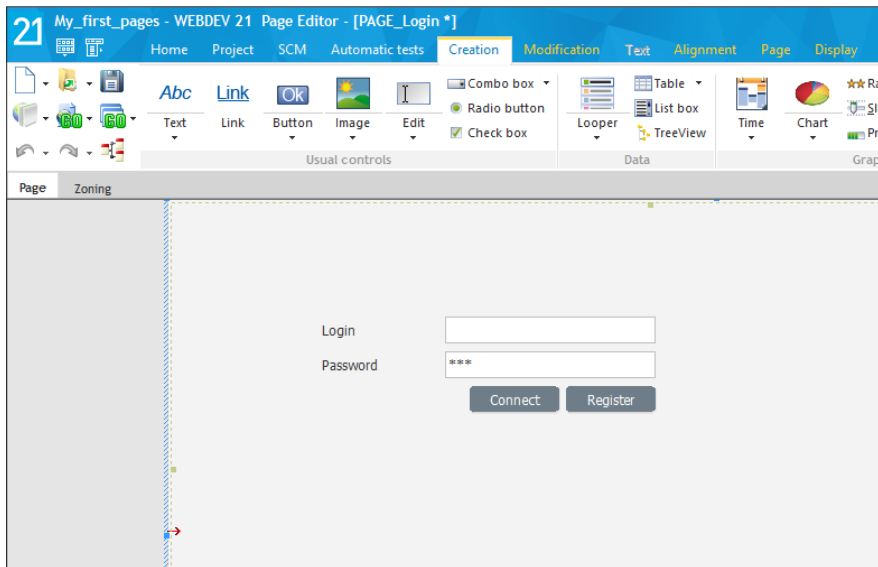
Notes

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

► To create a Button control:

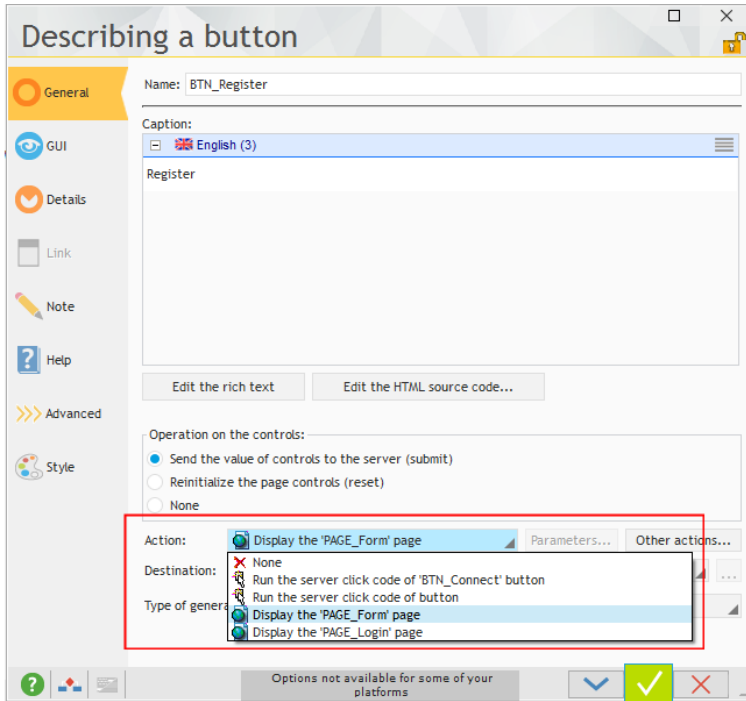
1. On the "Creation" pane, in the "Usual controls" group, click **Ok** above "Button".
2. Click in the page to create the button (below the edit controls for example). The control is created.
3. Press the space bar: the caption becomes editable.
4. Enter the caption ("Connect" and "Register") and press 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. Display 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 directly associated with the action to perform (opening a page here).

- For the "Connect" button, we must enter the WLanguage code required to perform the search:
  1. Select the "Connect" button and display the associated processes ([F2] key).
  2. Enter 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("OK, you're connected.")
    ELSE
        Error("Wrong password.")
    END
END
END
```

### 3. Let's study this code:

- This code is run in server code because we are performing a search in the database that is 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 ("Login not found" displayed) as well as the action to perform if the login is found.
- If the login is found, the password entered in the EDT\_Password control is compared to the one found in the data file (for the specified login). If the two passwords are identical, the Web 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 (x at the top right).

## Test of the page

### ► Run the test of this page:

1. Click  among the quick access buttons.
2. Enter the following data:
  - Login: john
  - Password: john1
3. Click the "Connect" button. An error message is displayed.
4. Validate the message and modify the password: "john"
5. Click the "Connect" button. You are connected.
6. Close the browser.

## Conclusion

---

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

- 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 the database characteristics before explaining how to develop a full site.



DEVELOP 10 TIMES FASTER

## PART 2

My first  
database

# 21





## LESSON 2.1. OVERVIEW

**This lesson will teach you the following concepts...**

---

- Overview of the project developed in this part



Estimated time: 5 min

## Overview of the project developed in this part

---

We are going to create a project associated with a HFSQL database.

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

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

In a next section, we will concentrate on the development of the 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 min

## 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, an "analysis" must be created beforehand. An "analysis" contains the description of data files (or tables). The application data will be saved 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 mode or in 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 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 features of a database, 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 access

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

In WEBDEV, a native access 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)

Other native accesses will be available soon, contact our sales department for more details!

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

### Direct ODBC access

An access via direct ODBC is using a multi-database access standard. The 32-bit 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 via "ODBC administrator".

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 using this type of access.

The **SQL\*** and **HRead\*** functions of WLanguage 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 **SQL\*** and **HRead\*** functions of WLanguage 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 min

## 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 "Web 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 the WEBDEV menu.
2. The window for creating a new element is displayed: click "Project".

3. First, the wizard allows you to enter the name of the project, its location and its description. In our case, this project will be named "My\_WebDev\_Site"

### Wizard for project creation

1 Description

2 SCM

3 Charters

4 Internationalization

5 Database

6 End

#### Description - General information

Give a name to your project

All the elements included in your project will be saved in the project directories.

Project name:

Location:

Describe your project in a few sentences

This description will always be visible in the project dashboard.  
It can be modified at any time during the entire lifetime of the project.

Summary of your project:

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 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.

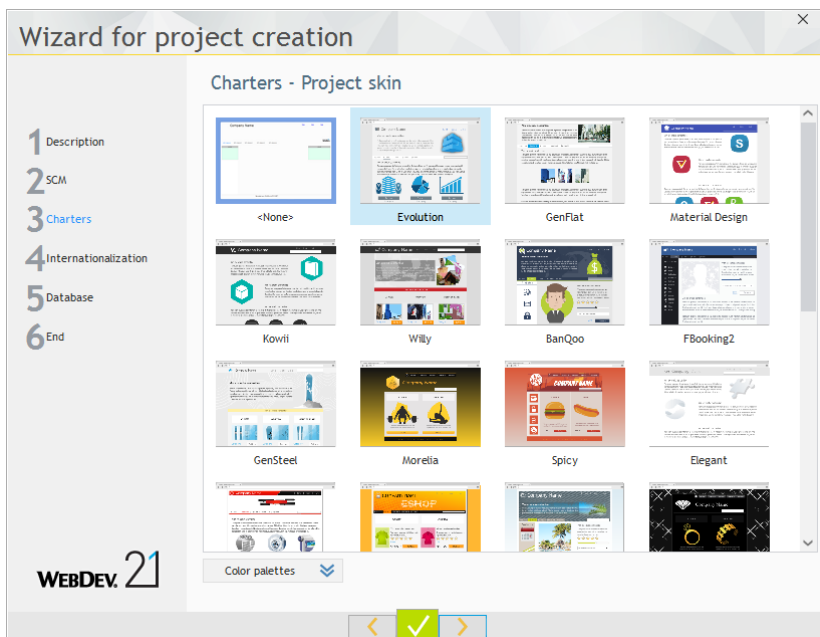
7. This step is used to define the style book (also called "Skin"). The style book is used to define the "look" of the site.



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 accessed at any time to check the validity of the 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 steps of the wizard for analysis creation are as follows:

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

The screenshot shows the 'Analysis Creation' wizard window. On the left, a sidebar lists three steps: '1 General points', '2 Database', and '3 End'. The main area is titled 'General information' and contains the following text: 'An Analysis enables you to describe the structures of the data used in your application or in your Web site.' Below this, there are two input fields. The first is labeled 'Choose the name of the analysis:' and contains the text 'My\_WebDev\_Site'. The second is labeled 'Directory:' and contains the text 'C:\My Sites\My\_WebDev\_Site\My\_WebDev\_Site.ana'. To the right of the input fields is a small icon of a folder with a document inside.

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).

The screenshot shows the 'Analysis Creation' wizard window, now on the 'Database type' step. The sidebar on the left shows the same three steps. The main area is titled 'Database type' and contains the text 'Select the type of database that will be used in the analysis:'. Below this text is a search bar. There are six database options displayed in a grid, each with an icon and a description. The first option, 'HFSQL Classic', is highlighted with a blue background and a green checkmark. The other options are 'HFSQL Client/Server', 'SQL Server', 'SQL Azure', 'Oracle', and 'MySQL'. Each option also includes a note about accessibility, such as 'Accessible via the native access'.

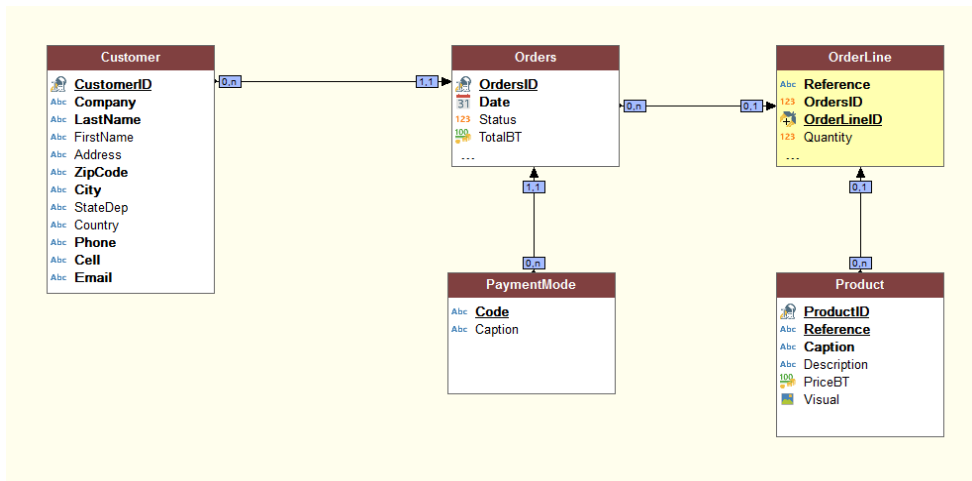
Go to the next wizard step.

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):

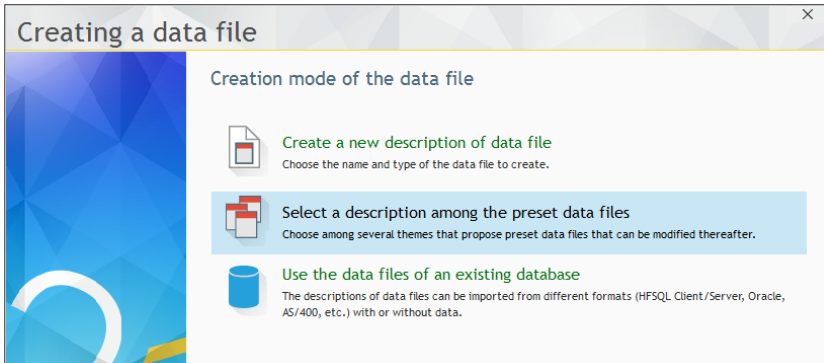
- Customer,
- Orders,
- PaymentMode,
- OrderLine,
- Product.



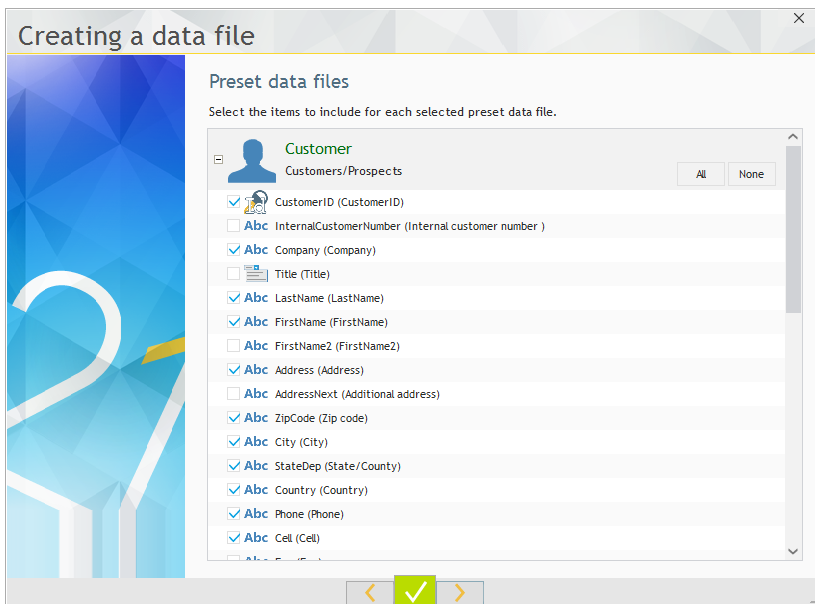
To create the data files of this analysis, we will be using the different methods proposed by WEBDEV.

## Creating a data file: using a preset file

- The steps of the wizard 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).

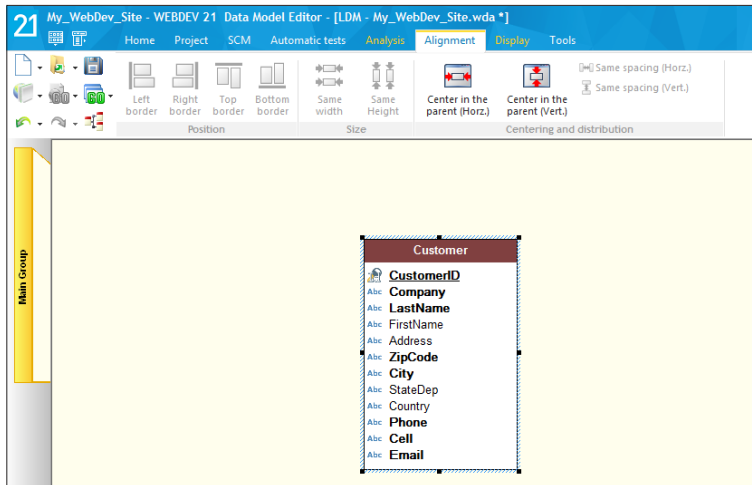


2. The list of preset data files is displayed. We are going to create the "Customer" file. In the list of data files, select "Customer". Go to the next step.
3. The wizard proposes the list of items to include in Customer file. This list is very impressive because it is used to manage several types of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

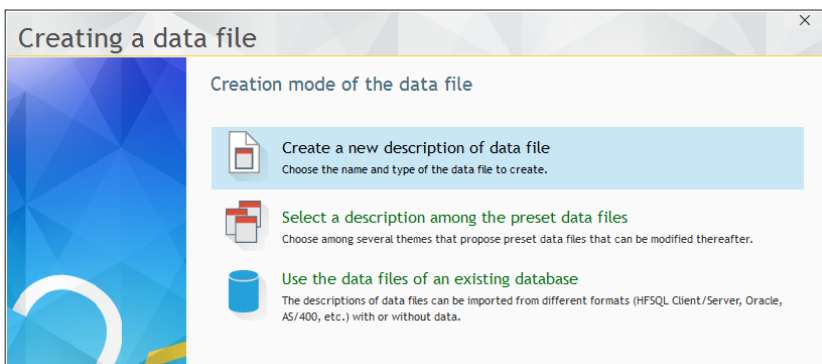
- To create a data file from the window for creating a new element:
  1. Click "Data" then "Data file".



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. Enter 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 the associated physical data file ("Orders.fic" file).

The caption and description of the 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 the links between data files. A caption is automatically proposed from the name of the file.

In our case, type "An order".

**General parameters**

Name of Data File

Name: Orders

Caption: Orders

A record represents: An order

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

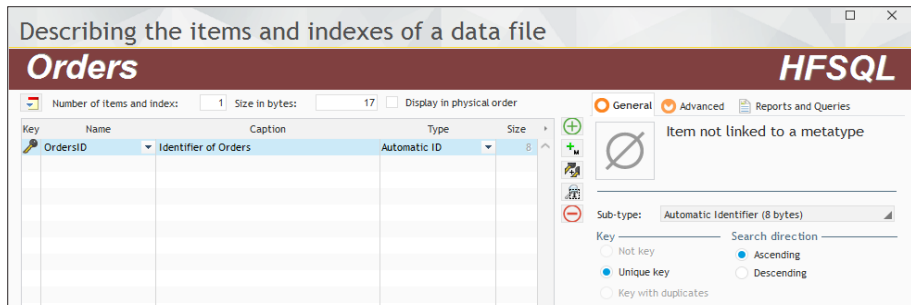
To create the identifier (an identifier is a unique key), you have the ability to create a numeric item whose type is "Automatic identifier".

This identifier is automatically managed by WEBDEV. Whenever a record is added into the data file, WEBDEV automatically assigns a value to the identifier of the file. 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.

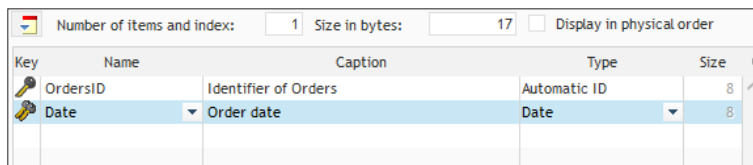




We are going to create the items of the "Orders" file. In the description window of the data file, you will notice that an item was automatically created: "OrdersID". This item corresponds to the automatic identifier of the data file. This item includes the name of the file and the letters "ID".

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 the first empty row. This column automatically becomes editable. Type "Date".
  2. Click the "Caption" column. The name of the item is automatically displayed. We are going to modify the caption of the item 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 (an index) in our data file: the keys are used to improve the speed for accessing the data and for the sorts.

- For a database in SQL format, the engine is using the indexes at the best.
- 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 characteristics of an item. 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 records found in the data file),
- key with duplicates: the value of this key can be found several times in the data file.

4. The definition of the key is performed as follows: re-select the line of the "Date" item to enable the description controls found on the right of the 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 browse is performed 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. Enter:

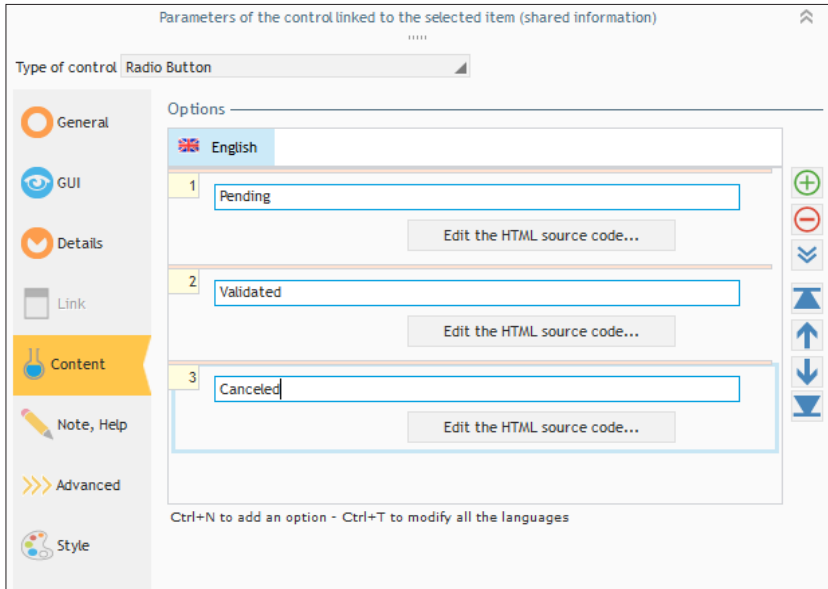
- the name: Status
- the caption: Status of the order
- the type: Radio button, List box, Combo box. In the window that is opened, you have the ability to select the type of control created by default for this item. It will be a radio button in this case. Validate the window.

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

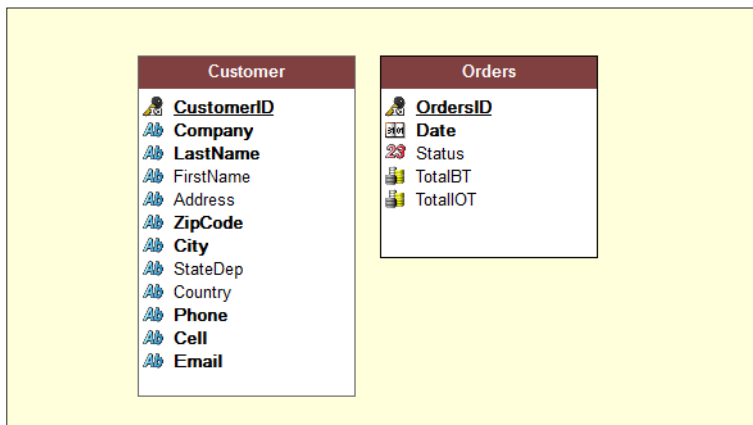
The information entered here will be automatically used when creating the pages linked to the data file. You find here the control type and the caption.

We are going to enter the different options corresponding to the status of the order 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 the 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 (a comma, a semi-colon or a tab in most cases).

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

- the description of the 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".



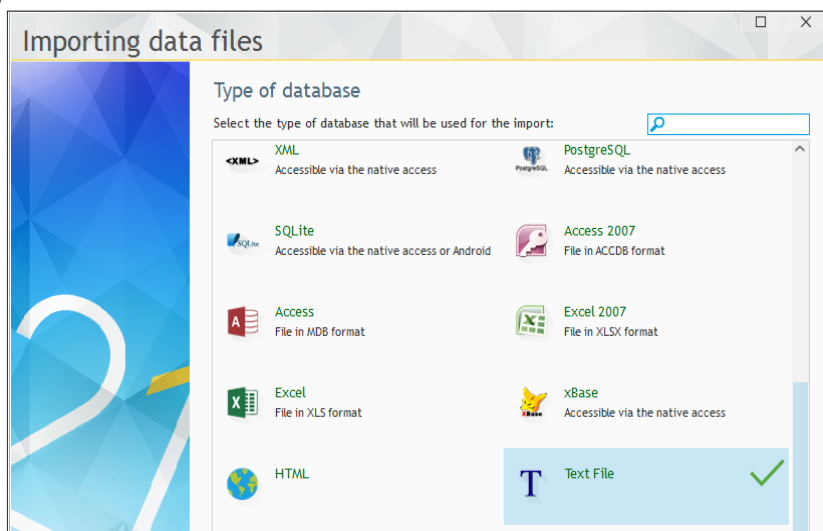
Tip

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 file 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.



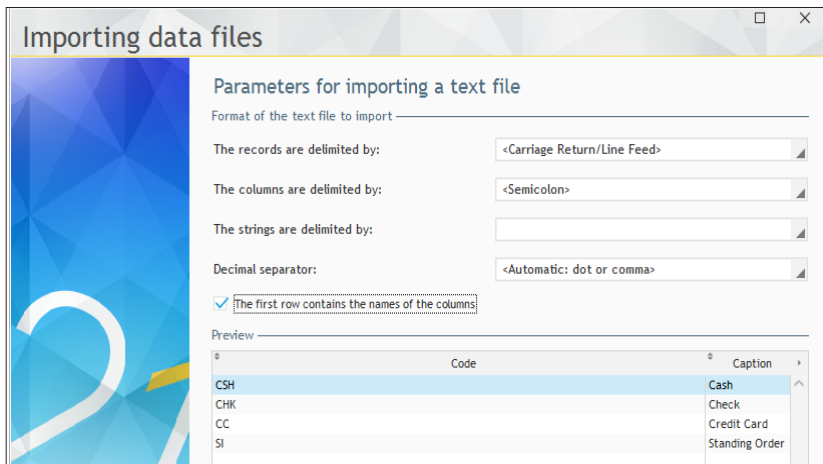
5. Specify the path of the file to import: "\\Tutorial\\Exercises\\My\_WebDev\_Site\\PaymentMode.csv" found in the setup directory of WEBDEV.

Note: By default, the file picker does not propose the csv files: specify the "\*.csv" extension to get this type of file.

6. Go to the next wizard step.

7. Specify the following import parameters:

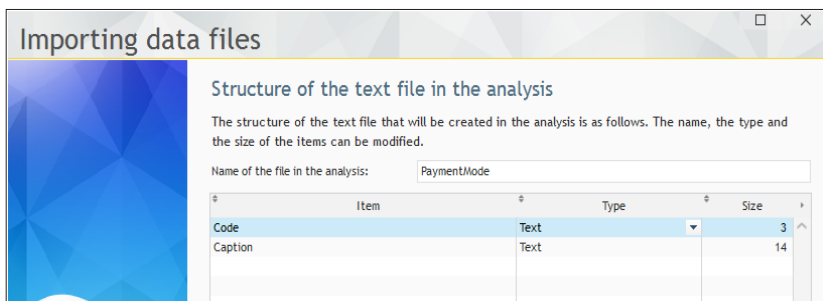
- Records delimited by: "<Carriage return/Line feed>"
- Columns delimited by: "<Semicolon>"
- Strings delimited by: "<None>"
- Decimal separator: "<Automatic: dot or comma>"



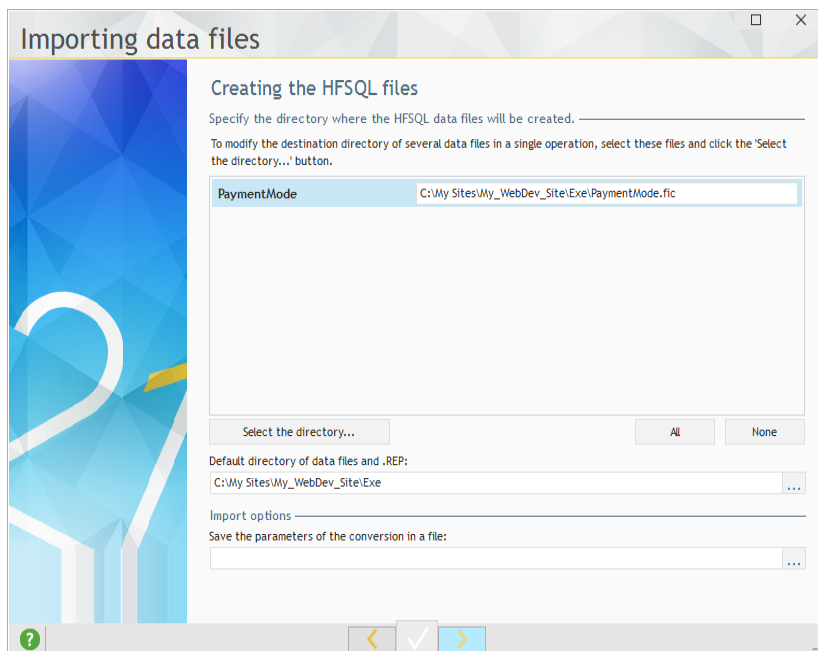
8. Don't forget to check "The first row contains the names of the columns".

9. Go to the next step.


10. The structure of the data file that will be created is displayed. Keep the default options. Go to the next step.

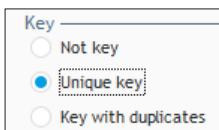


11. The content of the CSV file will be automatically converted to HFSQL format. The wizard proposes to create the HFSQL file in the project directory. Keep the proposed options and go to the next step.



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

- Let's study the description of the imported data file:
  1. Select the "PaymentMode" file then select "Description of data file" from the popup menu.
  2. In the window that is displayed, modify the caption of the file: delete "(Imported)".
  3. Click  to display the description of the 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 the screen, click "Unique key".



5. We get the following data:

PaymentMode				
Number of items and index:		2	Size in bytes:	28
		<input type="checkbox"/> Display in physical order		
Key	Name	Caption	Type	Size
	Code	Code	Text	3
	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 file explorer of Windows, open the following WEBDEV sub-directory:  
"Tutorial\Exercices\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 !

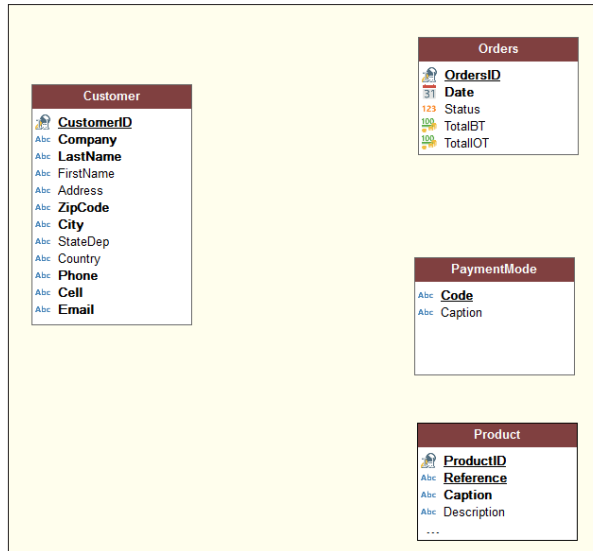
Only the description of the "Product" file was imported into the analysis of our project. The data found in the "Product" file was not imported into the project.

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\Exercices\My\_WebDev\_Site") to the EXE sub-directory of the directory of your project.

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

## Creating the links

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



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 the "Customer" file and the "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 window. It displays a diagram of a link between the "Customer" and "Orders" data files. The "Customer" file is connected to the "Orders" file with a cardinality of "0,n" on the "Customer" side and "1,1" on the "Orders" side. Below the diagram, there are four questions with radio button answers:

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

At the bottom right, there is a checkbox labeled "Display the advanced cardinalities" which is currently unchecked.

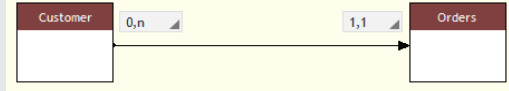


- Each Customer has at least one Order: No
- Each Customer can have several Orders: Yes
- Each Order has at least one Customer: Yes
- Each Order can have several Customers: No



Notes

You also have the ability to directly enter the cardinalities of the link in the wizard.



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

Describing a link

Unique key

Select the unique key of "Customer" file used by the link.

CustomerID

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

Describing a link

Linked key

"CustomerID" key of "Customer" file must be migrated into "Orders" file.  
Create or select the key copy in "Orders" file.

☒ Create a new key

Name: CustomerID

☐ Use one of the items below

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 the 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

**Describing a link**

Describing the link

Orders (0,n) — (0,1) Product

Each "An order" has at least one "a Product"

☐ Yes  
☒ No

Each "a Product" has at least one "An order"

☐ Yes  
☒ No

Each "An order" can have several "a Products"

☒ Yes  
☐ No

Each "a Product" can have several "An orders"

☒ Yes  
☐ No

Check the integrity between the relation file and ...

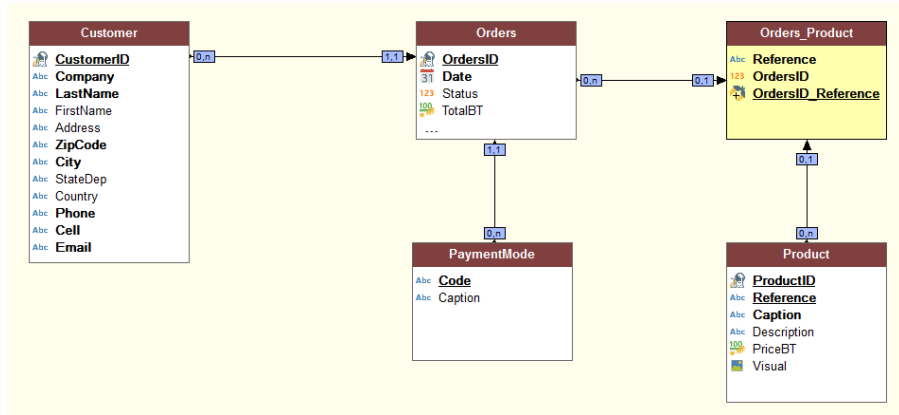
☐ "Orders" file  
☐ "Product" file

☐ Display the advanced cardinalities

Navigation: ? < [Green Arrow] >

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 the 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 the 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.

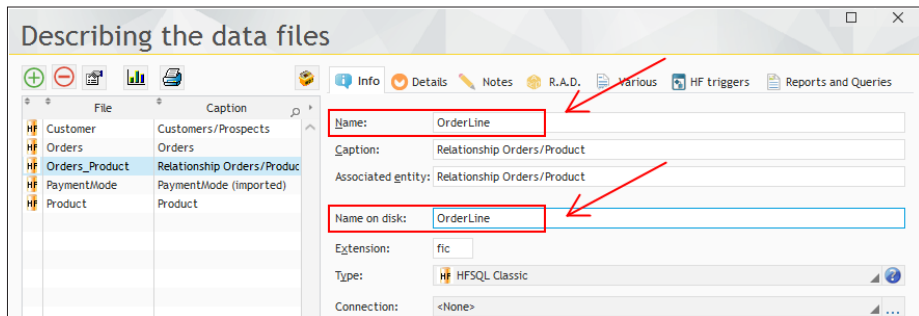



We are now going to modify the relation file that was created by WEBDEV. Indeed, this file will contain the order lines.

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 the imported file was modified. However, it is a little bit more complex in this case: 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 name of the file: "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 the items found in the data file.

OrderLine					
Number of items and index:		3	Size in bytes:	68	<input type="checkbox"/> Display in physical order
Key	Name	Caption	Type	Size	
Reference	Reference		Text	50	
OrdersID	OrdersID		Numeric	8	
OrdersID_Reference	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".

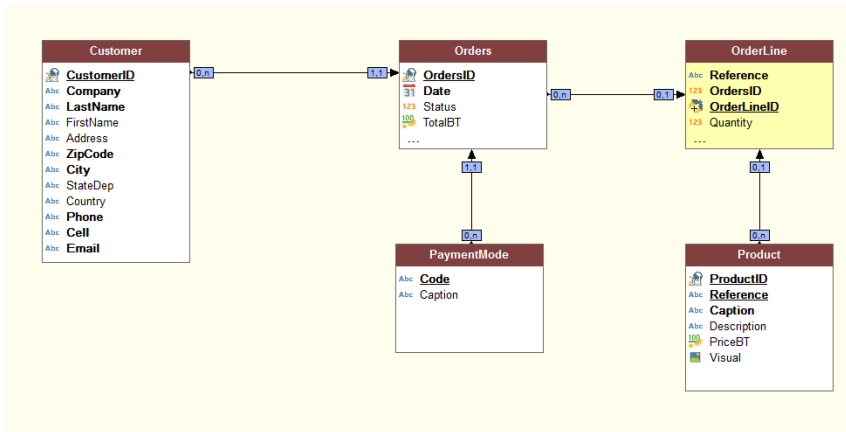
- We are now going to add 3 new items into this relation file: Quantity, TotalBT and TotalIOT.
1. Position on a new row and create the "Quantity" item. This item is a "Numeric" item.
  2. Position on a new table row and create the "TotalIOT" item. This item is a "Currency" item.
  3. A window is displayed, 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				
Number of items and index: 6    Size in bytes: 92 <input type="checkbox"/> Display in physical order				
Key	Name	Caption	Type	Size
Reference	Reference	Reference	Text	50
OrdersID	OrdersID	OrdersID	Numeric	8
OrderLineID	OrderLineID	Identifier of OrderLine	Composite Key	58
Quantity	Quantity	Quantity	Numeric	4
TotalIOT	TotalIOT	TotalIOT	Currency	10
TotalBT	TotalBT	TotalBT	Currency	10

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

► The analysis is as follows:



## Generating the analysis

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

The generation is automatically proposed when closing the data model editor while modifications have been performed.

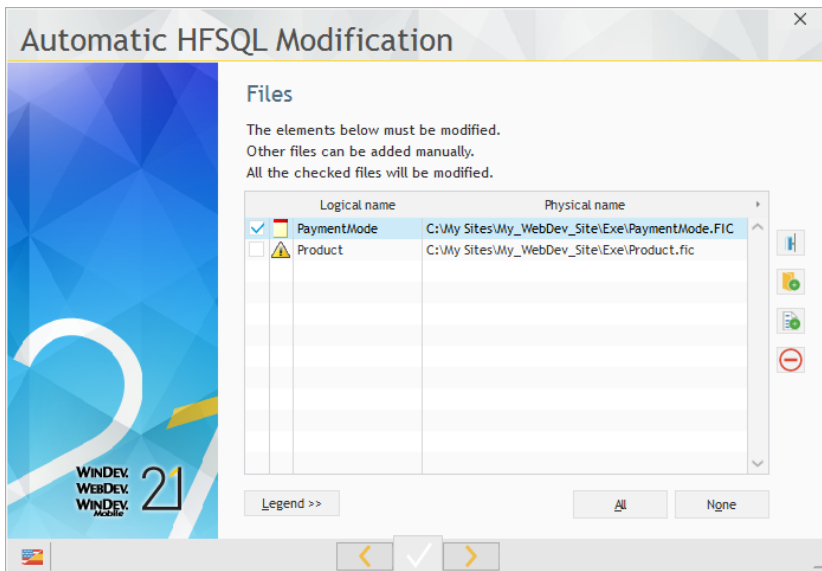
You also have the ability to generate the analysis manually. That's what we are going to do.

- 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 the deployed data (also called "modification procedure of data files"). This operation is used to update the 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 enter the passwords for protecting the data files modified by the automatic modification. Keep the default options 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.





DEVELOP 10 TIMES FASTER

## PART 3

Intranet site  
with data

21





## LESSON 3.1. OVERVIEW

**This lesson will teach you the following concepts...**

---

- Overview of the site created in this part



Estimated time: 5 min

## Overview of the 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 us to develop the Internet section of the site. You will have the ability to list the new products and to view their details.

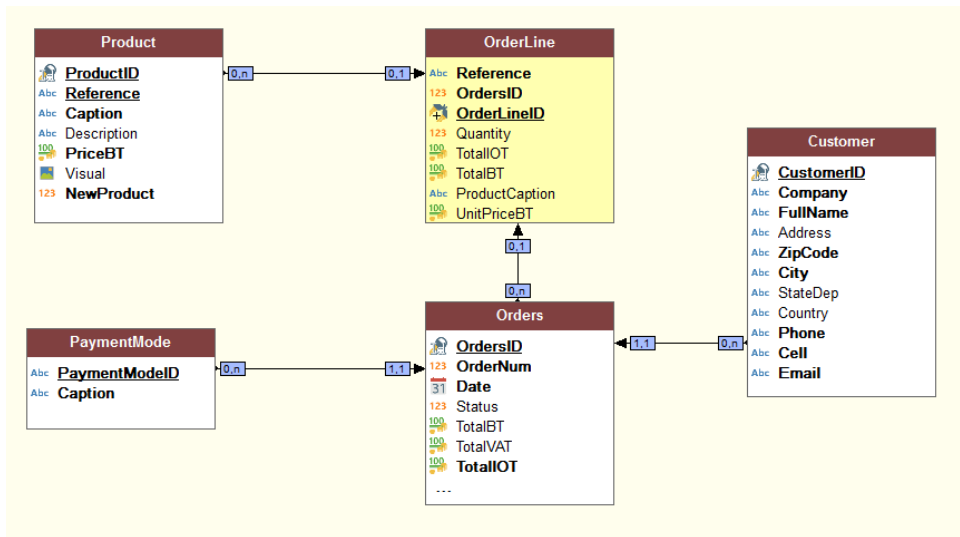
Part 5 will allow us to continue the development of the site, by proposing the following features:

- Printing reports,
- Managing the 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 analysis of the site is an improved version of the analysis that was created in the previous part.

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 in part 3.
  - A project used to start the operations in part 5 (this project includes all the operations performed in parts 3 and 4).
  - A final project used to start 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 min

## 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 the WEBDEV session.

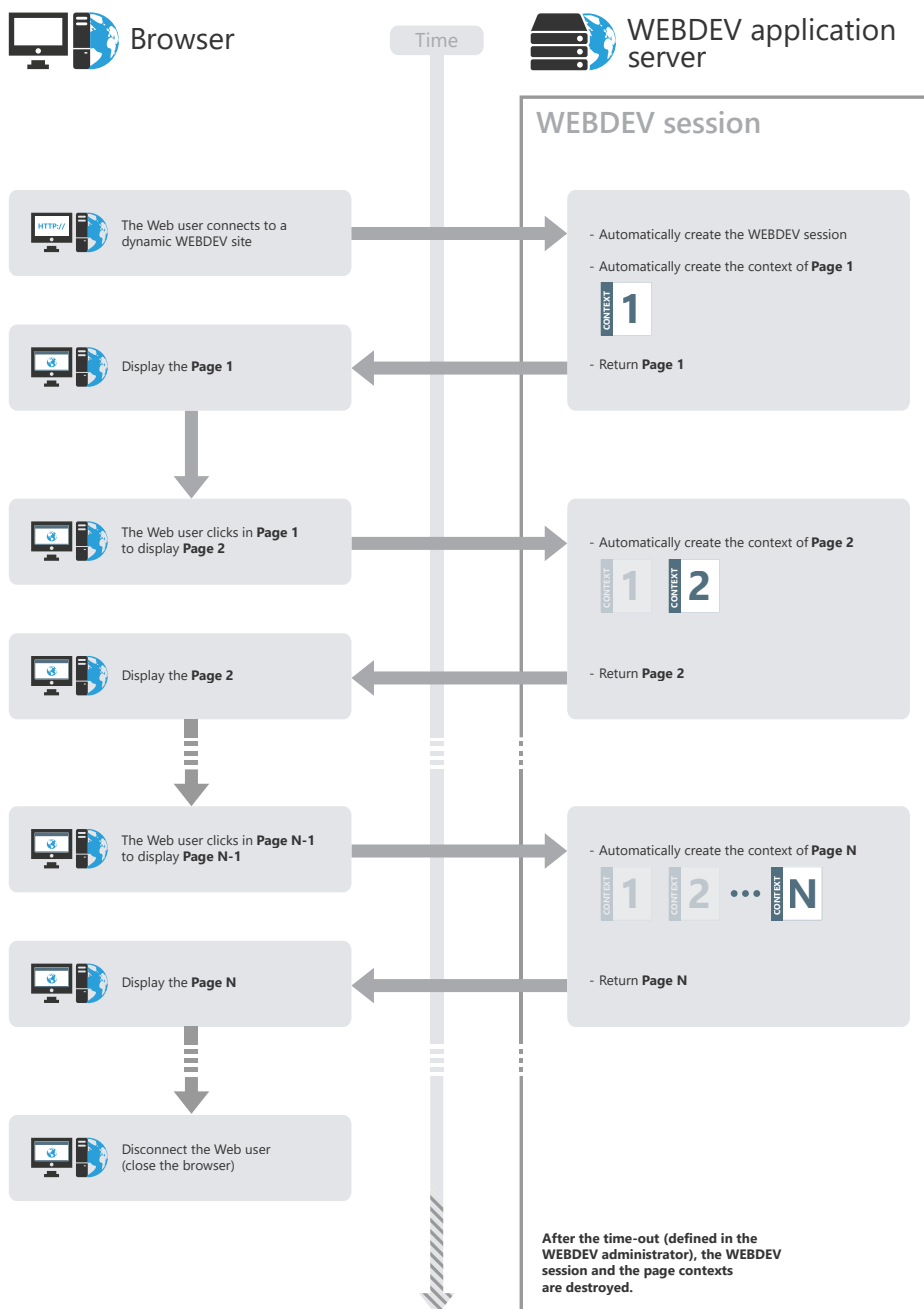
If the same page is called several times:

- 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 min

## 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 in order to display the home window.
2. In the home window, click "Tutorial" and select "Full WebDev Site (Exercise)".



Answers


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)".

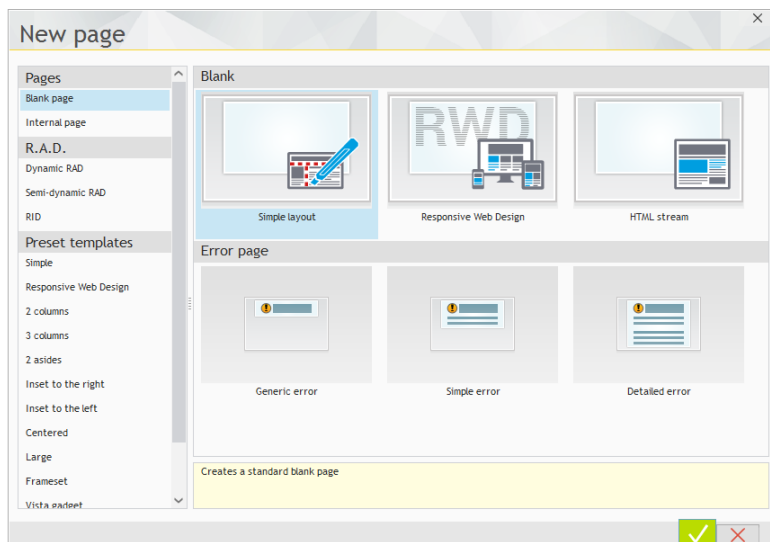
## Creating a dynamic page used to list the products

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.

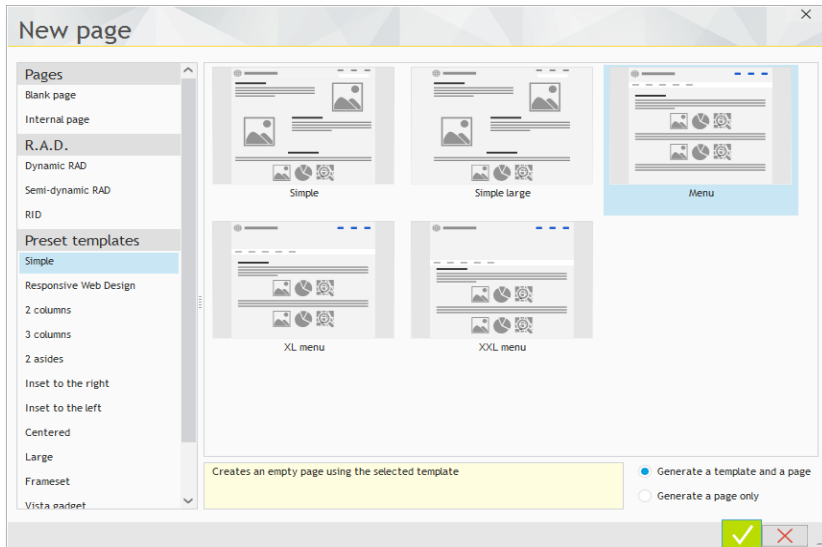
### Creating a page that is using template

► 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 is used to group part of the interface and processes in 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 appears 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 directly displayed in the editor 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. This operation will be presented later in this lesson.


► By default, this page contains the different controls used to view the presentation possibilities. In our case, we are going to delete these controls:

1. Select the controls found in the page (press the [CTRL] + A keys).



Notes

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

2. Delete the controls ([DEL] key).
3. Only the controls corresponding to the page template remain in the page.
4. Save the page: click  among the quick access buttons (or use the CTRL S shortcut).
5. In the window that is displayed, specify the title of the page: "List of products".

6. Validate.

## Creating the controls

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



Notes

### What is a Looper control?

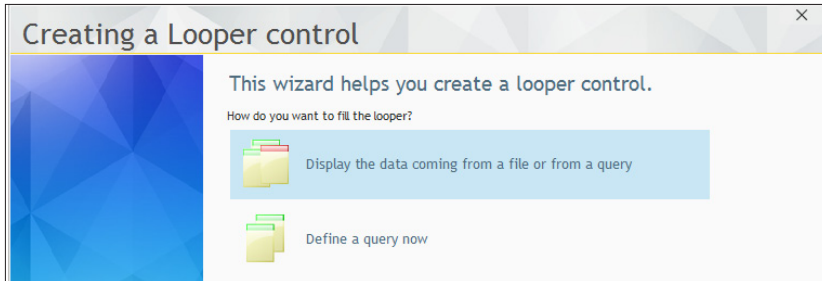
The best way for displaying elements in list format in a Web page is to use a Looper control. A Looper control includes an area (containing several controls) that is repeated a given number of times.

Different data can be found in each repeated area.

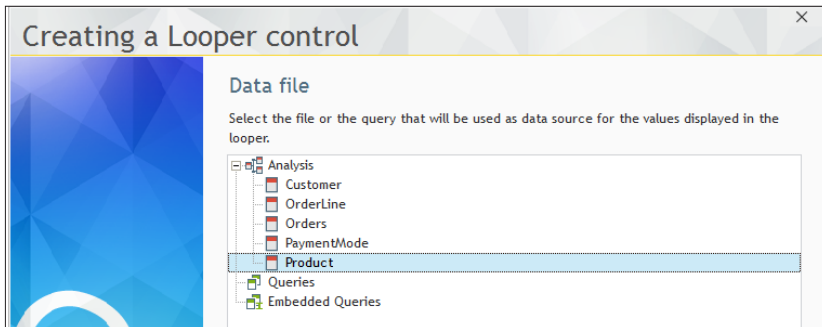
- To create the Looper control:

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 asks how to fill the Looper control:
  - by programming (this feature will be presented in a forthcoming lesson),
  - from the database, from a data file or from a query,
  - from a WLanguage variable.

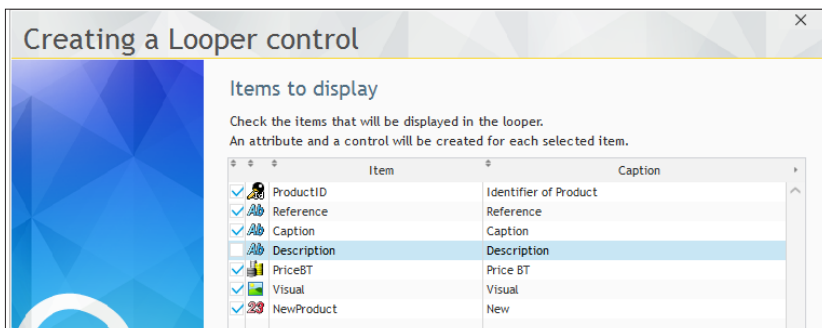
In this case, the Looper control must display all the products: select "Display the data coming from a file or from a query".



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 the queries found in the current project. Expand "Analysis" if necessary and select the "Product" file.

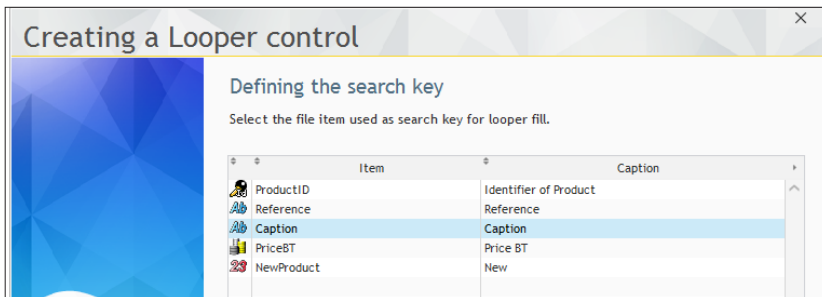


6. Go to the next step.
7. The wizard proposes the list of items found in the data file. By default, all the items are checked in order to be displayed in the Loop control. In our case, we are going to display all the items EXCEPT for the product description.



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.



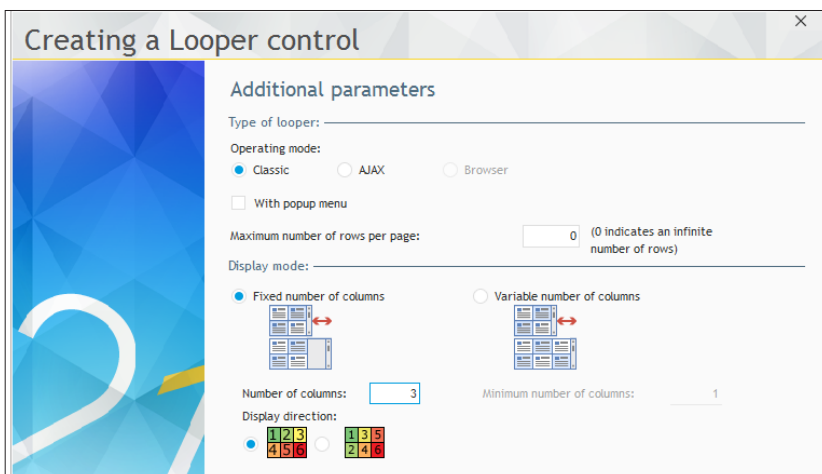
Select "Caption" and go to the next step.

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 during the page load.

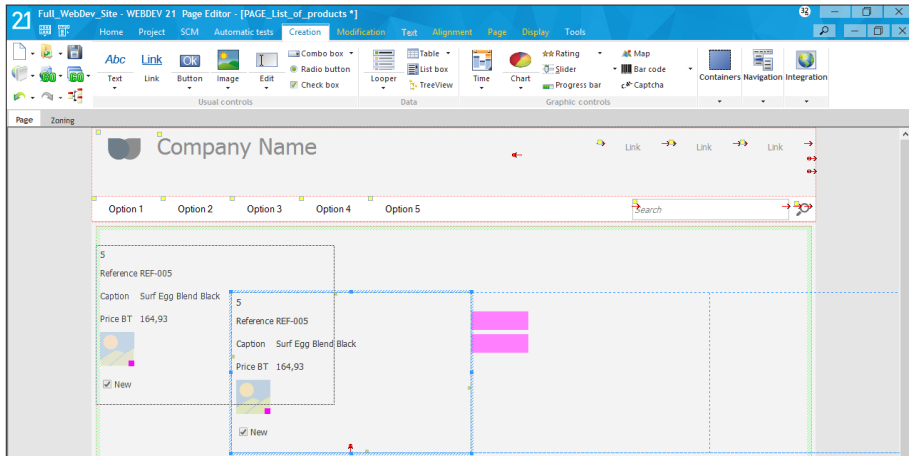
Select the "Classic" operating mode.

- 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. Type 0 in the "Maximum number of rows in a page" control.
- The Looper control will display the products on 3 columns. In the "Number of columns" area, replace 1 by 3.
- Keep the proposed display direction.



10. Validate the wizard. The Looper control is automatically created in the page.

- We are going to reposition the control in the page:
  1. Select the control.
  2. Move the control with the mouse in order to position it at the top left of the page. When the control reaches the limit of the display area in the page template, green lines are displayed: these lines are used to position the control inside the display area.

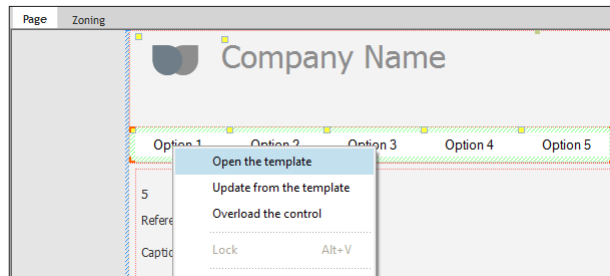


Let's take a look at the control that was just created: the data is already displayed in the control, even in the editor.

**This concept is called "Live data":** the content of the data files found on the development computer is used in the windows or reports handled in the editor. This feature is very useful to define the size of the 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 used later in our development.
  1. Select the "STC\_ProductID" control. This control corresponds to the identifier (control with a number at the top of the loop).
  2. Display the description window of the control ("Description" from the popup menu).
  3. In the "GUI" tab, uncheck "Visible".
  4. Validate the description window of the 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.
  1. Display the popup menu of the menu (right mouse click) and select "Open the template".





2. 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 the option (select "Option description" from the popup menu).
5. In the "General" tab:
  - Enter the caption of the option: replace "Option 1" by "List of products".
  - In the "Action" combo box, select the action "Display the PAGE\_List\_of\_products page".
  - Validate the description window of the 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:

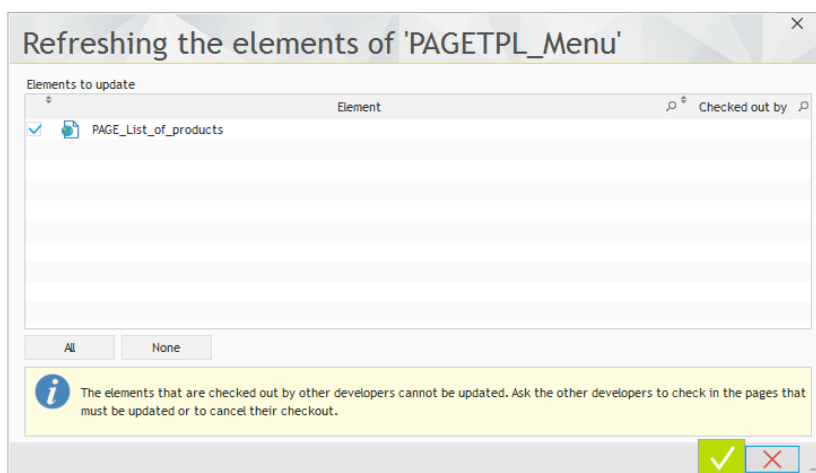
1. In the orange bar of the template, click  : this button is used to propagate the template modifications to all the pages that use the template.



Notes

If the modifications made to the template are not applied to 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.




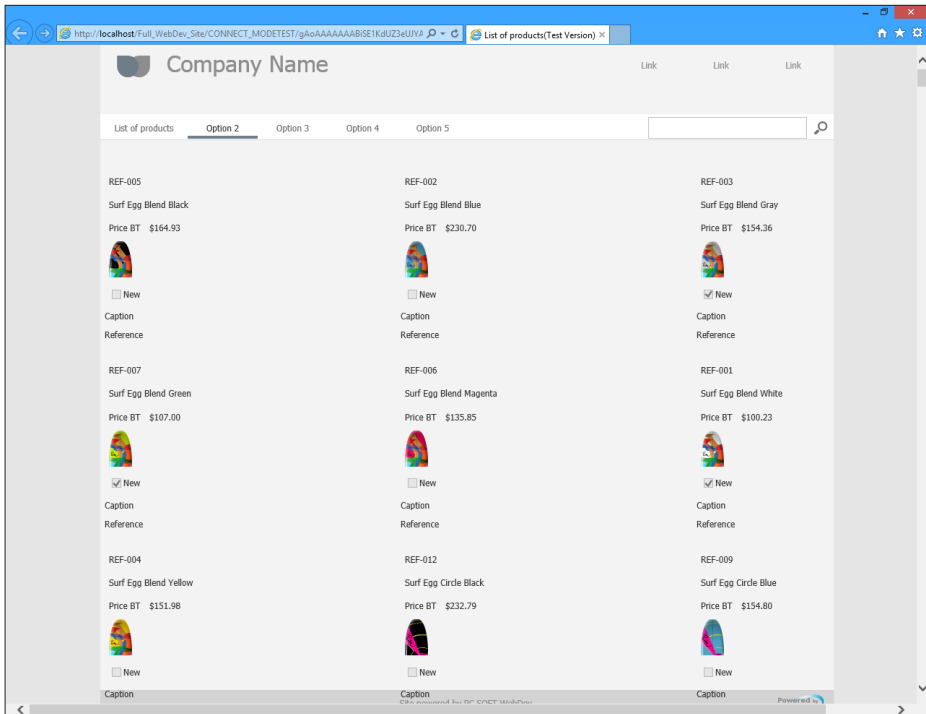
3. Validate the window for updating the template.

4. Close the page template.

Our page is ready to be checked.

## Test of the page

- 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 the products via a dynamic "Product form" page

Now that the list of products is displayed, you may want to modify a product. We are now going to create a page used to display the product details in order to modify them.

### Creating the page

- To create a page used to display the details of a product:
  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. Save the page (  or CTRL S). The title of this page is "Product form", its name "PAGE\_Product\_form" is automatically proposed. Validate the backup window.

### 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 "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 the different items found in the "Product" data file.

First, we are going to indicate to the page which product must be displayed. To do so, all you 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 processes of the page.
  2. Enter the following code in the "Global declarations" process:

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

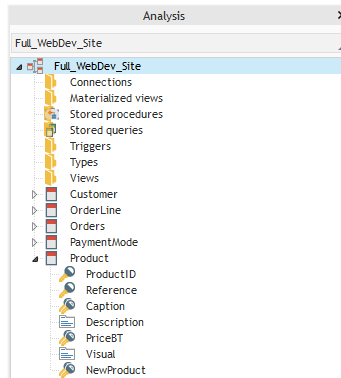
3. Let's study this code:
  - The PROCEDURE keyword in the "Global declarations" process is used to associate a procedure with the page opening.
  - The procedure is named "MyPage". At run time, this keyword will be automatically replaced by the name of the current page.
  - The procedure expects the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

## Creating the edit controls

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

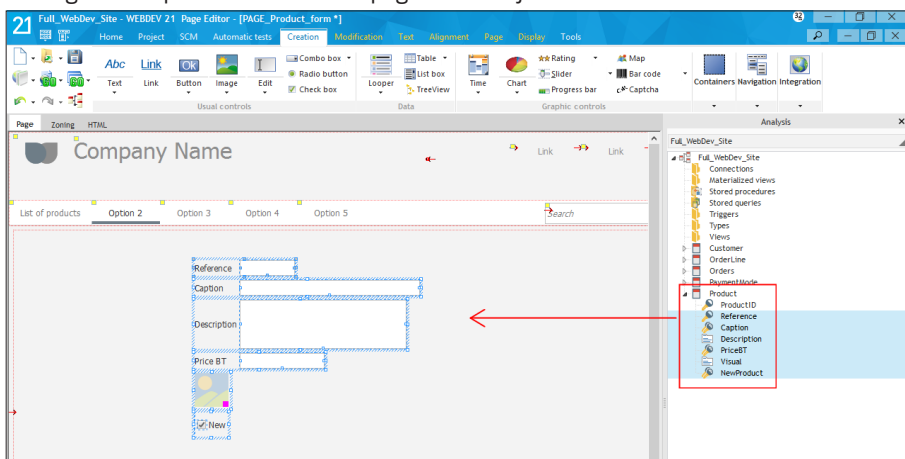
► To create the controls in the page:

1. Display the "Analysis" pane if necessary: on the "Home" pane, in the "Environment" group, expand "Panes" 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 the "Product" file: the items found in the data file are listed.



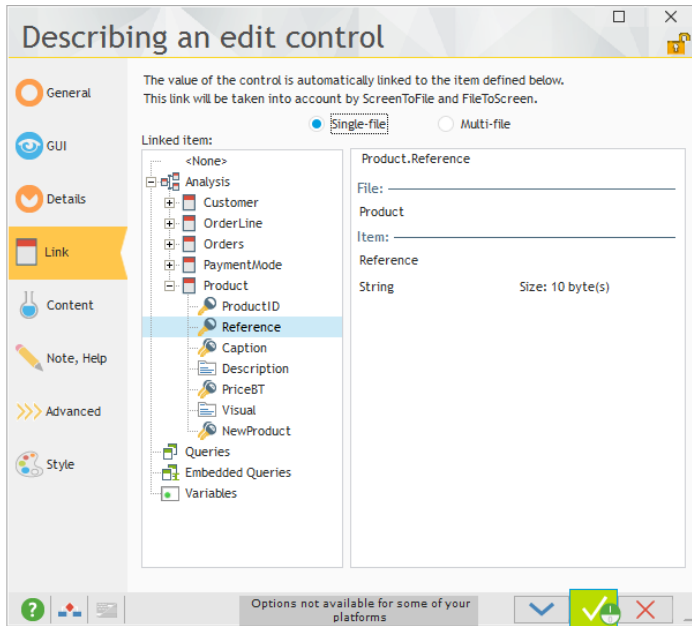
3. With the mouse, select all the items displayed in the pane (except for the "ProductID" item). You can for example use the mouse lasso or the multi-selection by keeping the [CTRL] key down.

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:

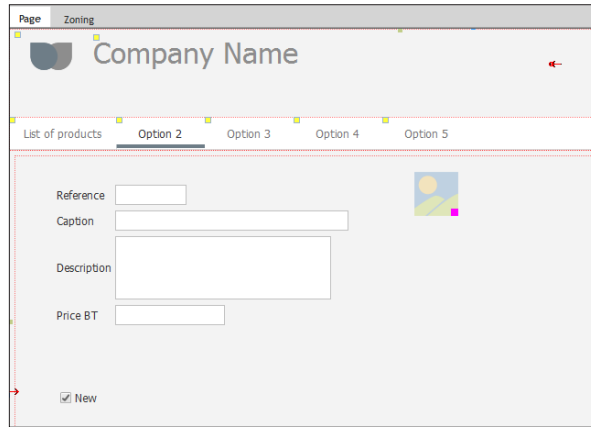
- Select the "Reference" control for example.
- Display the popup menu (right mouse click) and select "Reference".
- Display the "Link" tab of the description window:



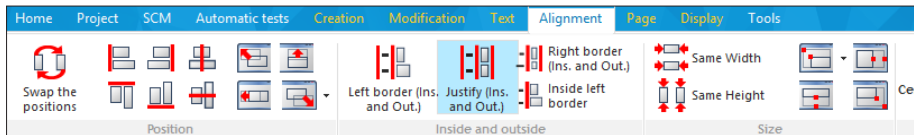
This tab is used to see that the current edit control is linked to the "Reference" item found in the "Product" data file.

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

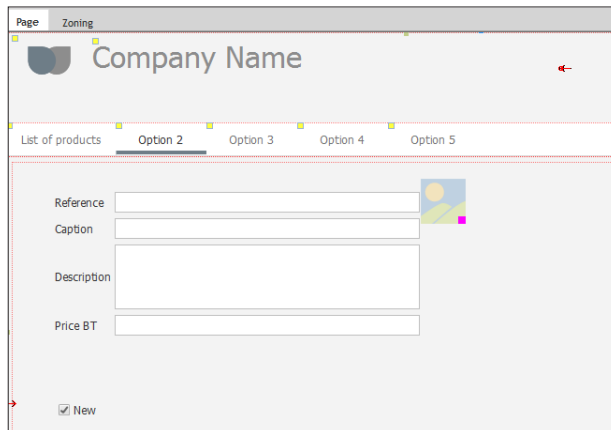
- Reposition the controls in the page in order to get the following interface:



- Enlarge the Image control used to view 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 the other controls.
  2. Display the "Alignment" pane of the WEBDEV menu. This pane contains all the alignment options available for the controls.



3. We want the outside and inside borders of the 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 code of the page, we are going to enter the code used to:

- find the product to display.
  - display the data in the page.
- To display the data in the page:
1. Press the [F2] key. The code editor displays the different processes of the page.
  2. In the "Global declarations" process, enter 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. In our case, if the record was not found (**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 the buttons

The "PAGE\_Product\_form" page will be used 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 be used to redisplay the previous page.
- the validation button will be used to check the data entered and to 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 caption of the button (press the [SPACE] key to edit the caption): the new caption is "Cancel".




► Enter 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 the 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 the previous page.
- Therefore, this code is used 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 the 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 must have a "Browser" action only.

## Validate button

► To create the "Validate" button:

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

The screenshot shows a web application interface with a header bar containing a logo and the text "Company Name". Below the header is a navigation bar with tabs: "List of products", "Option 2", "Option 3", "Option 4", and "Option 5". The main content area displays a form with the following fields:

- Reference: REF-005
- Caption: Surf Egg Blend Black
- Description: A surfboard with outstanding lines, ideal regardless of your practice
- Price BT: \$164.93
- ☒ New

At the bottom of the form are two buttons: "Validate" and "Cancel".

The validation button will be used to:

- check the data entered: this check consists in verifying that the different page controls have been filled by the user. This check does not require any access to the server and it can be performed 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.
- Enter 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. Enter 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("Enter a caption.")
    ReturnToCapture(EDT_Caption)
END
IF EDT_Description ~= "" THEN
    Error("Enter a description.")
    ReturnToCapture(EDT_Description)
END
IF EDT_PriceBT = 0 THEN
    Error("Enter a price.")
    ReturnToCapture(EDT_PriceBT)
END
```

3. Let's study this code:

- For each edit control found in our page, a check is performed to find out whether a value was entered.
  - The "~=" operator is used to check the equality, excluding the case and the punctuation.
  - If no value is entered, an error message asks the Web user to perform an input **Error**). The execution of the code is stopped and the input is forced in the relevant edit control via **ReturnToCapture**.
4. In the server click code of the button, enter the code for saving the 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 the 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, the "PAGE\_List\_of\_products" page is redisplayed.

6. Save the modifications (  or CTRL S).

7. Close the code window.

## Managing the product image

In the Product file, an item is used store the image associated with the product.

The Image control is already found in our page, allowing us to see the image, but we are going to give the Web 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 will be using an Upload control.



Notes

The upload consists in copying a file from the client computer to the server.

On the contrary, the download consists in copying a file from the server to the client computer.



Notes

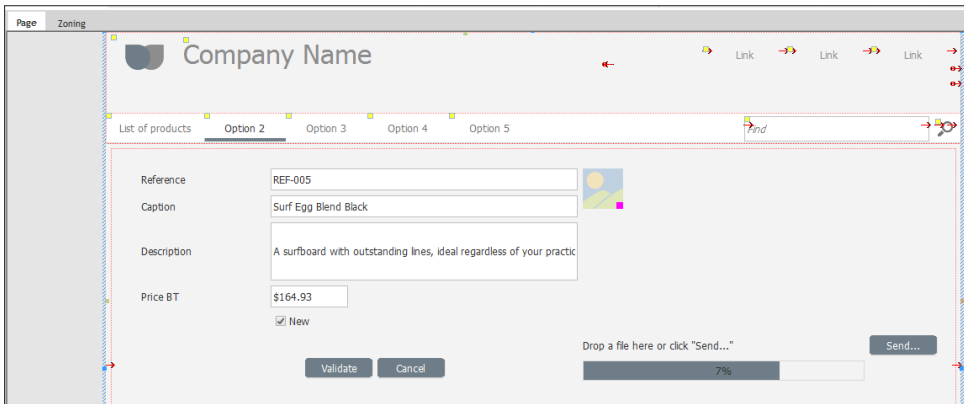
WEBDEV proposes to manage the file upload via two specific controls:

- an Upload control used to upload a single file,
- an Upload control used to upload several files.

In this example, the user will upload a single file at a time, therefore we will be using 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 "Send a single file".
3. Click the position where the control must be created in the page (below the Image control for example).



The Upload control includes:

- a Static control,
- a Progress Bar control, used to follow the progress of the upload,
- a button allowing the user to select the file to upload.

We are now going to adapt the code of the control in order to manage the file upload in our site.

► To configure the Upload control:

1. Display the code of the "Send" 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. Enter the following code in the "Receiving the uploaded files" 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 go to the next line in a code line. They are used here for layout purposes.  
In the code editor, you have the ability to delete them and to 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 name of the file is kept.
- Then, the path of the uploaded image is stored in a global variable **gsImagePath**.



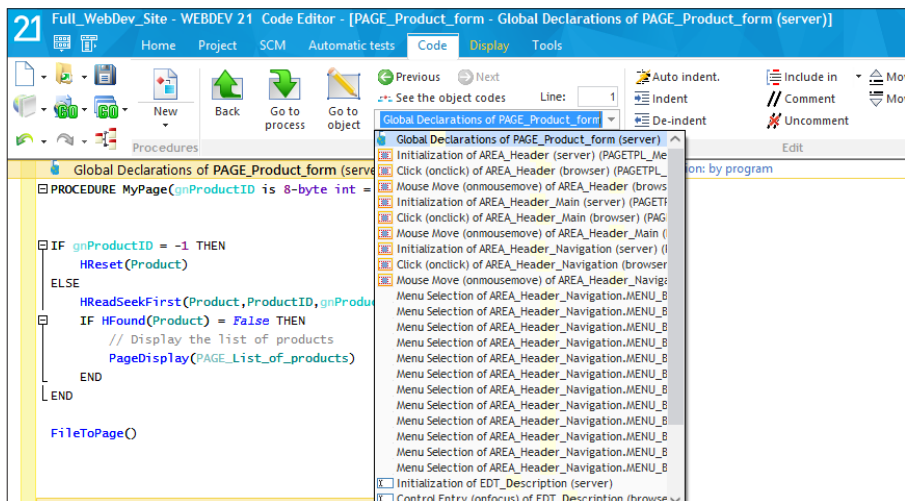
## Notes

**Why use a global variable?**

A global variable is required here because the path of the image will be used in the code of the "Validate" button, to save the new image in the data file.


See the appendices for more details about the use conditions of local and global variables.

- **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: "Error: The identifier 'gsImagePath' is unknown or inaccessible". Indeed, this global variable was not declared.
6. To declare the global variable:
- Display the code for declaring the 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).



- Enter the following code after the declaration of the procedure:


```
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 you have to do now is save the image in the database.

- To save the image in the Product file:
  1. Display the code of the "Validate" button:
    - Select the "Validate" button.
    - Press the [F2] key.
  2. In the server click code of the button, enter the following code AFTER the call to **PageToFile**:

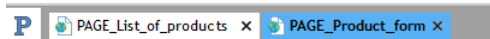
```
IF gsImagePath<>" " THEN
  Product.Visual = fLoadBuffer(gsImagePath)
END
```

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 by **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.

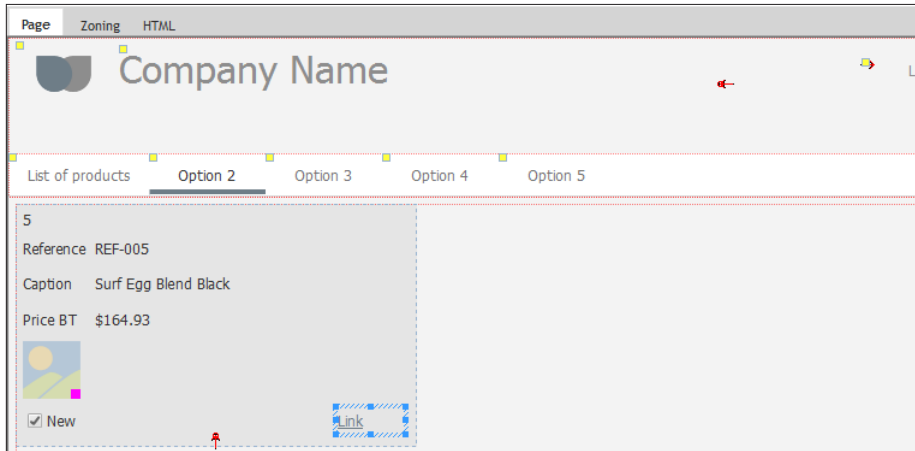
## Displaying the form from the list of products

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

- First, 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 in the Looper control to create the link (at the bottom right for example).



4. Modify the caption of the link (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, enter the following code in the "Click" process:

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



Notes

Try the assisted code input: as soon as you type the opening bracket "(", a popup list is displayed, proposing the name of all the pages found in the project. All you have to do is select the page with the keyboard or mouse.

If you do not find the name of the requested page in the list, it means that this one 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 the product to display. This identifier corresponds to the identifier of the 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 square brackets are used to specify the row and LOOP\_Product is used to get the current row in the loop.




Notes

By default, ATT\_ProductID returns the value of the attribute for the row that was clicked. The code can be written even more simply:

```
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 modification of a product have been implemented, we are now going to run a test to check that everything is operating properly.

- 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\_List\_of\_products" page and validate.



Notes

The first project page corresponds to the first page that is opened when starting the site.

The first project page can be defined:

- during the test of the project.
- in the project explorer: select the requested page, display the popup menu and select "First dynamic page of the project".

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

- Modify the price of a product and validate.
  - The new price of the product 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 also want to be able to add a product. To do so, we are not going 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, we are going to modify the opening mode of the "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 code of the page.
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 identifier of the element does not exist. To manage this case, we will be using 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 the items found in the "Product" file with the default values to manage a new record.
  - If the product identifier has a value that differs from -1, we retrieve the code used to open the form in modification.
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 "Click" process, replace the current code by the following one:

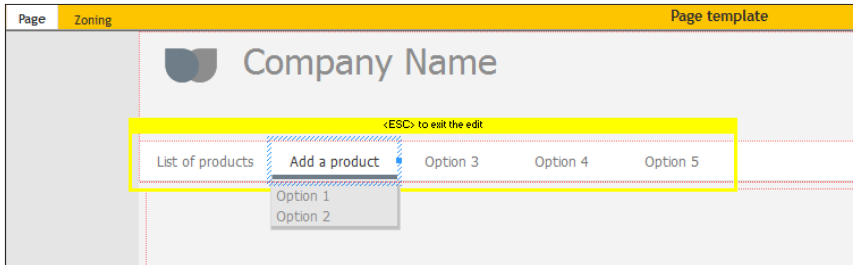
```
PageToFile ()
IF gsImagePath<>" " THEN
    Product.Visual = fLoadBuffer (gsImagePath)
END
IF Product..NewRecord THEN
    HAdd (Product)
ELSE
    HModify (Product)
END
PageDisplay (PAGE_List_of_products)
```


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

- We are now going to modify the page menu to give the user the ability 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 of the option ("Option description" from the popup menu).
6. In the "General" tab:
  - Enter the caption of the option: replace "Option 2" by "Add a product".
  - In the Action area, select the action "Display the PAGE\_Produt\_form page".
7. Validate. The menu option appears:



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

We now can check the addition of a product.

### Checking the addition of a product

- To check the addition of a product:
  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 20
    - Description: Surf with WebDev colors
    - Price: 150

Company Name

List of products **Add a product** Languages Find

Reference: REF-337

Caption: Surf 20

Description: Surf with WebDev colors

Price BT: 150

☐ New

Drop a file here or click "Send..."

4. Validate. The new product is displayed 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 20
  - Description: Surf with WebDev colors
  - Price: 150
7. Validate. A specific page is displayed:

**The security mechanism of Full\_WebDev\_Site was triggered.**

Duplicate error on the <Product.Reference> item

**Info**

ProductID	Caption	Description	PriceBT	Reference	Visual	NewProduct
338	Surf 20		150	REF-337	0	

Values of items: < >

1 area is concerned.

**Details**

A data that already exists was entered for the <Product.Reference> item. This is not allowed. Change this data before saving.

REF-337 The red areas indicate the values of the items that must be modified.

[More details...](#)

This page signals to the user that a duplicate was found: the reference (that is a unique key) is identical for two products.. This page is used to modify the value of the reference (displayed in a red area): enter "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 custom procedure or 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 created records can be immediately viewed in the Looper control of the "PAGE\_List\_of\_Products" page.

They can also be viewed with WDMAP that was already presented in the "My first pages" lesson.



DEVELOP 10 TIMES FASTER

## PART 4

Internet site  
with data

# 21







## LESSON 4.1. OVERVIEW

**This lesson will teach you the following concepts...**

---

- Overview of the site created in this part



Estimated time: 5 min

## Overview of the 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

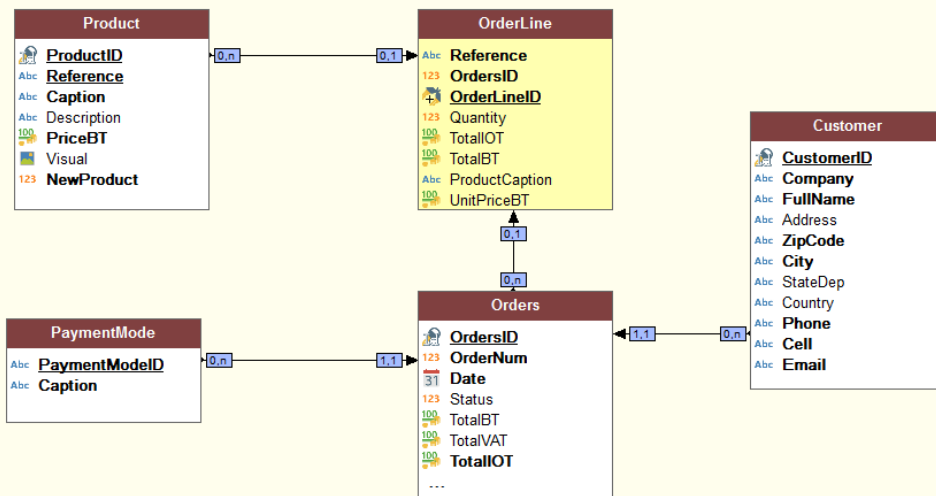
In a WEBDEV AWP site, all the site pages can be referenced.

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

- Printing reports,
- Managing the 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 min

## 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, ...

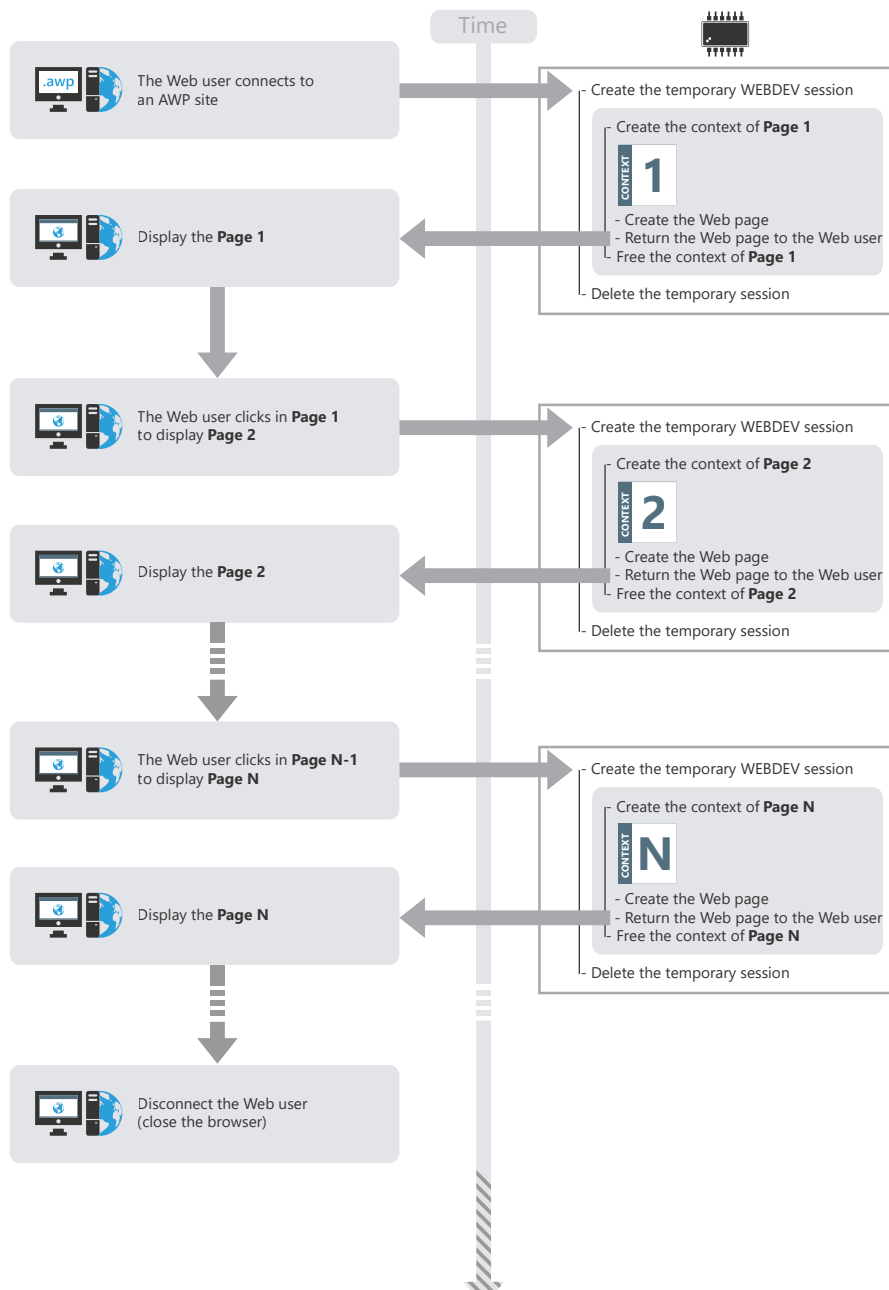
When the AWP page was sent to the Web user, these elements are destroyed.



Browser



WEBDEV application server



## 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.
- Creating an AWP product form.
- 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 min

## 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 the AWP mode.

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




Answers

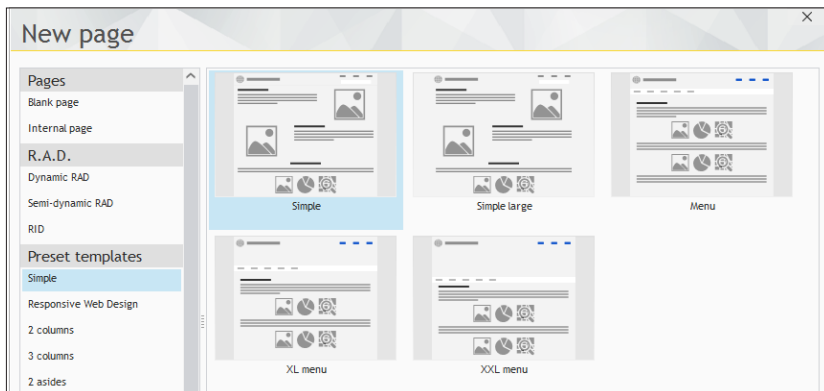
A corrected project is available. This project contains the different pages created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WebDev site (Answer)".

## 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 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 is displayed in the editor, with controls proposed by default. Delete these controls:

- Select the controls found in the page (press the [CTRL] + A keys).
- Delete the controls ([DEL] key).

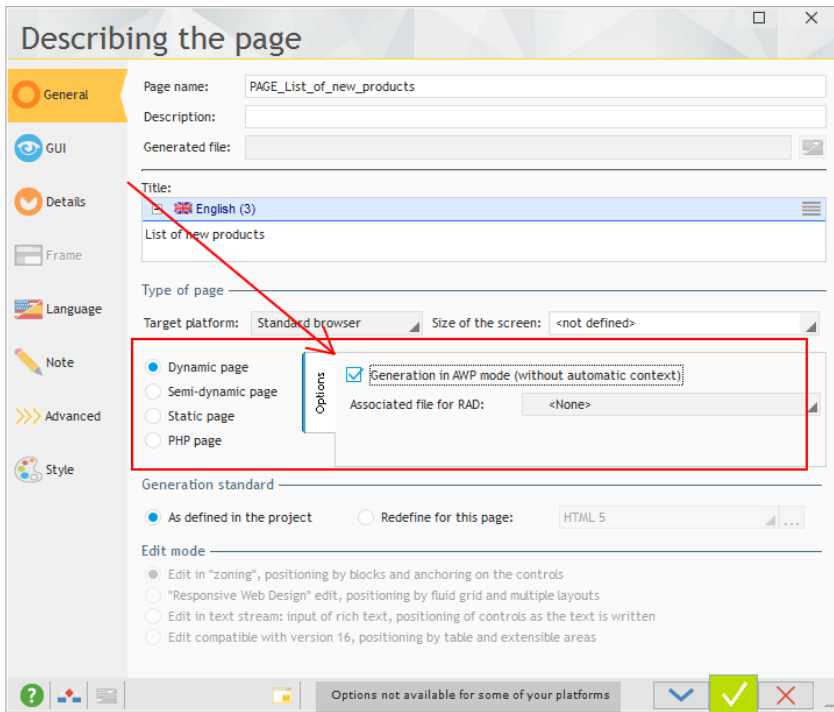
3. Save the page: click  among the quick access buttons.

4. In the window that is displayed, specify the title of the page: "List of new products".

5. Validate to save the "PAGE\_List\_of\_new\_products" page.

► This page must be referenceable. To do so, it must be generated in AWP mode:

1. Display the description window of the page: display the popup menu and select "Description".
2. Check "Generation in AWP mode (without automatic context)".



3. Validate.

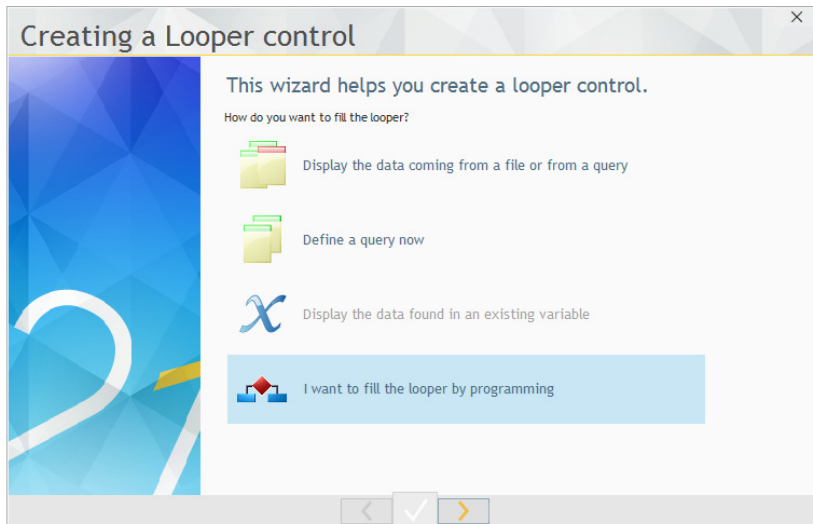
4. Save the modifications by clicking  among the quick access buttons.

## Creating the list of new products

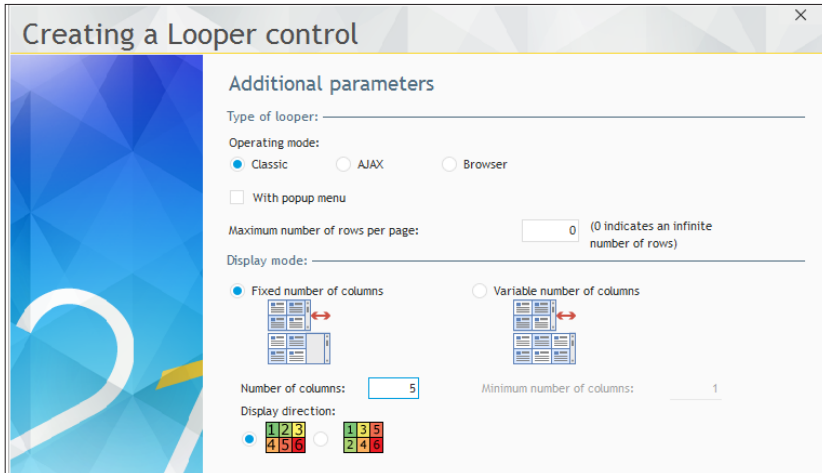
We are going to display the list of new products. To do so, we will be using a Looper control. We have already explained in the previous part how to create this type of control without programming, via the wizard. For this page, we are going to fill the Looper control by programming.

### 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 movement of the mouse.
  2. Click 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 loopers by programming". Go to the next step by clicking the arrow at the bottom of the screen.



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 during the page load.  
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. Type 0 in the "Maximum number of rows in a page" control.
  - The Looper control will display the new products on 5 columns.  
In the "Number of columns" area, replace 1 by 5.
  - Keep the proposed display direction.



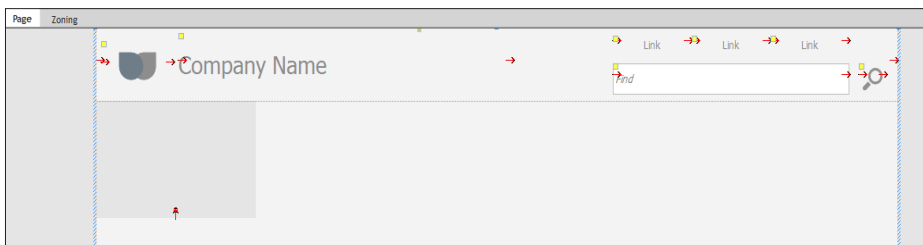
5. Go to the next step.

6. The last step is used to specify the name of the control (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 the 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.

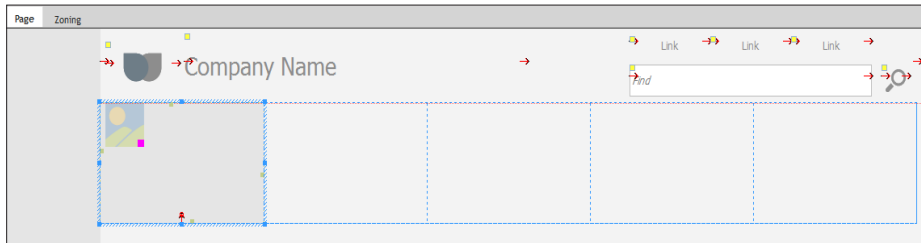
Each row will display:

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

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

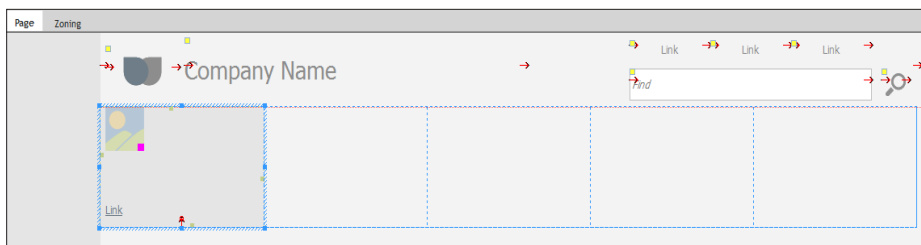
## Creating the controls found in the loop

- 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.
  2. Hover the first row: a green border appears, allowing you to view the available area.
  3. Click the top left corner of the first row found in the Looper control. The Image control is created.



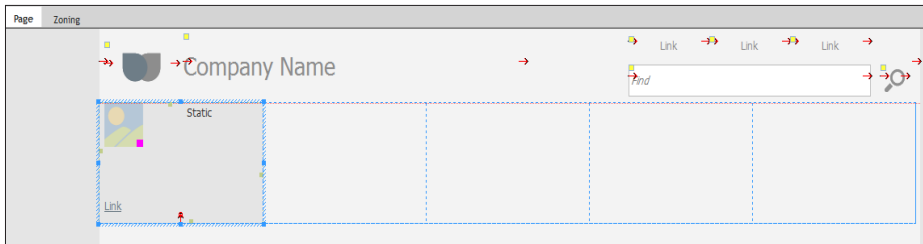
4. Display the description of the Image control to modify the characteristics of the control (double-click the control).
5. Modify:
  - the name of the control: IMG\_Visual.
  - the type of the image: "From a database: memo (medium speed)". Indeed, the image of the product is stored in memo format in an item of the Product data file.
  - the display mode of the image: "Homothetic centered" with "High-quality display" 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) in order to create the link.



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

- 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 Loopier control (on the right of the image) to create the static control.



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

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

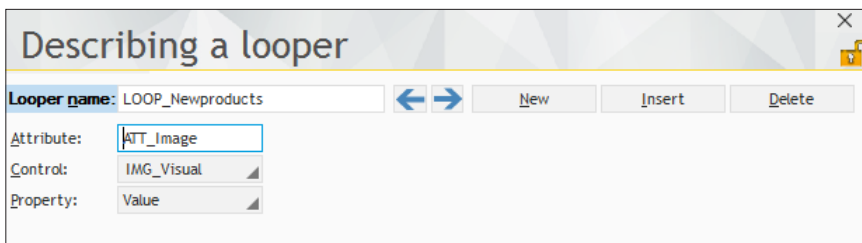


Notes

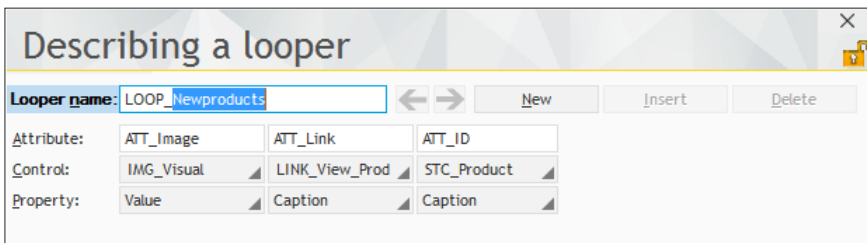
A Loopier control includes:


- controls, that are repeated on each row.
- attributes. An attribute is used to define the characteristic of the control that will be modified on each row. For example, if the value and the color of the 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 the Loopier control (double-click the control).
  2. In the top section of the window, a single attribute is created by default. We are going to define 3 attributes (one for each control found in the loopier).
  3. Modify the attribute created by default:
    - Rename the attribute to "ATT\_Image".
    - Select the "IMG\_Visual" control.
    - Select the "Value" property. Indeed, the value of the image will be modified on each row.



4. Create a new attribute by clicking the "New" button:
  - Rename the attribute to "ATT\_Link".
  - Select the "LINK\_View\_Product" control.
  - Select the "Caption" property. Indeed, the caption of the control will be modified on 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 caption of the control will be modified on each row.



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

The Looper control is now completed. We can now program the fill operation for the 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. Enter the following code in the initialization code of the 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 "NewProduct" item is set to True are selected.




- **LooperAddLine** is used to add a new row into the Looper control. The attributes of the row are updated with the data of the current record:
- the value of the "ATT\_Image" attribute corresponds to the image found in the "Visual" item of the "Product" file.
- the caption of the "ATT\_Link" attribute has for value the content of the "Caption" item found in the "Product" file.
- the caption of the "ATT\_ID" attribute has for value the content of the "ProductID" item found in the "Product" file.

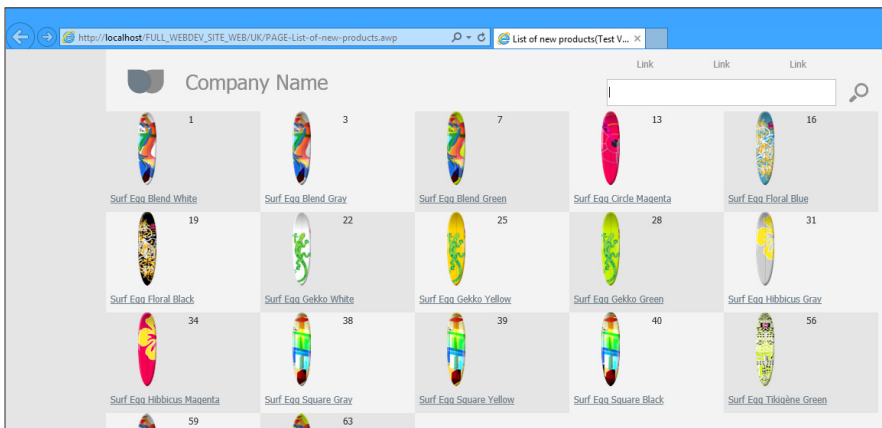
4. Save the modifications (  or CTRL S).

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.





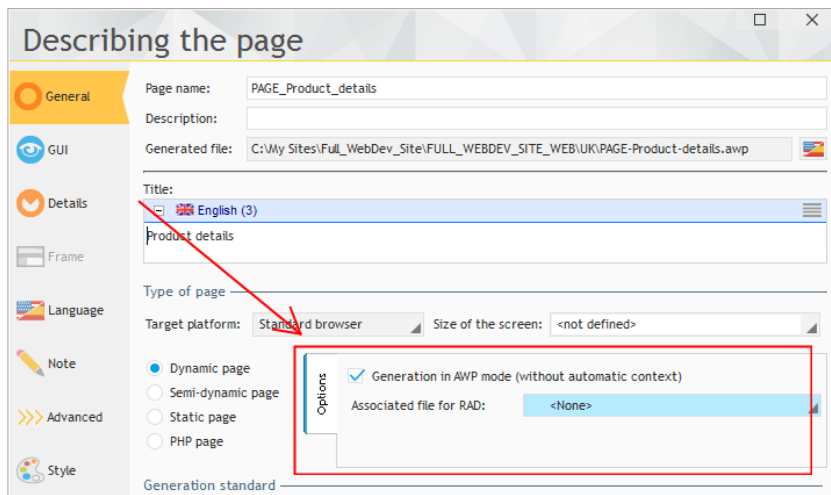
► Close the browser. The WEBDEV editor is redisplayed.

## 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 details of the product. This page will be displayed when the Web user clicks the link used to display the name of the product in the Loopier control.

### Creating the page

- ▶ To create a page used to display the details of a product:
  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. Save the page (  or CTRL S). The title of this page being "Product details", its name "PAGE\_Product\_details" is automatically proposed. Validate the backup window.
- ▶ Like the page used to list new products, this page must be generated in AWP mode:
  1. Display the description window of the page (select "Description" from the popup menu).
  2. Check "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 the different items found in the "Product" data file.

First, we are going to indicate to the page which product must be displayed. To do so, all we 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 processes of the page.
2. Enter 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 the current page.
  - The procedure expects the gnProductID variable (that is an 8-byte integer) in parameter. This variable corresponds to the identifier of the product that must be displayed in the page. The type of this variable is identical to the type of the 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 the controls

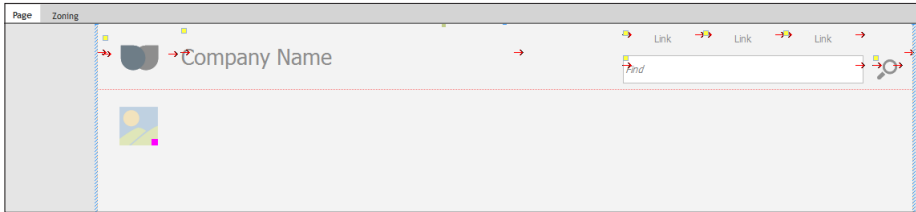
We are now going to create the edit controls used to display information about the selected product in the page.

In the previous part, we explained how to create controls by Drag and Drop from the analysis pane. In this part, we are 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 caption of the product, its name and reference.
- a rich static control for the description of the product.

- 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 the Image control to modify the characteristics of the control (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 size of the image 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 the control.

- To create the simple Static control used to display the caption of the product:
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 the image for example).



Notes

To simplify the positioning of controls, press the [F7] key. A press on this key is used to see the area occupied by the control, a second press on this key is used to display an outline around the control (in edit only).

This allows you to view the border of the 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 directly.
2. Click in the page (on the right of the image for example, below the STC\_Title control).

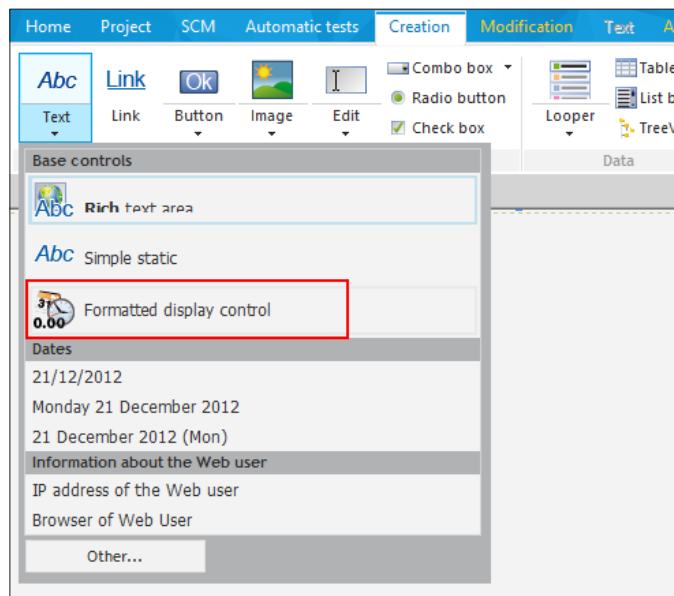


3. Rename the control: "RTA\_Description".

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

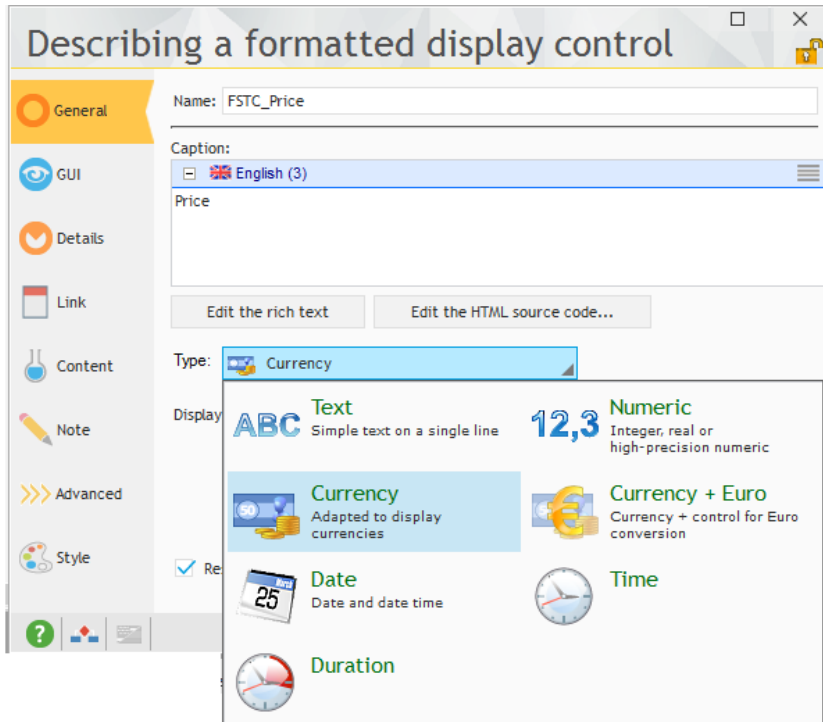
► To create a formatted display control:

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 name of the control: "FSTC\_Price",
    - Delete the caption.
    - Modify the type of control: select "Currency".



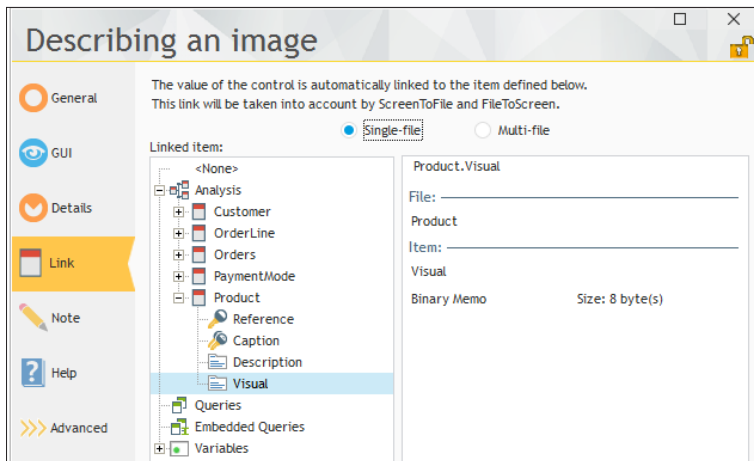
3. Validate.

- Create a simple Static control to display the reference of the product. 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 are going to link it with the Visual item of the Product data file.

4. In the "Linked item" area, expand "Analysis" then expand "Product". The list of items found in the Product data file is displayed.
5. Select the "Visual" 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 the "Product" file.
  - RTA\_Description control linked to the "Description" item of the "Product" file.
  - FSTC\_Price linked to the "PriceBT" item of the "Product" file.
  - STC\_Reference control linked to the "Reference" item of the "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 code of the page, we are going to enter the code used to:

- find the product to display.
  - display the data in the page.
- To display the data in the page:
    1. Press the [F2] key. The code editor displays the different processes of the page.
    2. In the "Global declarations" process, enter 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:

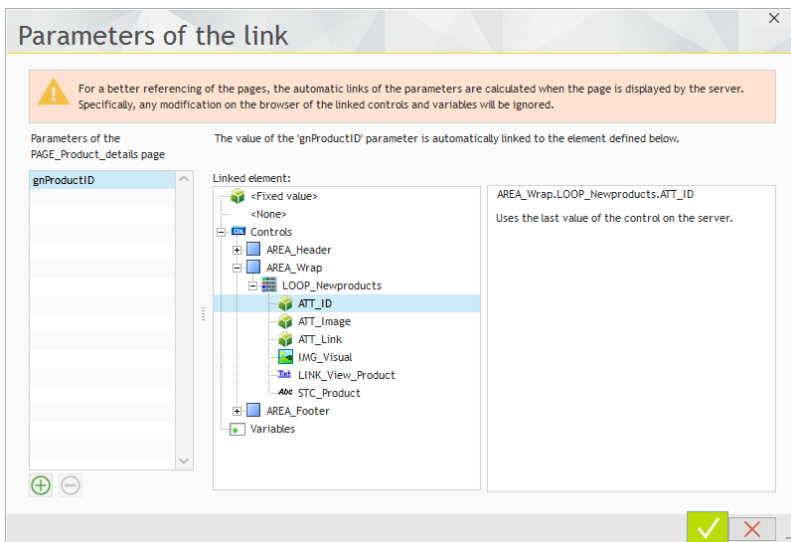
- **HReadSeekFirst** is used to find the first record of the Product file for which the ProductID item is equal to the gnProductID value, corresponding to the parameter passed to the page.

- **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

Now, let's see how to display the form of the product selected 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, we are going to modify the "PAGE\_List\_of\_new\_products" page in order for the link to open the description page of the product.
  1. Position on the "List of new products" page: click the "PAGE\_List\_of\_new\_products" button found in the button bar.
  2. Display the description window of the Link control: double-click the control.
  3. In the "General" tab of the 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 must be passed to the page in order to display the details of the selected product.



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





## Notes


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

7. Validate the description window of the Link control.

8. Save the modifications (  or CTRL S).

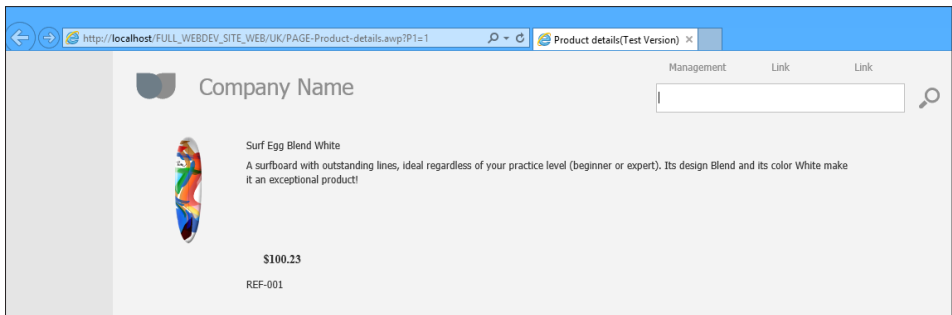
► We are going to run a test to check whether everything is operating.

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 details of a product.



► Close the pages to stop the test.

## Managing the styles

To improve our page, we are going to modify the styles of the 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. 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 of the input area.



Notes

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

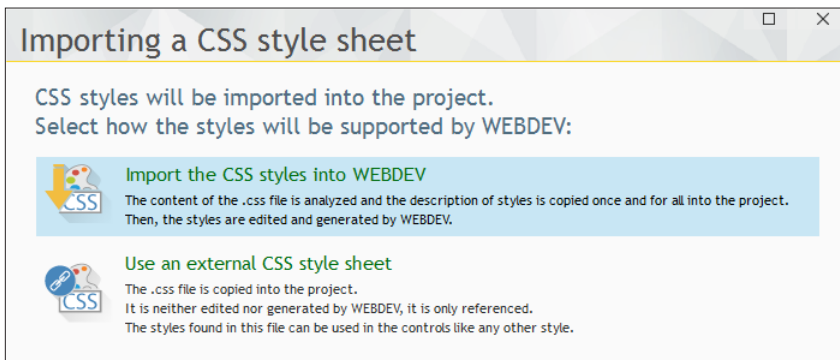
The skin provides the images, the fonts, the texture, the shape of buttons and the styles available for the project.

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.

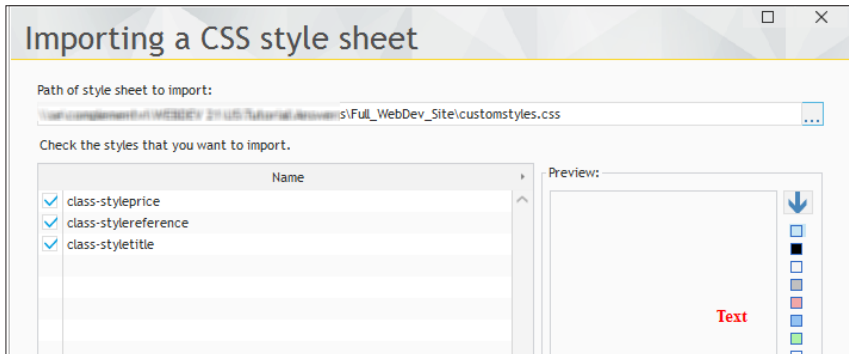
## Importing a CSS style sheet

► 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 (style book of the company).
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 different 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.

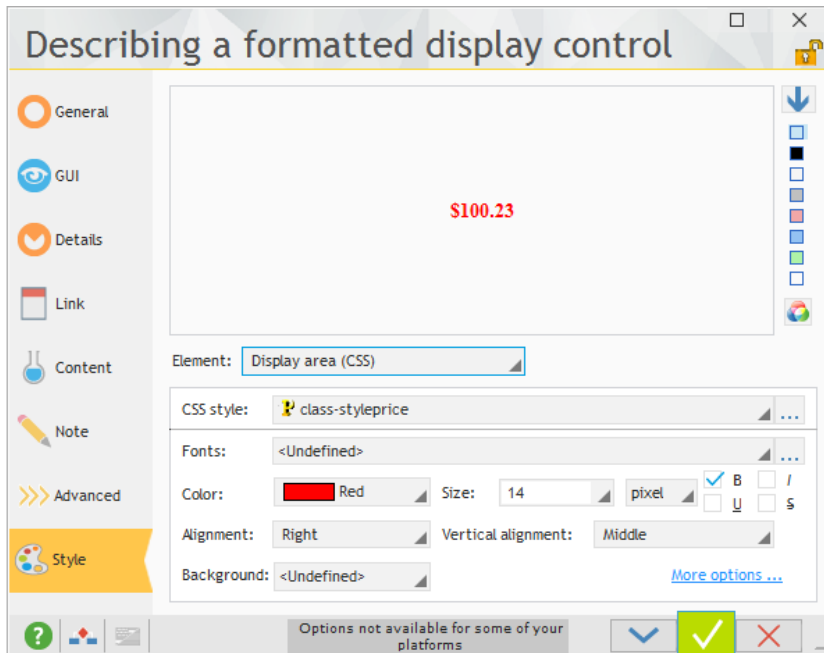
- The "class-" prefix means that it is a CSS class. This attribute means that this style will be applied to the elements with the HTML "class" attribute.


5. Keep all the proposed styles and validate. The styles are imported and they are available in WEBDEV.

## Apply the CSS styles


We are now going to apply the different styles to the controls found in the "PAGE\_Product\_details" page.

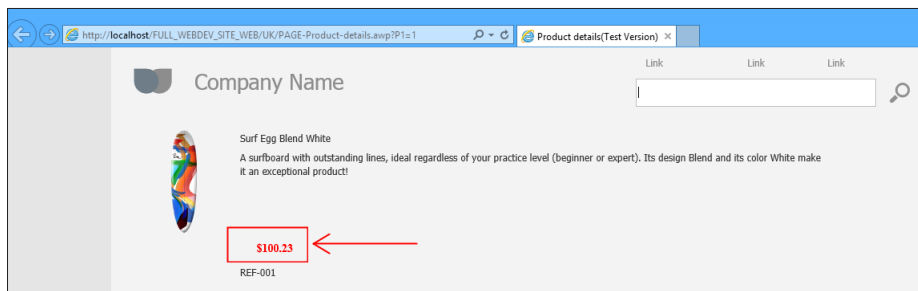
1. Position on the "PAGE\_Product\_details" page.
2. Display the description window of the "FSTC\_Price" control that displays the price of the product (double-click the control).
3. In the "Style" tab:
  - Select the "Display area (CSS)" element.
  - Select the "class-styleprice" style in the "CSS style"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



4. Validate. The 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 check whether everything is operating.

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 details of a product. 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 of 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.

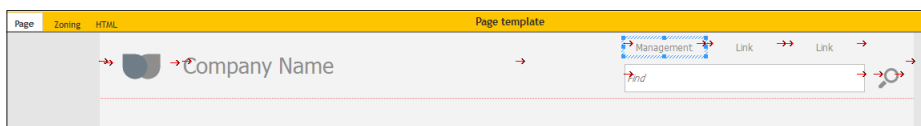
- To create a "Management" link:
1. Position on the "List of new products" page.
  2. Display the popup menu of one of the links found at the top of the page and select "Open the template".
  3. Display the description of the Link control (double-click the control).
  4. Modify the caption of the control: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

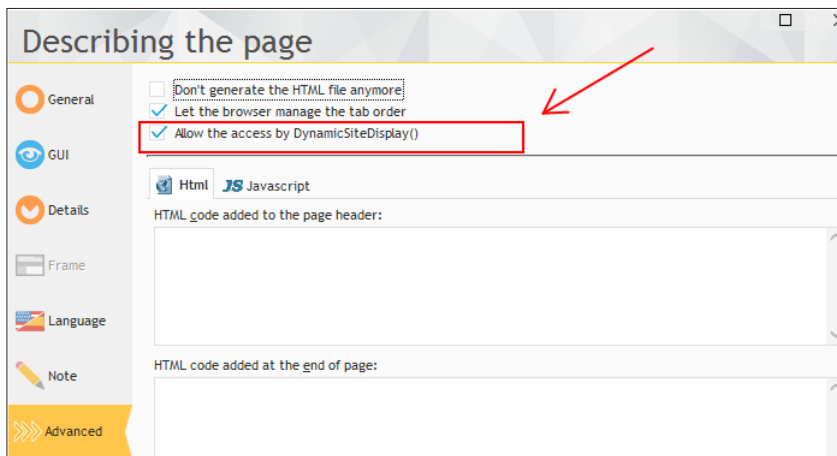
5. Select the operation on the controls: "None".
6. Validate.



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


```
DynamicSiteDisplay("Full_WebDev_Site")
```

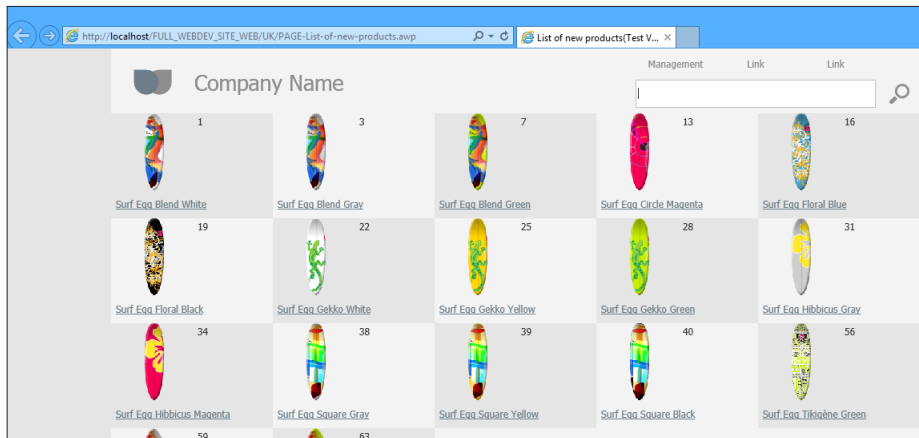
3. In this code, **DynamicSiteDisplay** is used to display the first dynamic page of the project (in our case, it will be the "PAGE\_List\_of\_Products" page).
  4. Close the code editor.
  5. In the orange bar of the template, click  to propagate the template modifications to all the pages that use this template.
  6. In our case, the two AWP pages of the project are proposed. Validate the window for updating the template.
  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 by DynamicSiteDisplay()".



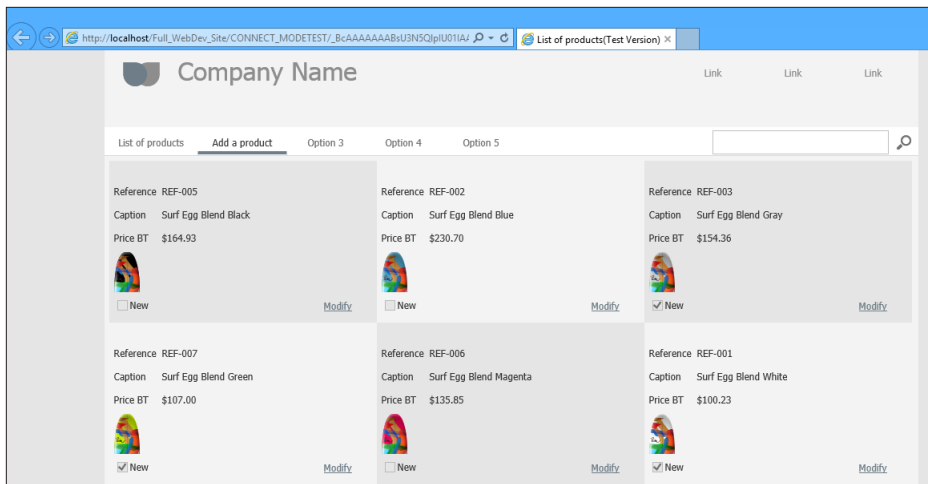
4. Validate.

► We are now going to run the test of 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 managing the products 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 used to display 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

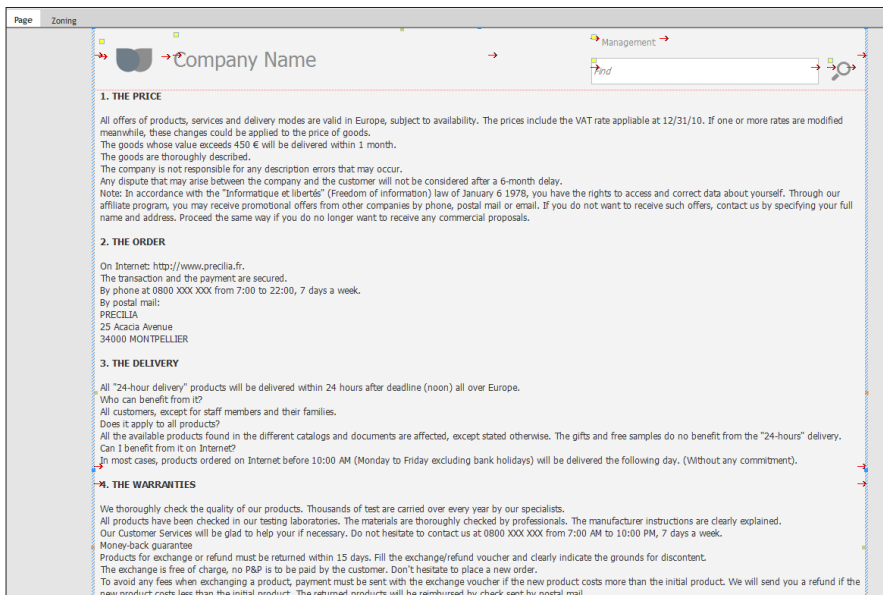
- ▶ To create a page used to display the general sales conditions:
  1. Create a new page.
    - Click  among the quick access buttons.
    - The window for creating a new element is displayed: click "Page" then "Page".
    - The wizard for page creation starts.
    - In the "Based on a project template" section, choose "PAGETPL\_Simple" and validate the wizard.
  2. Save the page (  or CTRL S). The title of this page being "General sales conditions", its name "PAGE\_General\_sales\_conditions" is automatically proposed. Validate the backup window.


### Creating the control

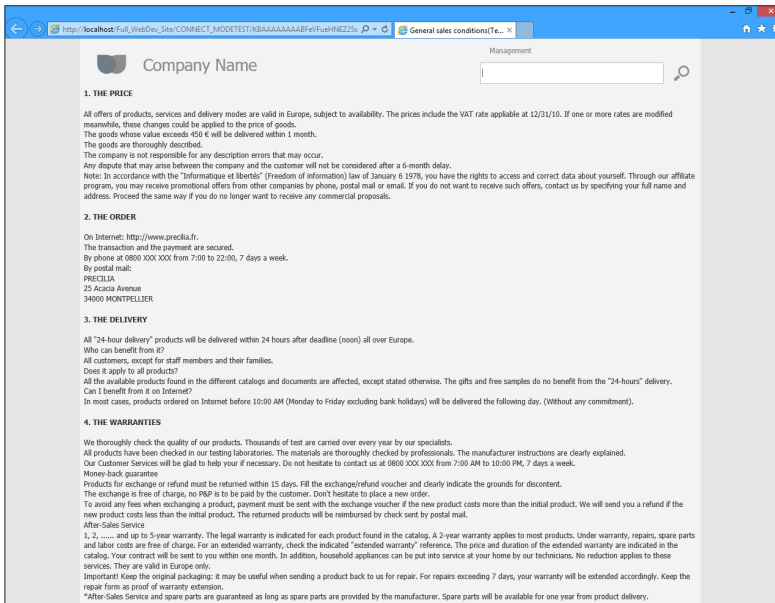
- ▶ To create a rich Static control:
  1. In the ribbon, on the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Rich Text Area".
  2. Click in the page (top left corner for example). The control is automatically created.



- 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 width of the control (with 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 the 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 appears.
  5. Perform some modifications in the text:
    - Add CR characters before and after the titles.
    - Switch the titles of the 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,
- Print,
- Management of multilingual feature, ...

DEVELOP 10 TIMES FASTER

## PART 5

Let's continue  
the development

21





## LESSON 5.1. MULTICRITERIA SEARCH

**This lesson will teach you the following concepts...**

---

- Creating a query with parameters.
- Creating the interface used to select 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

## 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 in order to display the home window.
  2. In the home window, click "Tutorial" and select "Full WebDev site (With pages)".



Answers

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

The interface of the "PAGE\_Multicriteria\_search" page will be as follows:

Company Name

Management

Find

Name:

Payment mode:

Find

Order status: ☒ All orders, ☐ Pending, ☐ Validated, ☐ Cancelled

Price range: \$119 - \$5 847

Search result

ID	Date	Order status	Total IOT	Customer	Payment mode
3	02/01/2010	Validated	727,07 €	TEISSIER Rosalie	Cash
6	03/01/2010	Validated	1 067,35 €	LOIRE Modestine	Cash
16	07/01/2010	Validated	317,30 €	LESCURE Cerise	Cash
24	11/01/2010	Validated	2 226,15 €	BEUCHER Gérald	Cash
25	11/01/2010	Pending	2 351,94 €	AUCLAIR Lothaire	Cash
28	11/01/2010	Validated	1 638,86 €	JAYET Honorine	Cash
44	16/01/2010	Validated	2 404,14 €	TRAV Anatole	Cash
64	23/01/2010	Validated	720,41 €	NAVET Doriane	Cash
70	25/01/2010	Validated	1 140,61 €	LENDORMAND Elise	Cash
76	27/01/2010	Validated	2 945,61 €	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 Koranza	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 result of the search. 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 results of the search.

The first step consists in creating the query used to select the records.



Notes


#### What is a select query?

A Select query is a query that will select the records corresponding to the specific criteria.

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

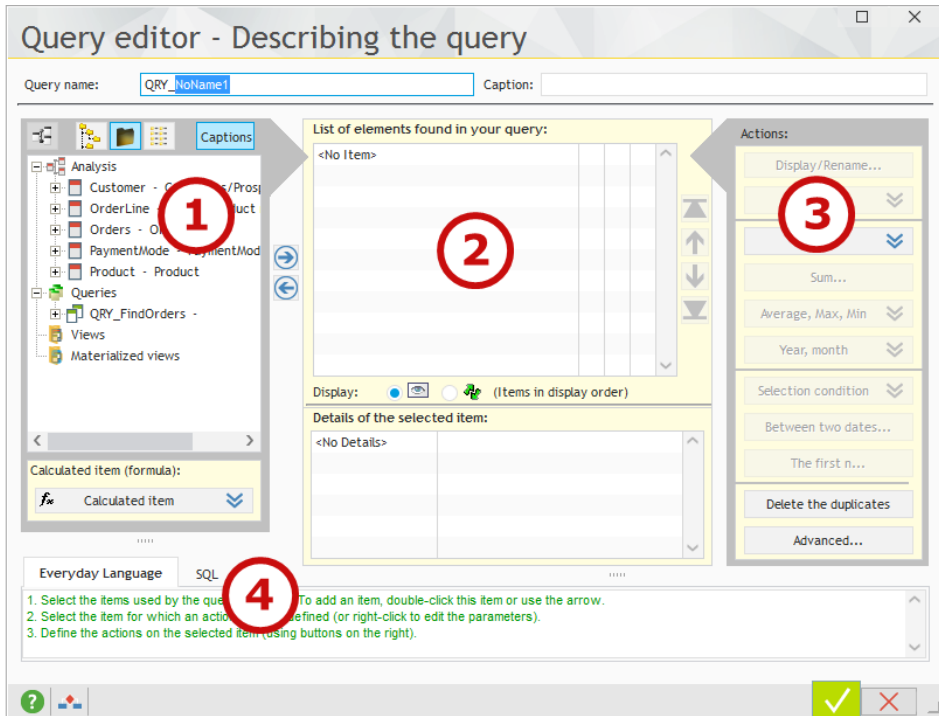
## Creating the query used to find the orders

### 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 "Select".  
Indeed, the query that will be created will allow us to select records. Go to the next step.
  3. The description window of the query is displayed.



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



This interface includes:

1. a reminder of the analysis (the description of the project 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.

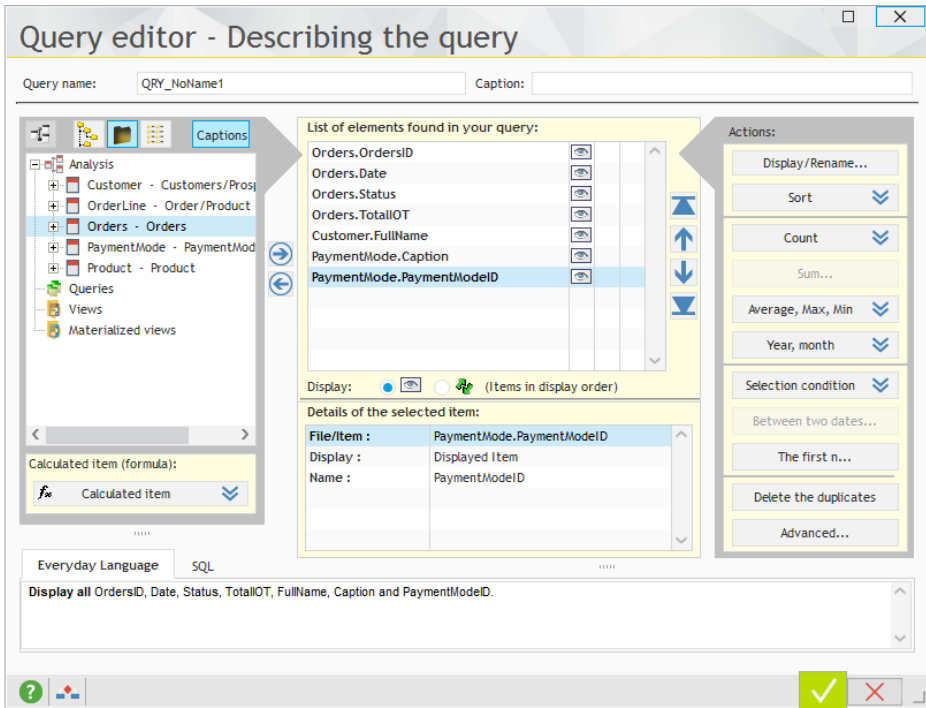
1. Double-click the items found in the analysis on the left of the description window. The items taken into account are displayed in the middle of the screen.

We want to display:

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

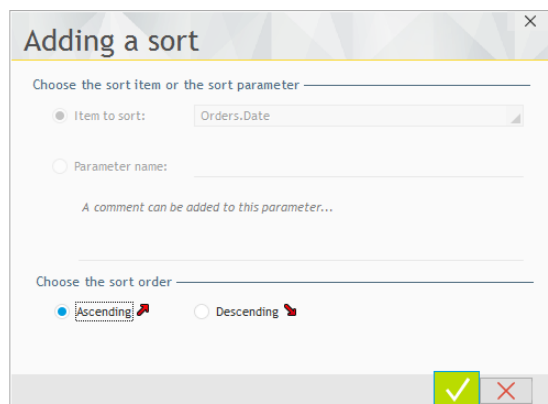


The description window of the query is as follows:



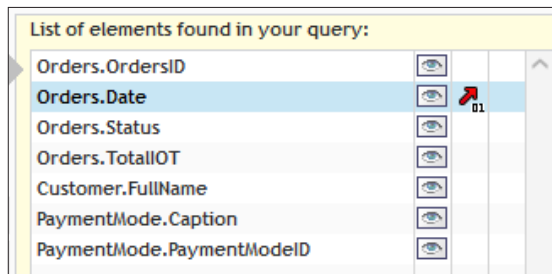
2. The data will be sorted by date.

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

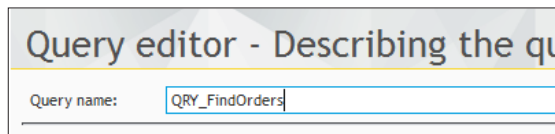


- In the window that is displayed, specify an ascending sort on the item and validate.

3. A red arrow with the number 01 appears on the right of the "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.

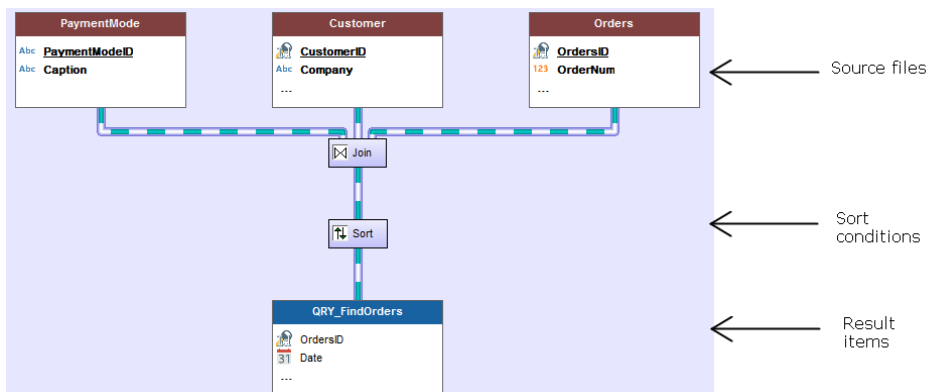


4. Give a name to the query: enter "QRY\_FindOrders" instead of "QRY\_NoName1" in the "Query name" area:



5. Validate the description window of the query (green button at the bottom of the screen).

6. The graphic representation of the query is displayed:




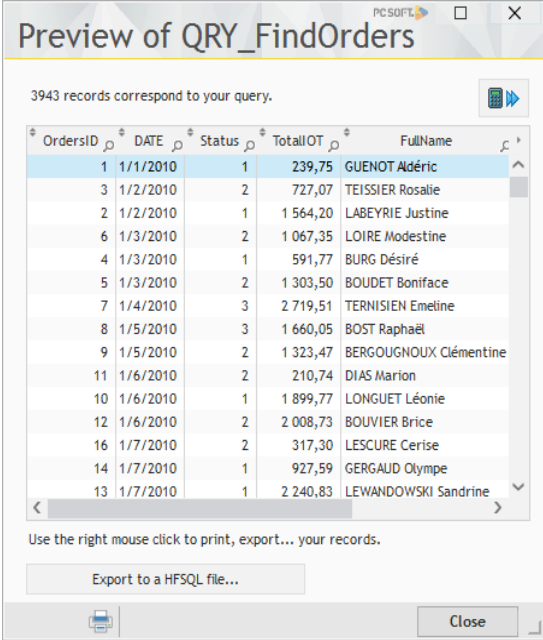
7. Save the query by clicking  among the quick access buttons.

8. Validate (if necessary) the backup window of the query.

## Test of the query

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:



3943 records correspond to your query.

OrdersID	DATE	Status	TotalOT	FullName
1	1/1/2010	1	239,75	GUENOT Aldéric
3	1/2/2010	2	727,07	TEISSIER Rosalie
2	1/2/2010	1	1 564,20	LABEYRIE Justine
6	1/3/2010	2	1 067,35	LOIRE Modestine
4	1/3/2010	1	591,77	BURG Désiré
5	1/3/2010	2	1 303,50	BOUDET Boniface
7	1/4/2010	3	2 719,51	TERNISIEEN Emeline
8	1/5/2010	3	1 660,05	BOST Raphaël
9	1/5/2010	2	1 323,47	BERGOUNOUX Clémentine
11	1/6/2010	2	210,74	DIAS Marion
10	1/6/2010	1	1 899,77	LONGUET Léonie
12	1/6/2010	2	2 008,73	BOUVIER Brice
16	1/7/2010	2	317,30	LESCURE Cerise
14	1/7/2010	1	927,59	GERGAUD Olympe
13	1/7/2010	1	2 240,83	LEWANDOWSKI Sandrine

Use the right mouse click to print, export... your records.

Export to a HFSQL file...

Close



Notes

A popup menu is displayed when a right click is performed on the result of the query. You have the ability to export the result to:

- an XLS file (Excel).
- an XML file (eXtensible Markup Language).
- a text 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.

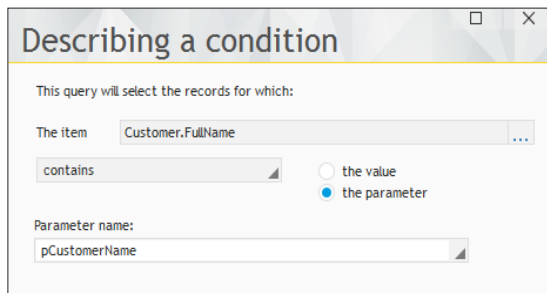
## Using parameters in the query

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

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

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

- ▶ To define the query's parameters, display the description window of the query: double-click the background of the graphic representation of the query (or use the popup menu, "Query description" option).
- ▶ 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 name of the parameter: "pCustomerName".



Notes

We advise you prefix the parameters of queries with the letter "p". Therefore, you will be able to easily find them in the code editor.

When searching for a query parameter, all you have to do is type 'p' and the completion feature of the code editor will propose all the parameters.

4. Validate the description window of the condition. The number "1" appears on the right of the "Customer.FullName" item, indicating that a selection condition was defined.

List of elements found in your query:

Orders.OrdersID		
Orders.Date		01
Orders.Status		
Orders.TotalIOT		
Customer.FullName		1
PaymentMode.Caption		
PaymentMode.PaymentModelID		



Notes

The query description in natural language (at the bottom of the query editor) is automatically updated according to the added condition.

Everyday Language SQL

Display all OrdersID, Date, Status, TotalIOT, FullName, Caption and PaymentModelID as FullName contains 'pCustomerName'. The result is sorted on value of Date in ascending order.

- Let's take a look at 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:

Describing a condition

This query will select the records for which:

The item: Orders.Status


is equal to

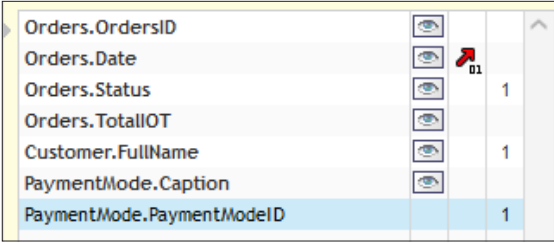
☐ the value  
☒ the parameter  
☐ all the values of (ALL)  
☐ any value of (ANY)

Parameter name:

pStatus

- Select "Is equal to".
  - Select "the parameter".
  - Specify the name of the parameter: "pStatus".
4. Validate the description window of the condition. The number "1" appears on the right of the "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 name of the parameter: "pPaymentModelID".
  3. Validate the definition of the selection condition.



Orders.OrdersID			
Orders.Date			01
Orders.Status			1
Orders.TotalIOT			
Customer.FullName			1
PaymentMode.Caption			
PaymentMode.PaymentModelID			1

- The last selection condition to define affects the amount of the order. 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 the 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 "the parameter".
    - Specify the name of the parameter, "pMinPrice".
  4. Validate the definition of the selection condition.
  5. Define another condition on the same "Orders.TotalIOT" item: display the popup menu of the item (right click) and select "Selection condition .. New condition".
  6. In the window that is displayed:
    - Select the condition "Is less than or equal to".
    - Click "the parameter".
    - Specify the name of the parameter, "pMaxPrice".
  7. Validate the definition of the selection condition.

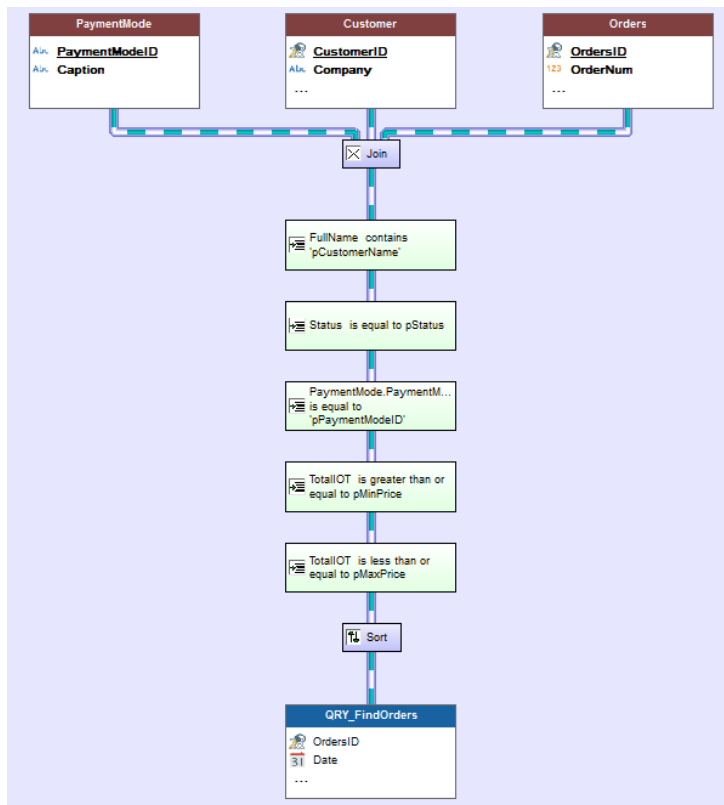


## Notes

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, the use of a selection condition such as "Is included between" means that the two values must be entered (which is not the case if two different conditions are defined).


8. Validate the description window of the 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 the query with parameters

► To run the test of the query with parameters:

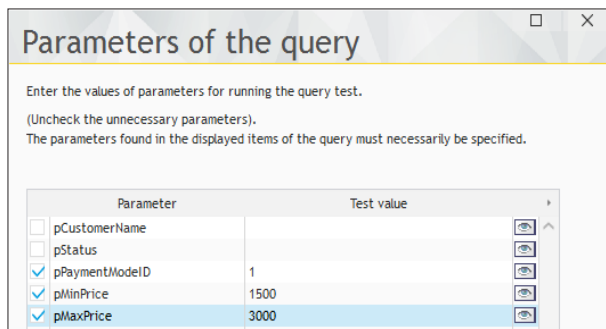
1. Click .
2. A window is displayed, allowing you to enter the different parameters of the query.



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 of all the customers will be taken into account by the query.

Note: If all the parameters are unchecked, the query returns all the results.



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. Enter 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

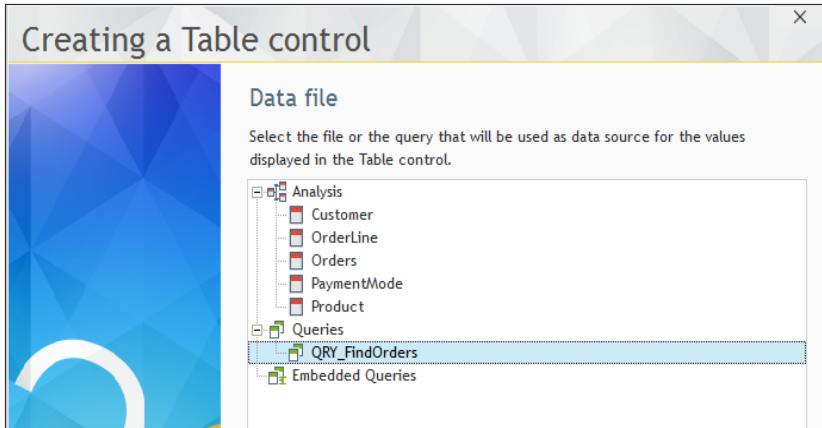
- ▶ To create a page used to display the result of the multicriteria search:
  1. Create a new blank page.
    - Click  among the quick access buttons.
    - The window for creating a new element is displayed: click "Page" then "Page".
    - The wizard for page creation starts.
    - In the "Based on a project template" section, choose "PAGETPL\_Simple" and validate the wizard.
  2. Save the page (  or CTRL S). The title of this page being "Multicriteria search", its name "PAGE\_Multicriteria\_search" is automatically proposed. Validate the backup window.

### Creating the controls used to configure the criteria and to display the result

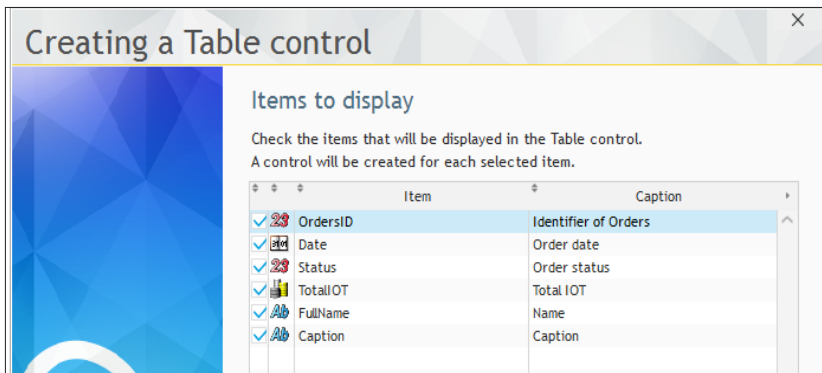
We are going to create a Table control based on the query then the different controls allowing the user to select the search criteria.

#### Creating the Table control

- ▶ To create the Table control used to display the result of the search:
  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.



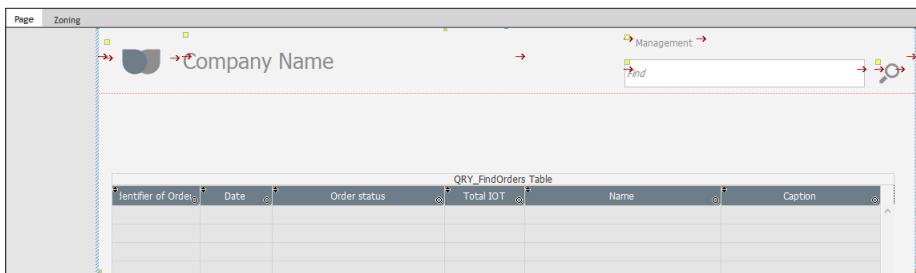
5. Select all the proposed items: all of them will be displayed in the Table control. Go to the next wizard step.



6. Keep the default options in the different wizard screens and validate the creation of the Table control.

7. The Table control is automatically created in the page.

8. Modify (if necessary) the position of the Table control so that it is entirely displayed in the page.



- For better readability, we are going to modify the description of the Table control.
1. Display the description of the Table control (double-click the control).



Notes

The description window of a Table control includes two sections:

- the upper section, presenting the name of the control, the name of columns as well as their type.
- the lower section, containing the different description tabs.

If the name of the Table control is selected, the lower section presents the characteristics of the Table control.

If a column is selected, the lower section presents the characteristics of the 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 title of the column is displayed in the lower section of the screen. Replace the "Identifier of the order" caption by "ID".

### Describing a table control

Table name:  ← → New Insert Delete

1/6	2/6	3/6	4/6
Name: COL_OrdersID	COL_Date	COL_Status	COL_TotalOT
Type: Numeric	Date	List of Values	Currency

General Title: English (3)

GUI 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 the Table control. The control is automatically updated with the modifications performed.
7. Select the Table control and enlarge the control to the right with the sizing handles (blue squares located on the selection rectangle).
8. Reduce the size of the "ID" column in order for all the columns to be displayed in the Table control.
9. Increase the size of the Customer and PaymentMode columns in the Table control.


Page Zoning

Company Name

Management

Search results

ID	Date	Order status	Total IDT	Name	Caption

10. To save the page, click  among the quick access buttons. We are going to check the sizes of columns by running the page.




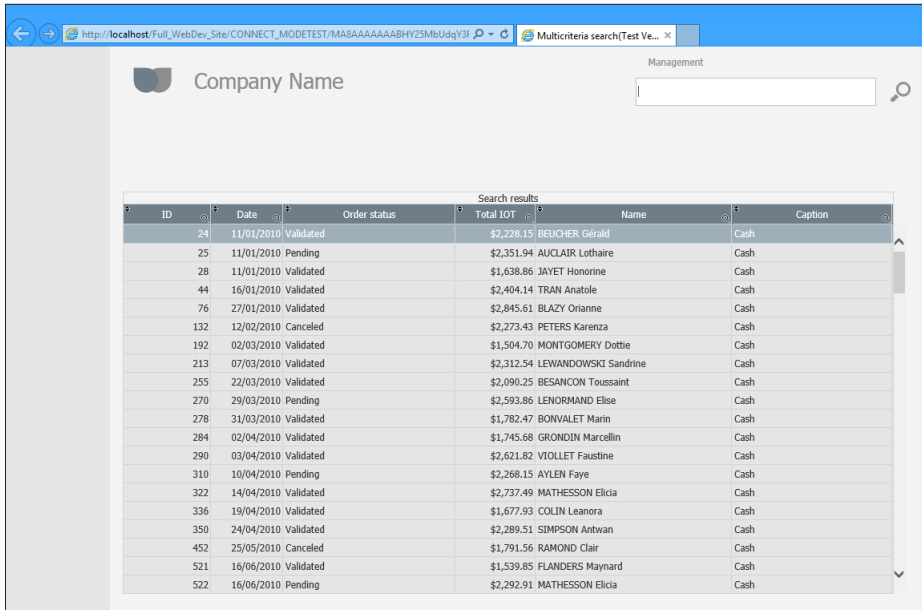
Notes

#### 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 displayed data depends on the result of the query and it can only be known at run time.

► We are going to run a first test of this page:

1. Click  among the quick access buttons.
2. Only the orders paid in cash and whose price is included between \$1500 and \$3000 are displayed, like in the last test of the query run in the editor, when parameters were specified in the test window of the query.



ID	Date	Order status	Total TOT	Name	Caption
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	JAYET Honorine	Cash
44	16/01/2010	Validated	\$2,404.14	TRAN Anatole	Cash
76	27/01/2010	Validated	\$2,845.61	BLAZY Orianne	Cash
132	12/02/2010	Canceled	\$2,273.43	PETERS Karenza	Cash
192	02/03/2010	Validated	\$1,504.70	MONTGOMERY Dottie	Cash
213	07/03/2010	Validated	\$2,312.54	LEWANDOWSKI Sandrine	Cash
255	22/03/2010	Validated	\$2,090.25	BESANCON Toussaint	Cash
270	29/03/2010	Pending	\$2,593.86	LENORMAND Elise	Cash
278	31/03/2010	Validated	\$1,782.47	BONVALET Marin	Cash
284	02/04/2010	Validated	\$1,745.68	GRONDIN Marcellin	Cash
290	03/04/2010	Validated	\$2,621.82	VIOLETT Faustine	Cash
310	10/04/2010	Pending	\$2,268.15	AYLEN Faye	Cash
322	14/04/2010	Validated	\$2,737.49	MATHESSON Elicia	Cash
336	19/04/2010	Validated	\$1,677.93	COLIN Leanora	Cash
350	24/04/2010	Validated	\$2,289.51	SIMPSON Antwan	Cash
452	25/05/2010	Canceled	\$1,791.56	RAMOND Clair	Cash
521	16/06/2010	Validated	\$1,539.85	FLANDERS Maynard	Cash
522	16/06/2010	Pending	\$2,292.91	MATHESSON Elicia	Cash

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.
- Press [CTRL] + [/ on the numeric keypad].


4. Close the code editor (click the cross at the top right of the editor).

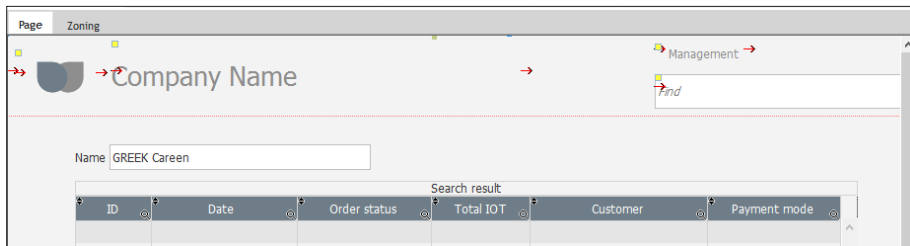
We are now going to create in our page the different controls allowing the user to select the query parameters. These controls will be positioned above the Table control.

- Move (if necessary) the Table control in the page and reduce its height in order to leave space for creating the different control for criteria selection.

### First parameter: Customer name

In order for the user to be able to enter 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 the  icon on the left of "Customer" file: the items found in the data file are listed.
  3. Select the "FullName" item found in 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 entered 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 the Table control, replace the line:


```
//MySource.pCustomerName = <Value of pCustomerName parameter>
```

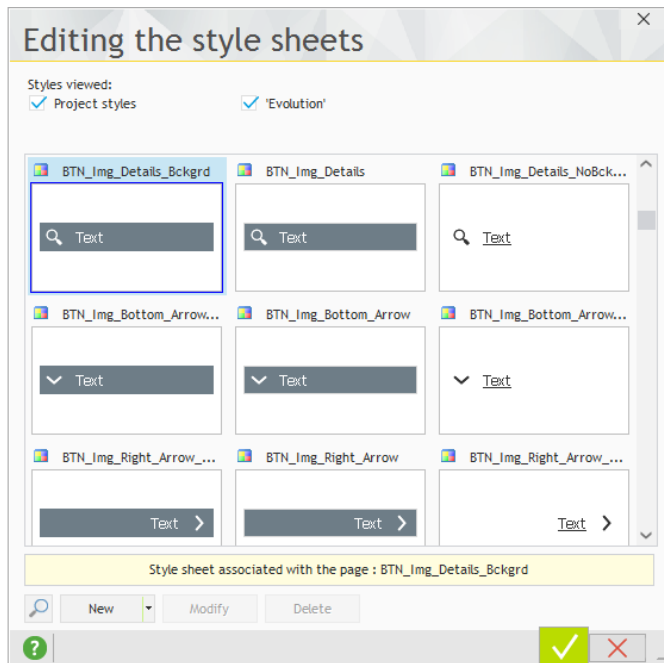
by

```
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 the 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 at the top right of the page.
  2. Modify the caption of the control (press the [ENTER] key). The new caption is "Find".
  3. Modify the style of the control:
    - In the popup menu of the control (right mouse click), select "Choose a WEBDEV style".
    - In the window that is displayed, select "BTN\_Img\_Details\_Bckgrd" and validate.





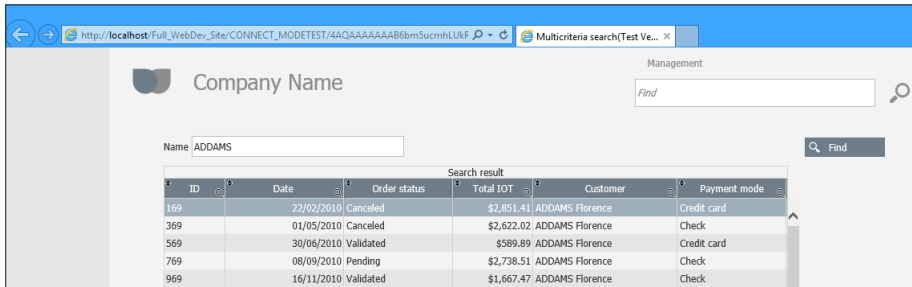
4. Resize the control if necessary:
  - select the control.
  - in the popup menu, select "Resize".
5. Display the code associated with this control: press the [F2] key.
6. Enter the following code in the "Click (server)" process:

```
// Refreshes the display of the Table control
TableDisplay(TABLE_QRY_FindOrders, taInit)
```

In this code, the **taInit** constant is used to re-run the initialization process of the 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, enter the customer name ("ADDAMS") and click the "Find" button. The content of the Table control is modified: all the orders passed by the customer named "ADDAMS" are displayed.



The screenshot shows a web browser window with the URL `http://localhost/Full_WebDev_Site/CONNECT_MODETEST/4AQAAAAAAB6bmSucmHLUKF`. The page has a header with a logo and the text "Company Name". Below the header, there is a search bar with the text "Name ADDAMS" and a "Find" button. The search results are displayed in a table with the following data:

ID	Date	Order status	Total IOT	Customer	Payment mode
169	22/02/2010	Canceled	\$2,851.41	ADDAMS Florence	Credit card
369	01/05/2010	Canceled	\$2,622.02	ADDAMS Florence	Check
569	30/06/2010	Validated	\$589.89	ADDAMS Florence	Credit card
769	08/09/2010	Pending	\$2,738.51	ADDAMS Florence	Check
969	16/11/2010	Validated	\$1,667.47	ADDAMS Florence	Check

4. Close the test page.

### Second parameter: Order status

Three states can be assigned to an order:

- pending,
- paid,
- canceled.

In our analysis, the status of the order is saved in the "Status" item found in the "Orders" data 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 of the "Orders" data file.




Notes

The radio buttons are also called "option boxes". They are used to select a single option among the proposed ones.

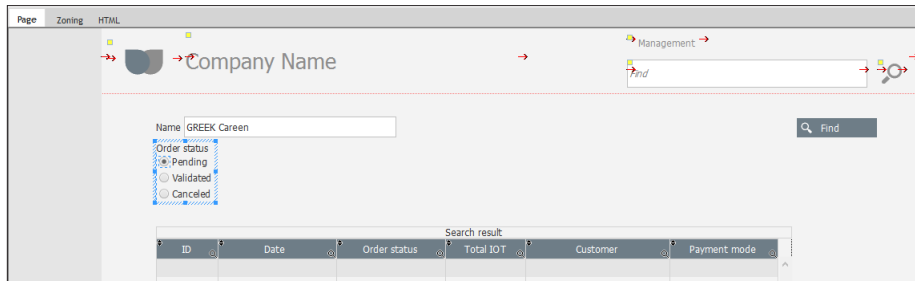
#### How to differentiate between a radio button and a check box?

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 the "Orders" file: the items found in the data file are listed.
  2. Select the "Status" item found in the Orders file and Drag and Drop this item to 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.
- Display the popup menu (right mouse click) and select "Code".

2. In the initialization process of the Table control, replace the line:

```
//MySource.pStatus = <Value of pStatus parameter>
```


by

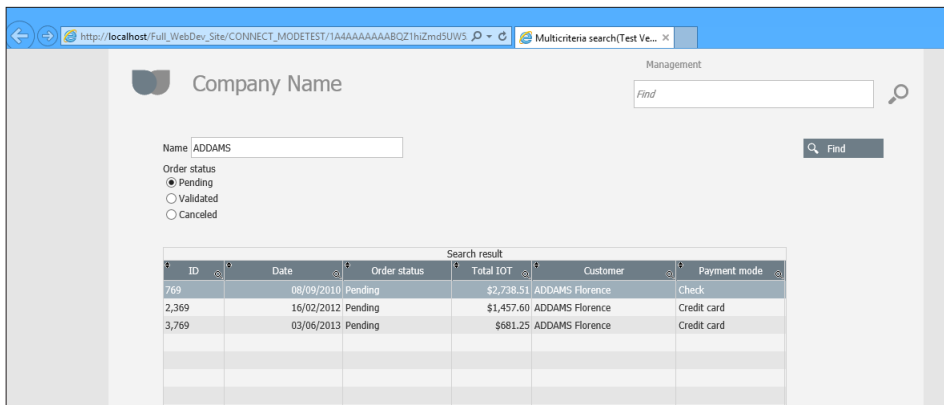
```
MySource.pStatus = RADIO_Status
```

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. Enter the name "ADDAMS" and select "Pending".
3. Click the "Find" button: only the orders placed by the customer named ADDAMS and waiting for payment are listed.





2. In the initialization process of the 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.

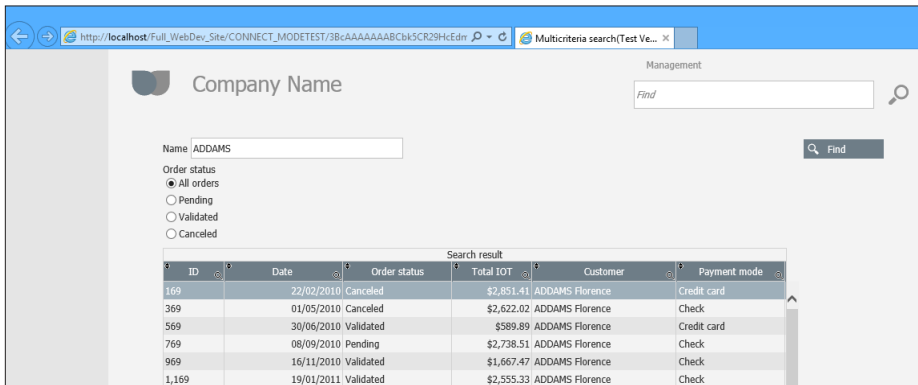
► Let's run the test of our page.

1. To save the page, click  among the quick access buttons.

2. Click  among the quick access buttons.

3. Enter the name "ADDAMS" and select "All orders".

4. Click the "Find" button: only the orders placed by the customer named ADDAMS are listed, regardless of their status.



The screenshot shows a web browser window with the URL `http://localhost:Full_WebDev_Site/CONNECT_MODETEST/38cAAAAAABCbK5CR29HcEdm`. The page title is "Company Name". There is a "Management" section with a "Find" button. Below this, there is a "Name" field containing "ADDAMS" and a "Find" button. Under "Order status", the "All orders" radio button is selected. The "Search result" table displays the following data:

ID	Date	Order status	Total IOT	Customer	Payment mode
169	22/02/2010	Canceled	\$2,851.41	ADDAMS Florence	Credit card
369	01/05/2010	Canceled	\$2,622.02	ADDAMS Florence	Check
569	30/06/2010	Validated	\$589.89	ADDAMS Florence	Credit card
769	08/09/2010	Pending	\$2,738.51	ADDAMS Florence	Check
969	16/11/2010	Validated	\$1,667.47	ADDAMS Florence	Check
1,169	19/01/2011	Validated	\$2,555.33	ADDAMS Florence	Check

5. Close the browser.

### Third parameter: Payment mode

Several payment modes can be used for an order: cash, check, ... The available payment modes are stored in the "PaymentMode" data file.

We will be using a Combo Box control based on this data file to allow the user to select the requested payment mode.



Notes

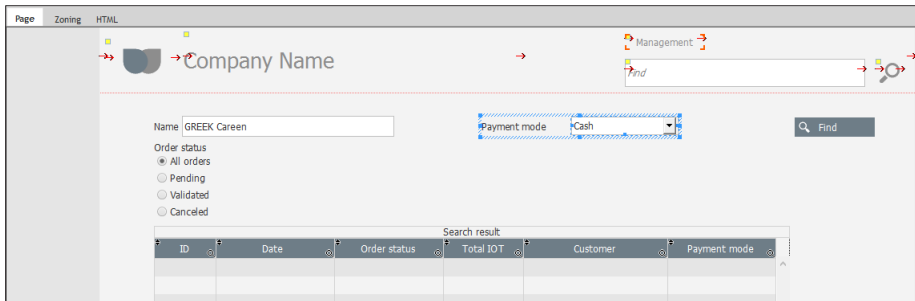
The "Combo Box" control is used to display a list of elements and to select an element from this list.

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 a query" and go to the next step.
  5. Select the "PaymentMode" data file and go to the next step.
  6. The item that must be displayed in the Combo Box control is "Caption". Select "Caption". Go to the next step.
  7. Select the sort item: "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 the 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 the Table control, replace the line:

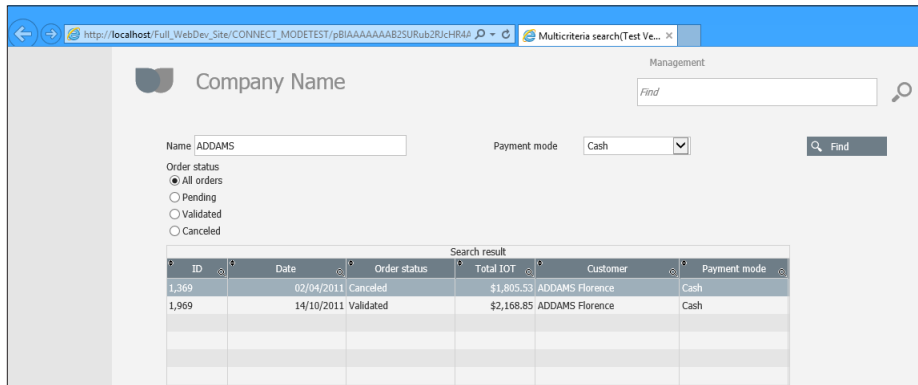
```
//MySource.pPaymentModeID = 1
```

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 the Table control is modified.



4. Close the browser.

### Last parameter: Order amount

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 enter a price interval. To do so, we will be using a Range Slider control.

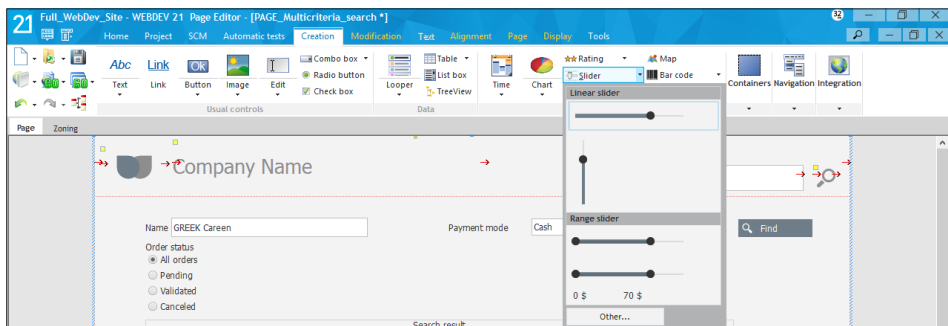


Notes

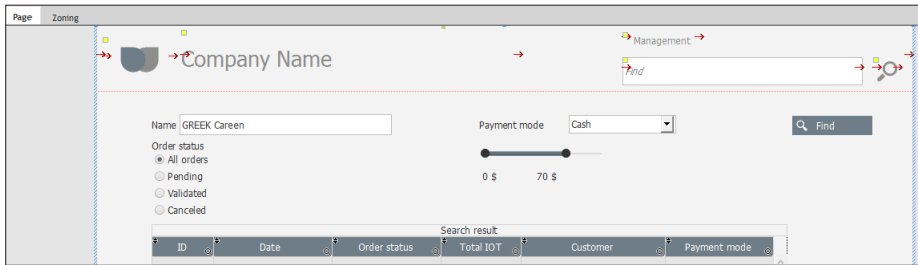
A Range Slider control is a graphic control that can be easily included in a WEBDEV site and that is used to select the bounds of an interval (start value and end value).

► To manage the price of the product:

1. On the "Creation" pane, in the "Graphic controls" group, expand "Slider" and select the range slider with prices.



2. Click in 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 data 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 the 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 the "Orders" file according to the search key, TotalIOT in this case. Therefore, this function will be used to read the record corresponding to the lowest amount.
  - Then, the amount read in the record is associated with the lower bound of the Slider control as well as its lower 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 highest amount. The only difference is the name of the function used: **HReadLast**, that will be used to read the last record of the Orders file according to the amount, which means corresponding to the highest 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 the Table control, replace the lines:

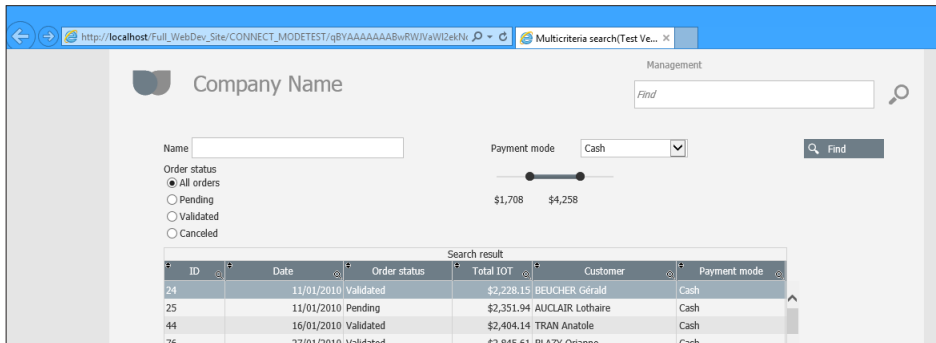
```
//MySource.pMinPrice = "1500"
//MySource.pMaxPrice = "3000"
```

by:

```
MySource.pMinPrice = RSLI_NoName2..LowerValue
MySource.pMaxPrice = RSLI_NoName2..UpperValue
```

In this code, the parameters corresponding to the prices are equal to the lower value and upper value of the 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 parameters of the search:
    - Customer name,
    - Status,
    - Payment mode,
    - Amount.
  3. Click the "Find" button. The content of the Table control is modified.



The screenshot shows a web application interface for 'Company Name' management. At the top, there is a search bar with the placeholder text 'Find' and a magnifying glass icon. Below the search bar, there are several input fields and a 'Find' button. The 'Name' field is empty. The 'Payment mode' dropdown is set to 'Cash'. There is a slider for 'Amount' with values \$1,708 and \$4,258. The 'Order status' section has radio buttons for 'All orders' (selected), 'Pending', 'Validated', and 'Canceled'. Below these, there is a table titled 'Search result' with the following columns: ID, Date, Order status, Total IOT, Customer, and Payment mode. The table contains four rows of 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
76	27/01/2010	Validated	\$2,845.61	BLAZY, Oriane	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.



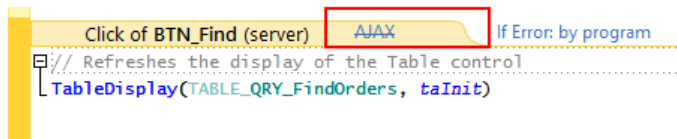
Notes

WEBDEV proposes several methods for using the Ajax technology in a site:

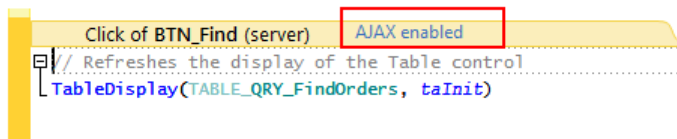
- **Ajax in one click:** all you have to do is click in the code editor to transform a process into an Ajax process.
- **Ajax controls** (Table control, Looper control). **The Ajax controls are used to display the data dynamically loaded from the server.** No need to download everything at once!
- **Ajax by programming** with 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 the button, the AJAX link appears to be struck out in the code bar.




3. Click the AJAX link: the "JAVA enabled" link appears, indicating that the process is automatically transformed into an Ajax process.



4. Close the code editor.

► Run the test of the page:

1. Click  among the quick access buttons.
2. Define the different parameters of the search:
  - Customer name,
  - Status,
  - Payment mode,
  - Amount.
3. Click the "Find" button. Only the content of the 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 min

## Overview

We are going to give the user the ability to directly print the invoice corresponding to the sought order.



### Answers

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) then 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. But 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 the sales department.

## Generating documents (HTML, PDF, XML, ...)

Printing from HTML, PDF or XML documents consists in creating a document and in displaying it on the browser of the 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 (to transmit an order form in PDF format to the Web user, ...).

## What is a report made of?

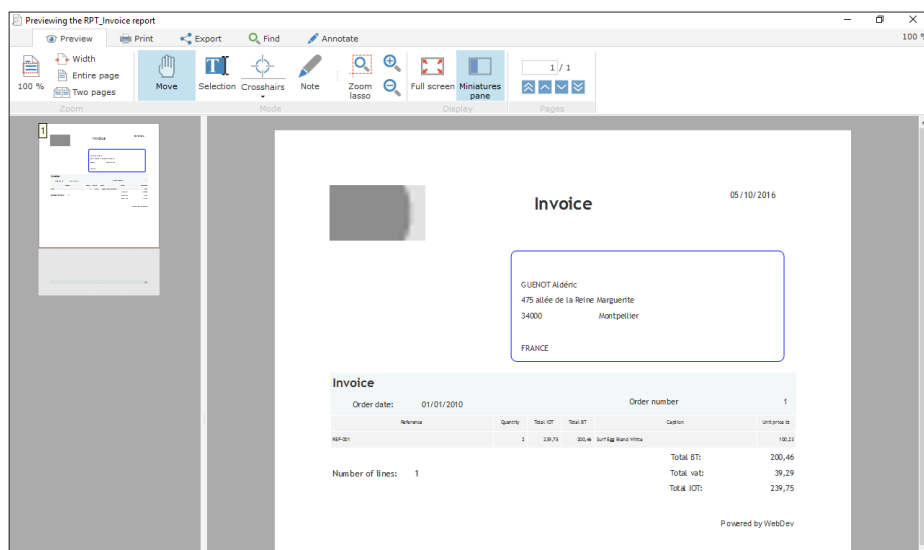
In our example, the invoice that must be printed will correspond to a report. A report is a project element used to group and format the data to print.

To create and use the 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 get:



## Creating the "Invoice" report

First of all, we are going to list the information that must be displayed in the report:

- The characteristics of the order: date and order number.
- The personal details of the customer: name, address, zip code, city and country.
- The characteristics of the 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.

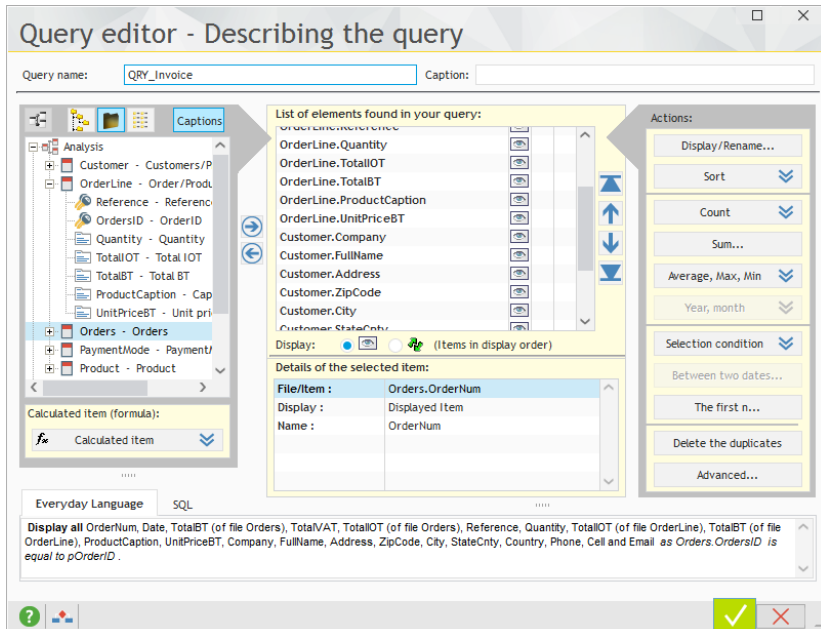
The reports based on data files must be simple reports, which means reports used to display data coming from a single data file.

## 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 "Select".

Indeed, this query will be used to select the records that will be printed in the report. Go to the next step.
  3. The description window of the query is displayed.
- ▶ First, 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 the Orders file, the content of the OrderLine file and the content of the 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 the query is as follows:



At this time, this query is used to select all the orders and the corresponding order lines. 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 area that lists the query elements includes 4 columns:

- The name of the item,
- The ability to display (or not) the item in the query result,
- The management of the sort 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 name of the parameter: "pOrderID".

4. Validate the description window of the condition. The number "1" is displayed on the right of the Orders.OrdersID item, indicating that a selection condition was defined.

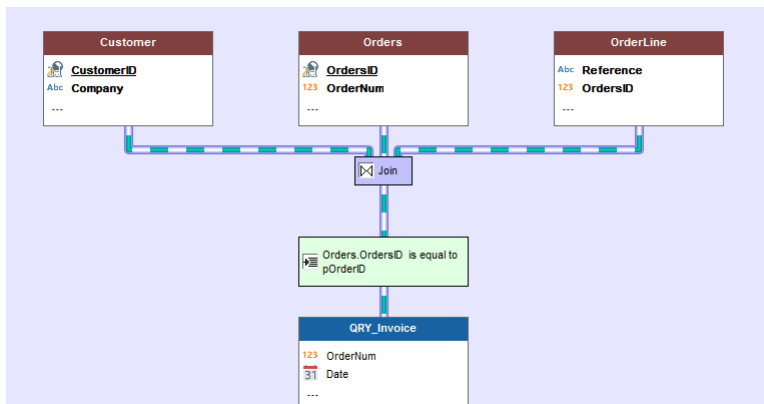
5. The identifier of the order will not be viewed so let's make it invisible: click the eye found on the line of the item and select "Don't display".


6. Similarly, make the following items invisible:

- Order.Status,
- Orders.CustomerID,
- Orders.PaymentModelID,
- OrderLine.OrdersID,
- Customer.CustomerID.


7. Validate the description window of the query (green button at the bottom of the screen).

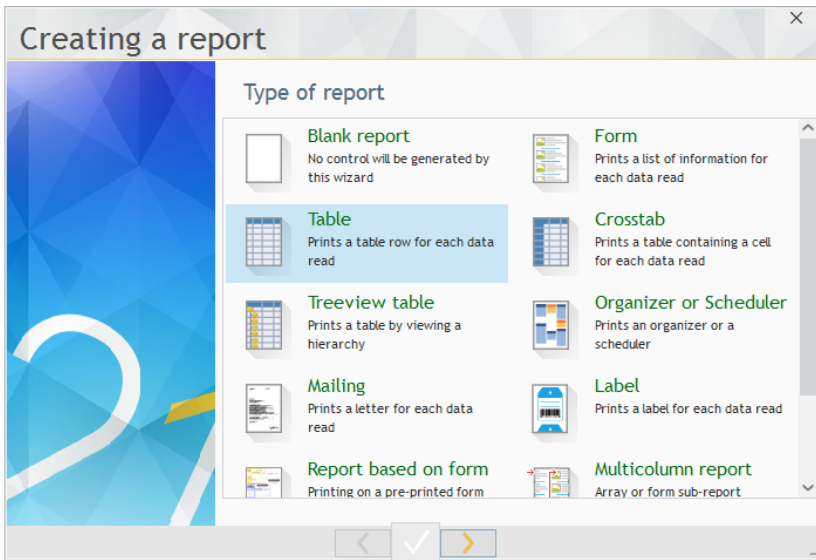
8. The graphic representation of the query is displayed:



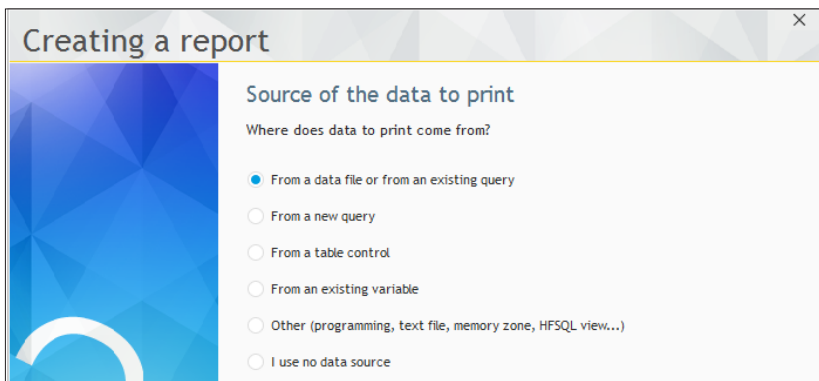
9. Save the query by clicking  among the quick access buttons. Validate the backup information if necessary.

## 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 the 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.



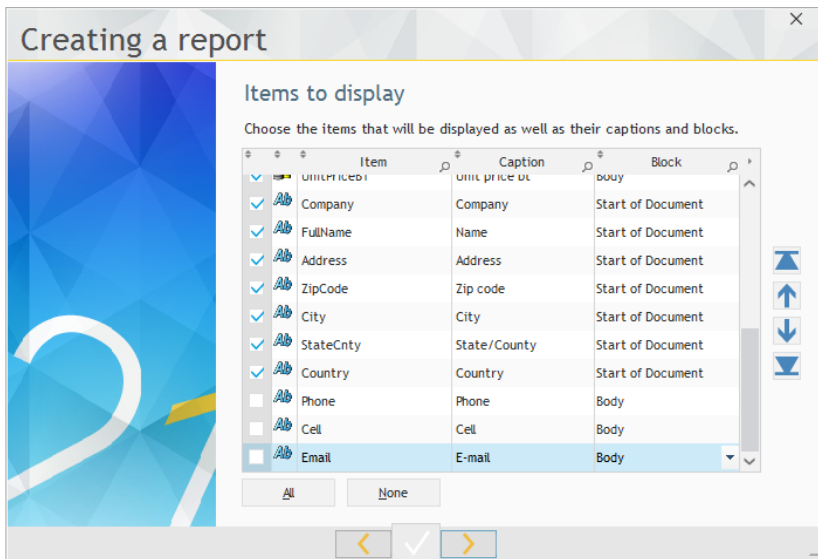
6. In the list of data files and queries, select the "QRY\_Invoice" query. Go to the next step.



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 line of the order.
- The items regarding the order will be displayed in the "Page header" block. Indeed, this information must not be repeated on each line of the order.
- The items regarding the order lines will be displayed in the body of the report. These items will be displayed on all the order lines of the 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:

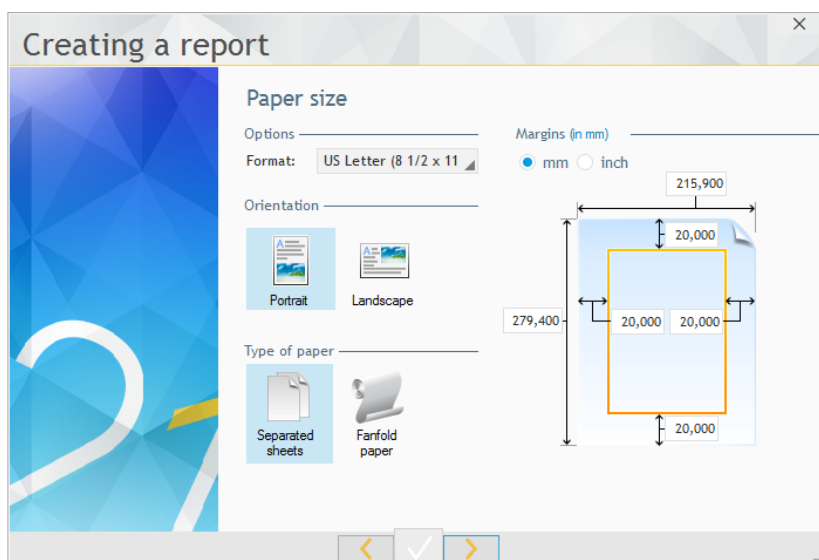
Item	Block
OrderNum	Start of document
Date	Page header
TotalBT	End of document
TotalIOT	End of document
Reference	Body
Quantity	Body
TotalIOT_Ca	Body
TotalBT_Ca	Body

ProductCaption	Body
UnitPriceBT	Body
Company	Start of document
FullName	Start of document
Address	Start of document
ZipCode	Start of document
City	Start of document
StateCnty	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 layout of the report.



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 the printer. The physical margins of the printer are margins where no print is allowed. Furthermore, the physical margins differ according to 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. All we have to do now is give a name and a title to the report.

- Enter the title: "Invoice".
- Enter the name: "RPT\_Invoice".

Creating a report

It's done!

Give a name and a title to the report.

Name:

Title:


15. Validate.


16. The table as it is defined does not fit on an A4 page in portrait mode. WEBDEV proposes:


- to print the report on 2 pages in width,
- to switch to landscape mode,
- to reduce the size of the table.

Choosing the reduction

The table exceeds the page width:

 Print the table on 2 pages

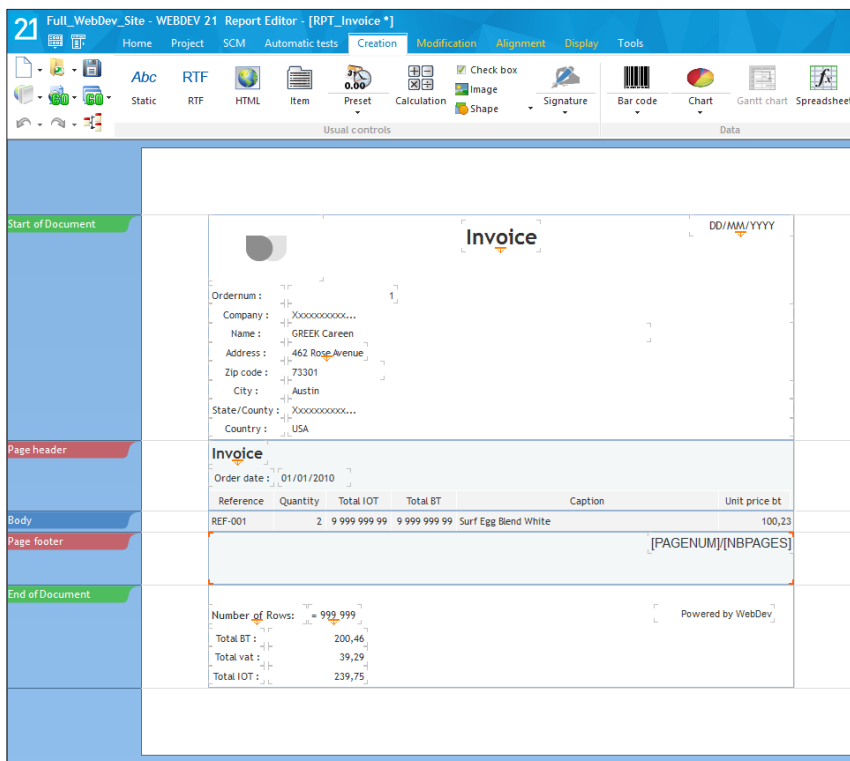
 Use the "landscape" mode

 Reduce the table to 98 % of the source one

☒ ☐

Choose "Reduce the table".

17. The report is displayed in edit in the report editor:



18. The different lines of the order are grouped in table format.

19. Save the report (  or CTRL S). Validate the backup information if necessary.

► 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 a screen in a specific window.

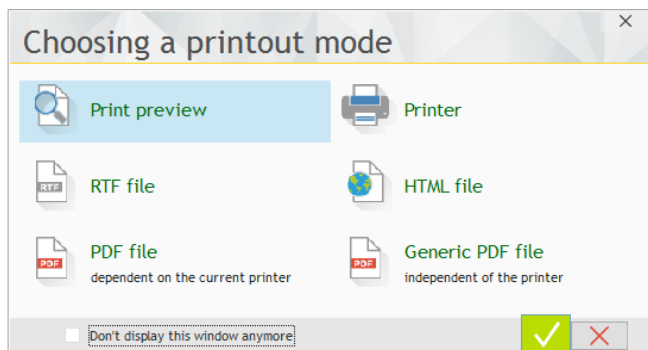


Notes

The print preview is available when developing the site only.

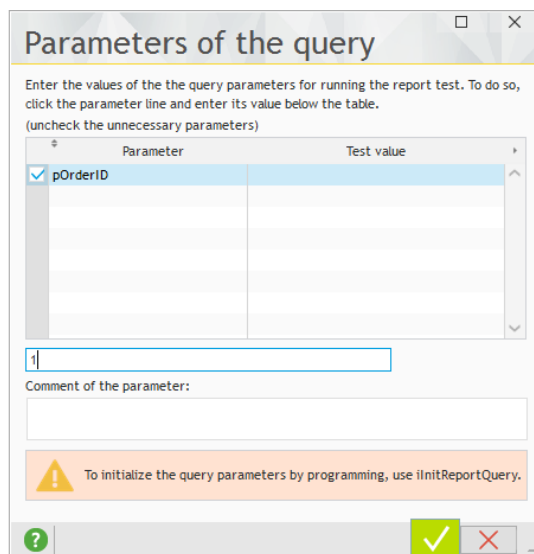
At run time, no print preview will be displayed to the user. The only available preview will be the display of the generated file by the browser.

- Printer: the report is printed on the default printer directly.
- Specific file: the report is printed in a file to the selected format.



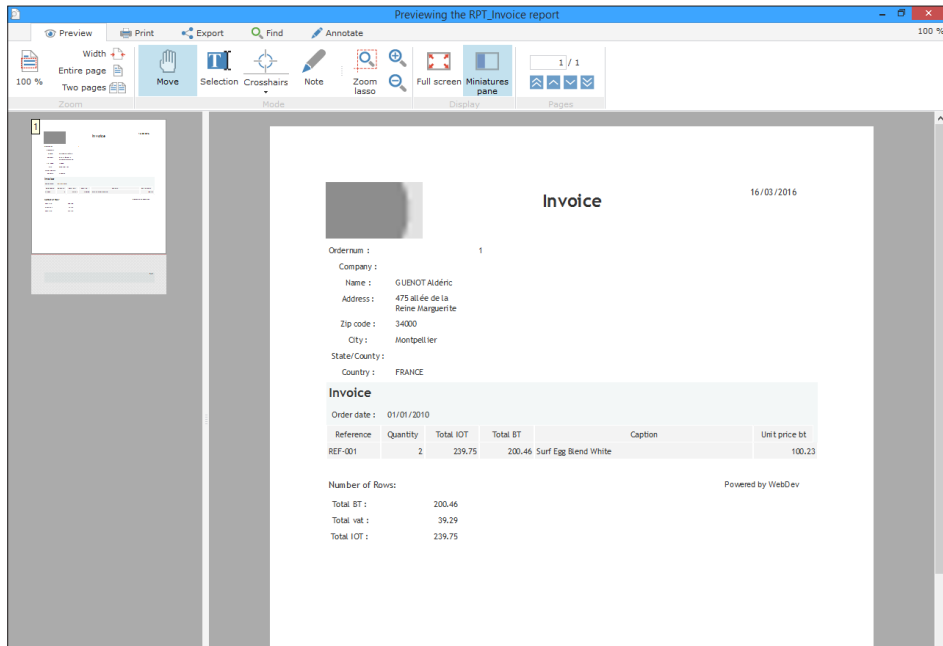
Choose "Print preview" and validate.

2. The report editor asks for the parameters of the 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 execution is displayed on the screen.

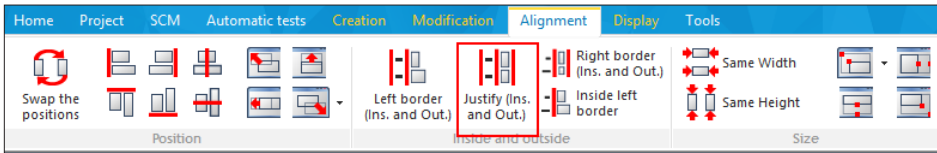


## Modifying the "Invoice" report

We are going to perform modifications in the report that was just created.

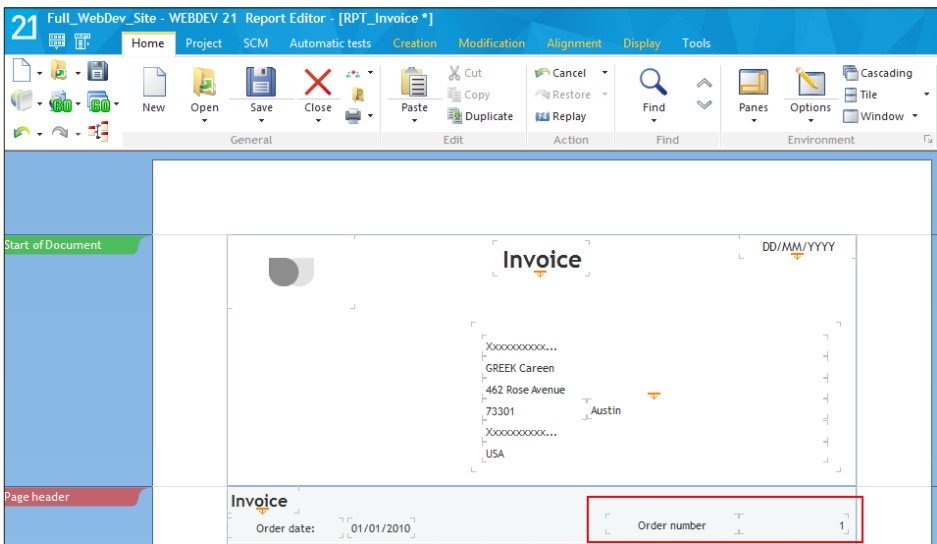
- We want to position the information regarding the customer and the order in the page header:
  1. Delete the captions found in front of the customer details (Name, ...).
  2. Position the item containing the city beside the zip code.
  3. Enlarge (with the handles) the control containing the company name: the size of the control 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 customer details and move them (with the mouse) on the right of the report.

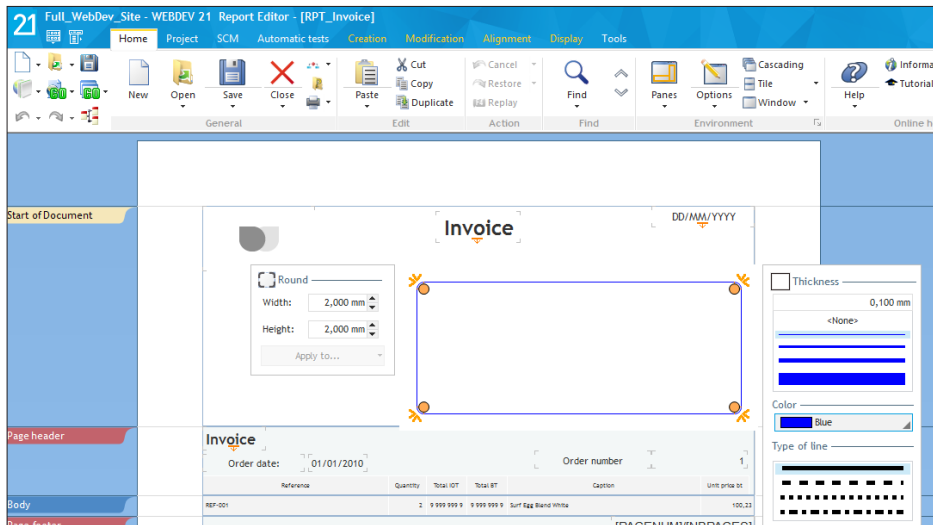
6. Move the order number up (to the top of the "Page header" block).



► We are going to create a border around the customer details:

1. Create a Static control: on the "Creation" pane, in the "Usual controls" 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 in the report to validate the modification.

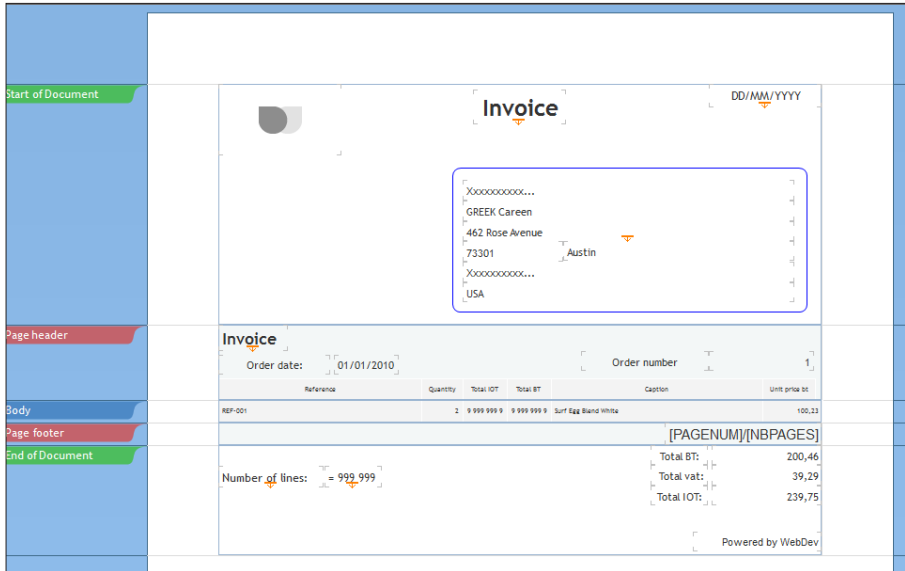
- To configure the border:
  1. Select the Static control that was just added.
  2. Press the [R] key and click the picto  that appears in the top right corner of the control.
  3. Select "Edit the border". The configuration options are displayed:




4. Select:
  - the dark blue color in the context-sensitive colors.
  - a simple border (thin 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 the table.

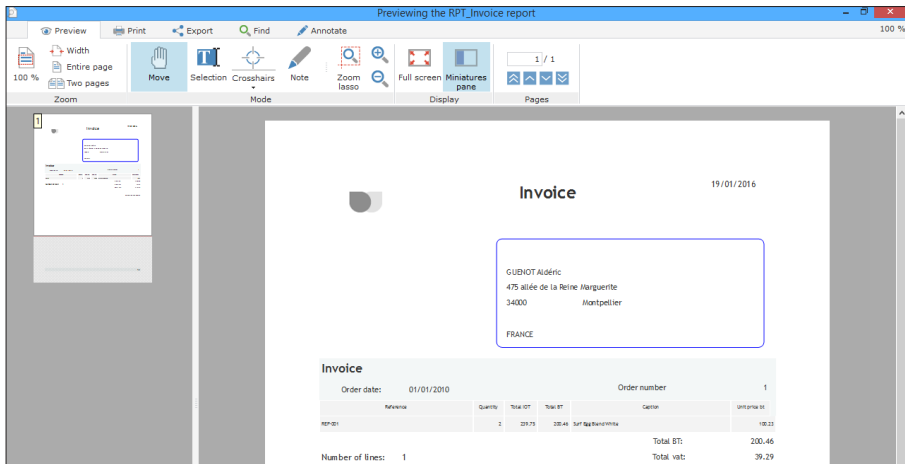


3. The report is displayed in the report editor:



4. 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 print preview.

Our "Invoice" report is completed.

## 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).

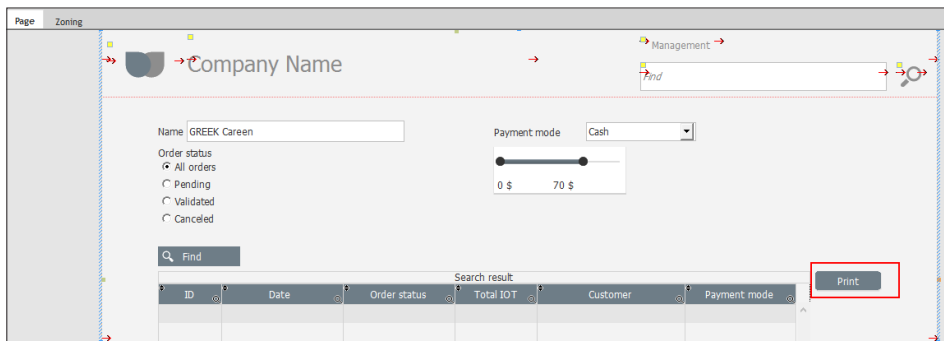
Therefore, the print will be done in PDF format on the server and the generated PDF document will be downloaded or displayed on the computer of the Web user. Then, the document can be printed by the user on his printer.

In our site, the "PRT\_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 the 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 the 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 enter 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")
```

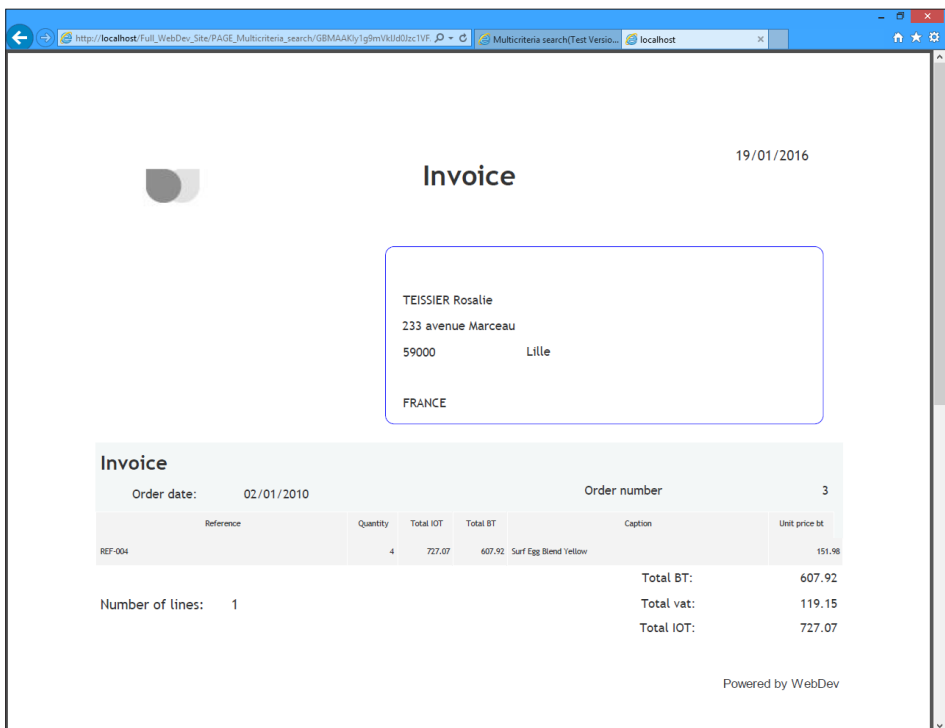
Let's study this code:

- **iDestination** allows you to define the print format used (PDF format in this case).
- The "RPT\_Invoice" report being based on a query with parameters, you must pass the parameter 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 start generating the "Report\_Invoice" report in the specified format (PDF in our case).
- **FileDisplay** is used to display the invoice to the user:
  - **iLastFile** is used to get the path of the last file generated by a report.
  - "application/pdf" if the MIME type of the 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 the Web user.
- 7. Close the code editor.
- 8. Save the report (  or CTRL S).

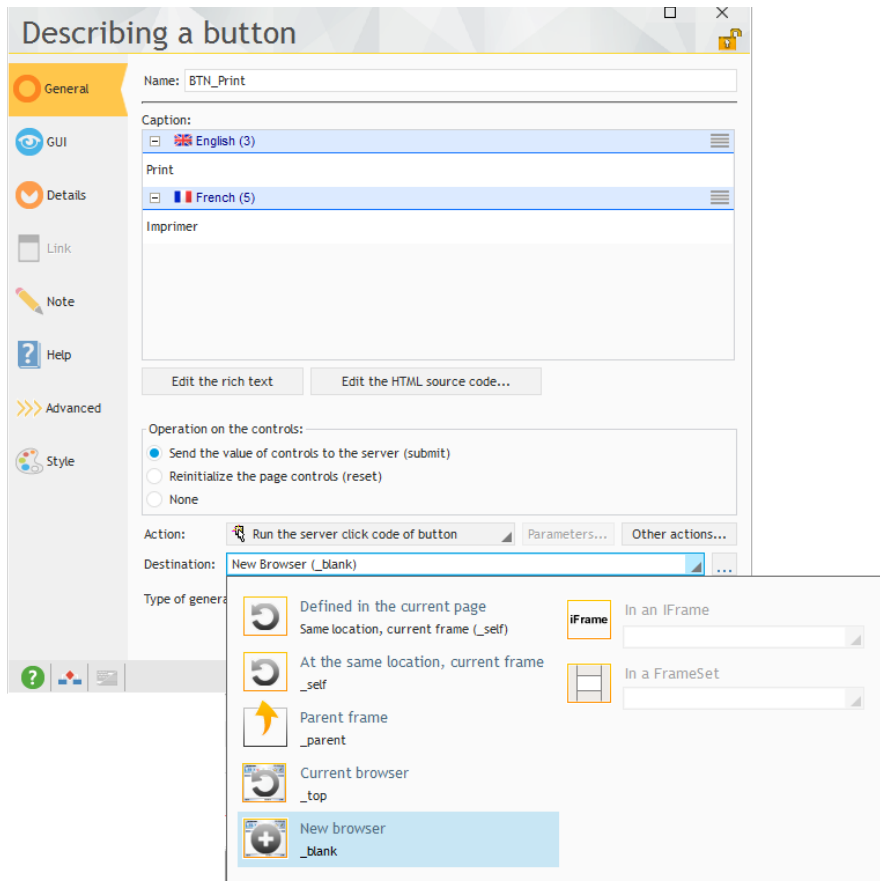
## Test of the print

- Now, all we have to do is run a real test:
1. Run the test of the "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 control, 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 min

## Overview

Several WLanguage functions allow you to manage the incoming and outgoing emails. You also have the ability to access all the characteristics of an email:

- 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 mode for email management 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 is used to leave the emails on the server so that they can be consulted from different messaging clients.
  - The **SMTP** protocol: this protocol for sending email is recognized by all the service providers.

In this lesson, we are going to create a Popup page allowing the user to send a suggestion by email to the site developer. This popup is as follows:

The screenshot shows a web form titled "Send suggestion" (partially visible). The form has a light gray background and a white content area. At the top, there are two input fields labeled "Sender" and "Subject". Below these is a large text area labeled "HTML text". Above the text area is a rich text editor toolbar with various icons for font style (bold, italic, underline), font color, background color, text alignment, list creation, and linking. To the right of the form, there are two "Modify" buttons. At the bottom of the form, there are "Send" and "Cancel" buttons. The form is displayed within a browser window with a dark gray border.

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.

**Answers**

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 of the application 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, gives more details about the email management. This example can be accessed from the home window of WEBDEV.

## A popup page to send emails

The popup page that we are going to create will contain all the controls allowing the user to enter 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. Increase the width and height of the Popup page via the handles.
6. Save the page (  or CTRL S).

### Creating the controls used to enter the characteristics of the email

To write an email, the user must have:


- A control used to enter the address of the sender.
- A control used to enter the subject of the email.
- A control used to enter the text of the email.

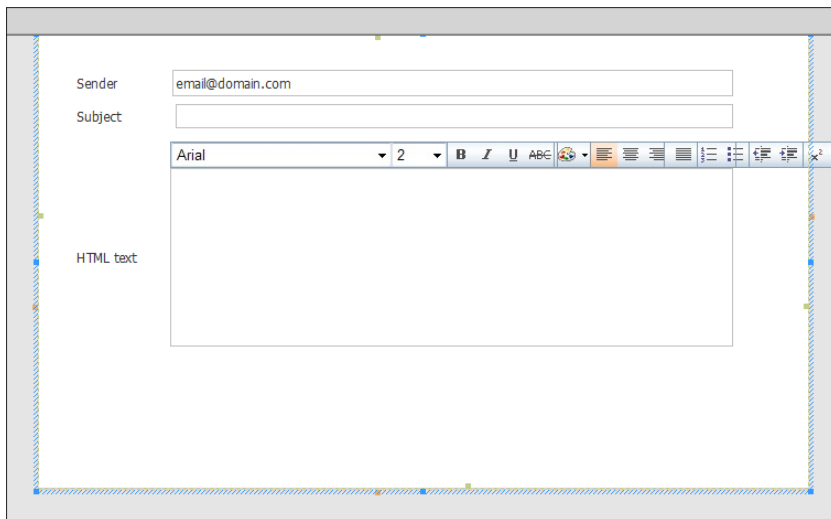
We are now going to add all these controls into our page.

Note: the recipient being the site developer, the corresponding address will be found in the sending code of the email. Similarly, the parameters of the SMTP server will be entered in the code directly.

► To create the edit control corresponding to the address of the sender:

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 caption of the control: "Sender".

- ▶ To create the edit control corresponding to the subject of the email:
  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 caption of the control: "Subject".
- ▶ An HTML edit control will be used for the body of the message: the user will have the ability to format the text of the email 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 shape of the control 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 the 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 the control.
  7. Reposition the control if necessary.
- ▶ Align the different controls found in the page.




We are now going to create the button used to send the 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. Enter 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"
// Enter the address of the SMTP server
MySession.Name = "user_name"
// Enter the user name (if necessary)
MySession.Password = ""
// Enter 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 the 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"
// Enter the address of the recipient for the suggestions
```

This code is using an Email variable. The different properties of this variable are used to define the characteristics of the email to send. This code associates the content of the different page controls to the properties of the 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 the SMTP session and the variable containing the characteristics of the 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 furtive 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 only for educational purposes. In a real site, we would have to check the parameters entered, process the errors, save a log file, ...

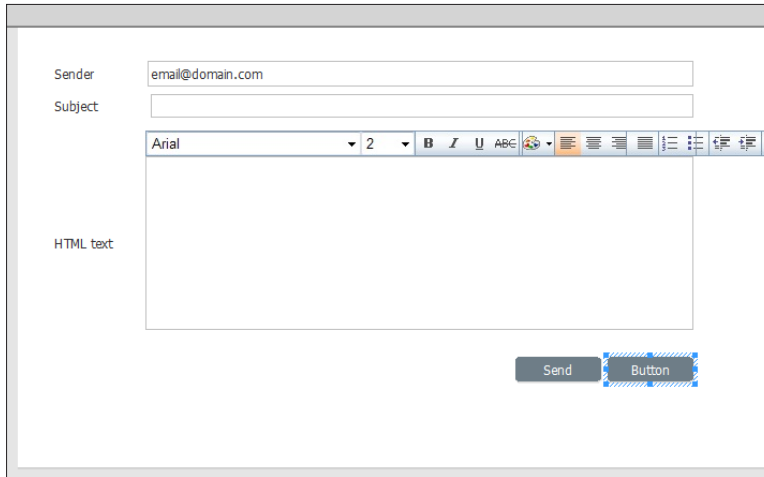
## Page improvements

We are going to improve our popup page:

- Add a closing button.
- Start the popup page from the "PAGE\_List\_of\_products" 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 must be created in the page (bottom right of "Send" button for example).

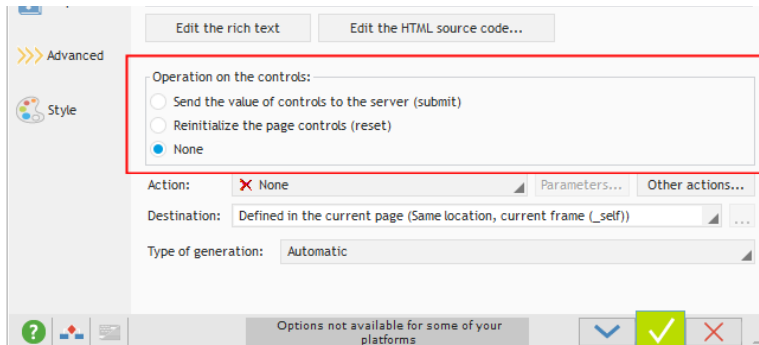


3. Select the control and press the [ENTER] key on the keyboard. The caption becomes editable. Enter "Cancel" and validate.
4. Display the processes associated with the button ([F2] key).
5. Enter the following code in the browser click code:

```
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 the "Cancel" button consequently:
  1. Display the description window of the "Cancel" button (double-click the control for example).
  2. In the "General" tab, in the "Operation on controls" area, select "None".

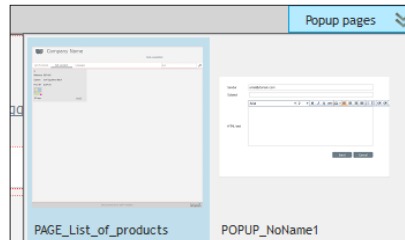


3. Validate the description window of the control.
- Let's now see how the popup page can be opened.

## Opening the popup page

The popup page for sending an email will be opened from the menu of the "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 used to send 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 the control (double-click the control).
    5. In the "Operation on controls" area, select "None".
    6. In the "Action" area, select "None".
    7. Validate the description window of the control.
    8. Display the processes associated with the button ([F2] key).
    9. Enter the following code in the browser click code:

```
PopupDisplay (POPUP_NoName1)
```

**PopupDisplay** is used to display the popup.





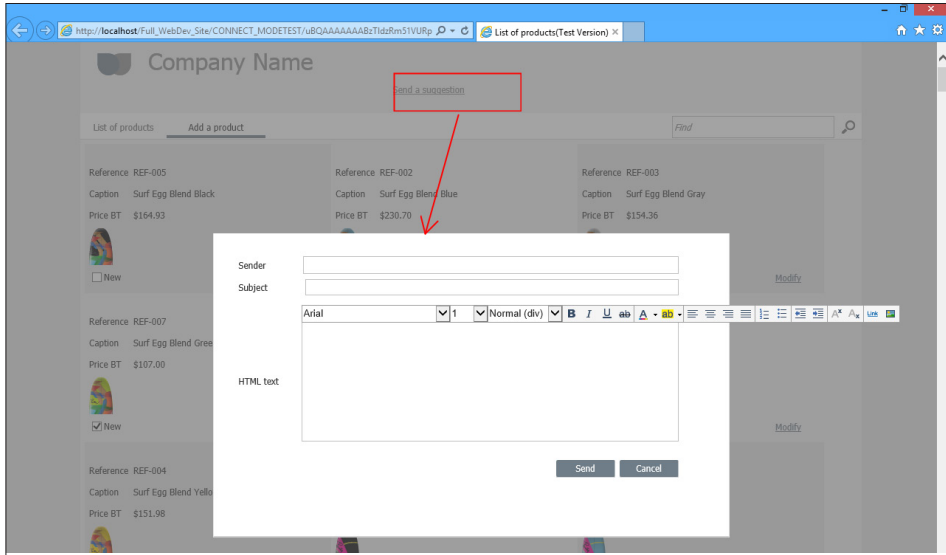
Notes

### 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 test of the page (  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 min

## 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 access to the page for adding products.

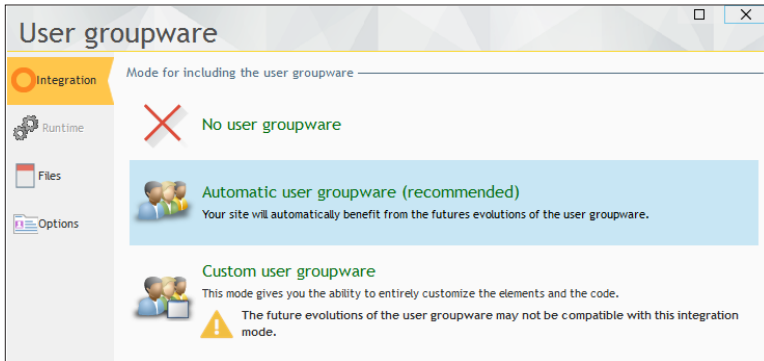


### Answers

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 enters 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.



Note

A single user exists by default, the supervisor. During the first startup of the site, connect by using the name: "supervisor" and the password "supervisor".

4. Validate this message. The user groupware is included 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 configuration of the user groupware is performed when running the site. This configuration consists in defining the different 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 along with the site.
- when the site is installed, by the administrator.

## Including the customer area in the site

In this example, we are going to include the connection link in the "PAGE\_List\_of\_products" page. The connection link is supplied in the format of a control template.



Notes

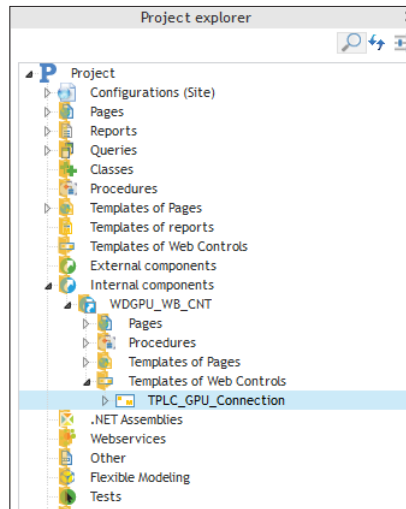
A control template is a specific page containing several controls. All types of controls can be found in this page. A control template is a file whose extension is "WDT".

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 it is WEBDEV that applies the modifications automatically.

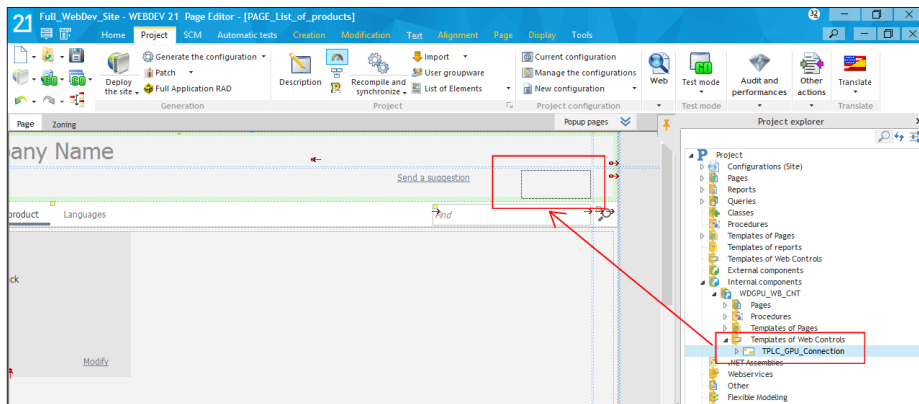
The control templates can be overloaded: code can be added, the controls can be moved in the page that is using the control template. The controls can be modified.

► 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 elements required to its management, especially the control template used to connect.
3. Then, expand "Template of Web Controls".




4. Select the control template named "TPLC\_GPU\_Connection" and perform a Drag and Drop to the "PAGE\_List\_of\_products" page. Drop the control template above the search control.

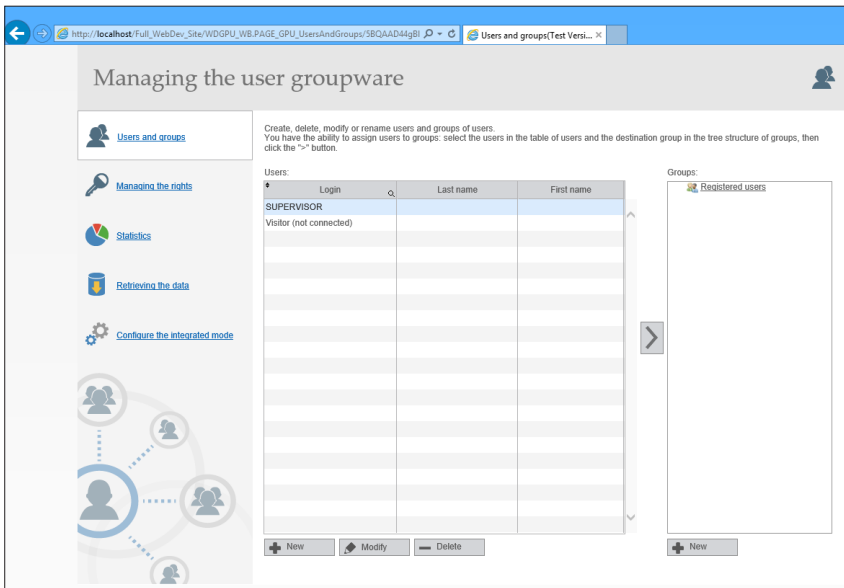


5. The control template is integrated. The "Connection" link appears.

## Test of the site

► Let's now run the test of our site:

1. Run the project test ( among the quick access buttons).
2. The page corresponding to the list of projects is displayed.
3. Click the "Connection" link. The connection page is displayed.
4. Connect yourself as supervisor:
  - Login: supervisor
  - Password: supervisor
5. Validate.



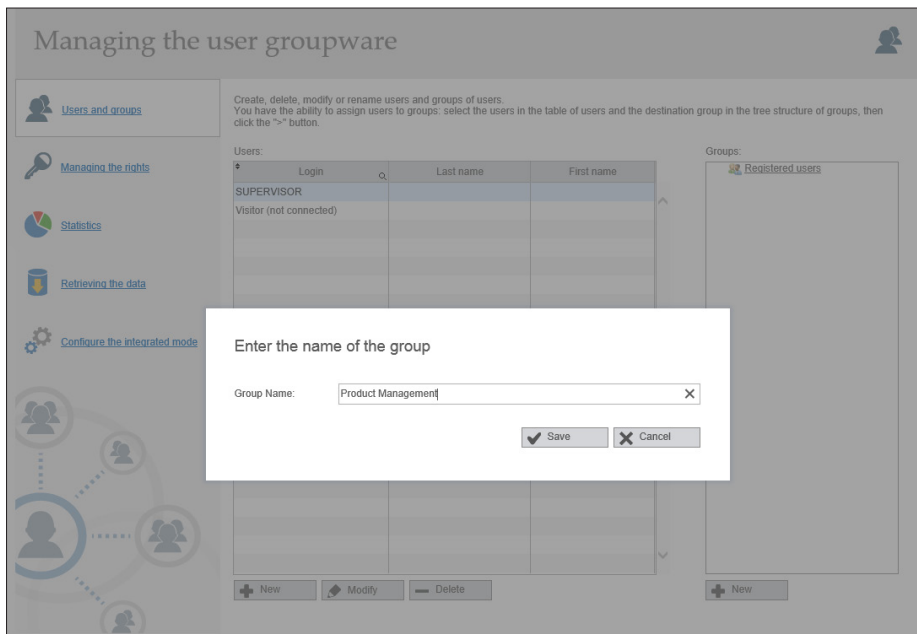
6. The page for groupware configuration is displayed.

## Creating the users and the groups

To configure the user groupware, we are first going to create a "Product Management" group. This group will group all the users allowed to modify and add products in the site.

Then, we are 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. Enter the name of the group: "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. Enter the following information:
    - Login: Alice
    - First name: 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 will be allowed to modify the users, the groups and the rights.

3. Click the "Save" button. The user named "Alice" appears in the list of users defined for the user groupware.

► To associate the user "Alice" with the "Product Management" group:

1. Select the user "Alice" in the page.
2. Select the "Product Management" group.
3. Click the addition arrow.
4. The association is performed.

## Defining the rights

We must now define the access rights to the menu for adding products.

**A good practices consists in refusing the default accesses and in allowing the access to the selected groups only.**

- ▶ We are going to define the 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 "Managing the rights" on the left of the page.
  2. To prevent all the users from accessing the page for adding products, select the default group for new users "Visitor (not connected)".

Managing the user groupware

Select the user or the group whose rights must be managed then click the "Next" button to choose the element to configure.

Users and groups:

- Groups
  - Product Management
  - Registered users
  - Users
    - Alice (Product Management)
    - SUPERVISOR
    - Visitor (not connected)

Information:

User login:

Last name:

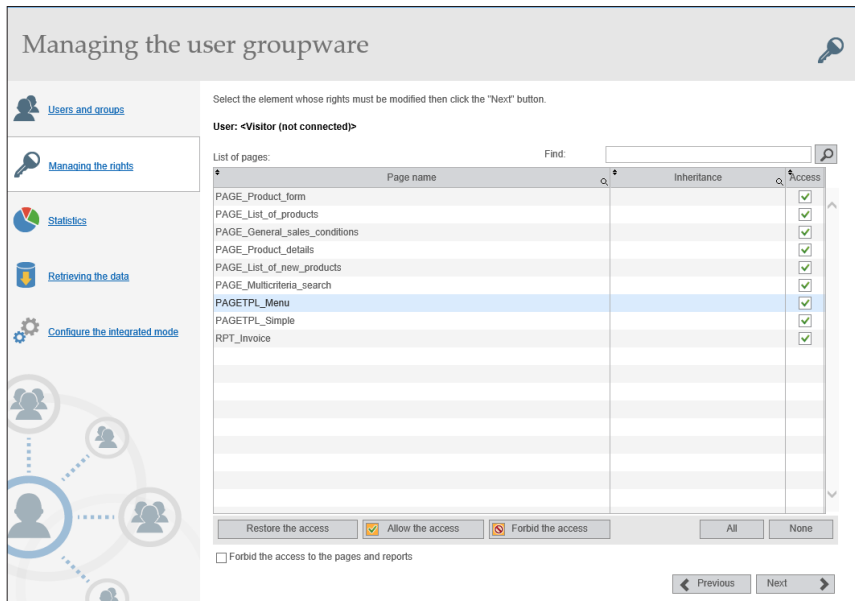
First name:

Email address:

Phone:

Next

3. Click "Next".
4. The page that is displayed is used to select each application page, page template or report.



- For each page, page template or report, you have the ability to specify whether the element will be accessible or not by the group.
  - For each page or page template, you have the ability to define whether the controls will have the site behavior (default) or whether they will be inactive, invisible or grayed.
- 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 the rights on the page controls is displayed.
  4. Select the menu option to configure: "Sub-menu Option\_2"

Managing the user groupware

Define the rights on the page controls.

Element: <PAGETPL\_Menu>  
User: <Visitor (not connected)>

Find:

List of controls:

Name	Caption	Status
Cell AREA_Header		
Cell AREA_Header_Navigation		
Menu control MENU_bro	Button	
Sub-menu Languages	Languages	Default
Sub-menu Option_2	Add a product	Grayed
Menu option Option_1	Option 2	Default
Menu option S_Option_1	Option 1	Default
Menu option Option_1	List of products	Default
Clickable Image IMG_Search		Default
Menu option OPT_NoName1	French	Default
Menu option OPT_NoName2	English	Default

Restore the default status Disabled Grayed Invisible All None

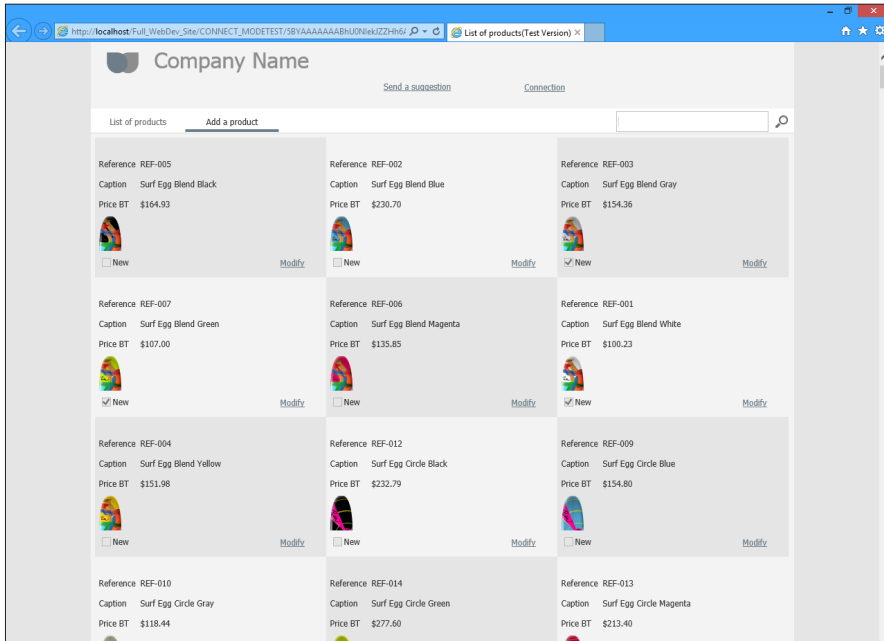
Previous Save

5. Click "Grayed".
6. Save the modifications by clicking "Save".
7. Close the browser.

## Test of the site

- 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 min

## 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 into French.

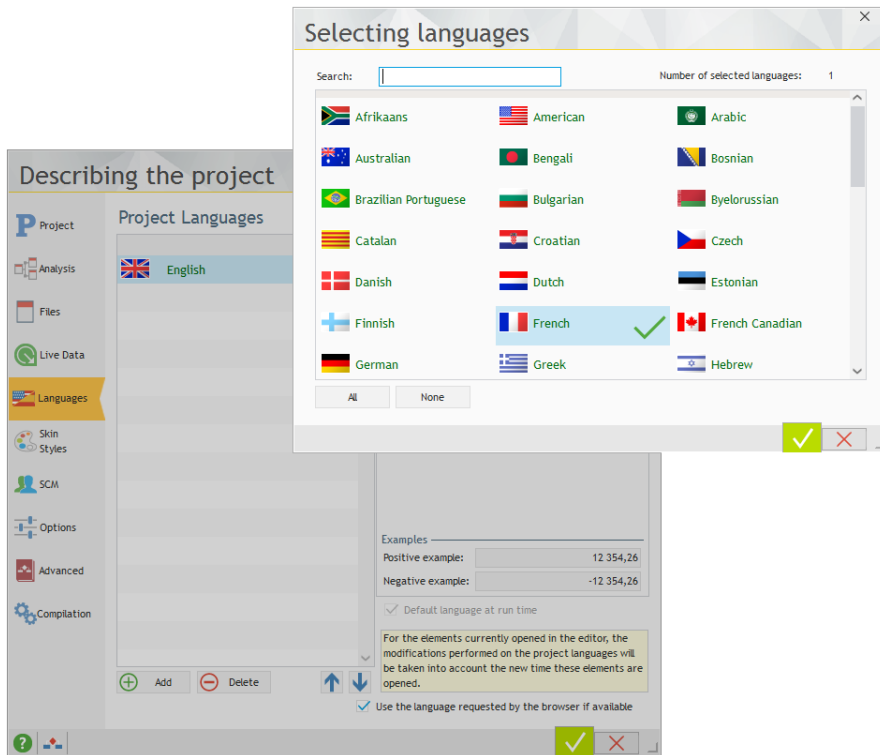


Answers

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)"

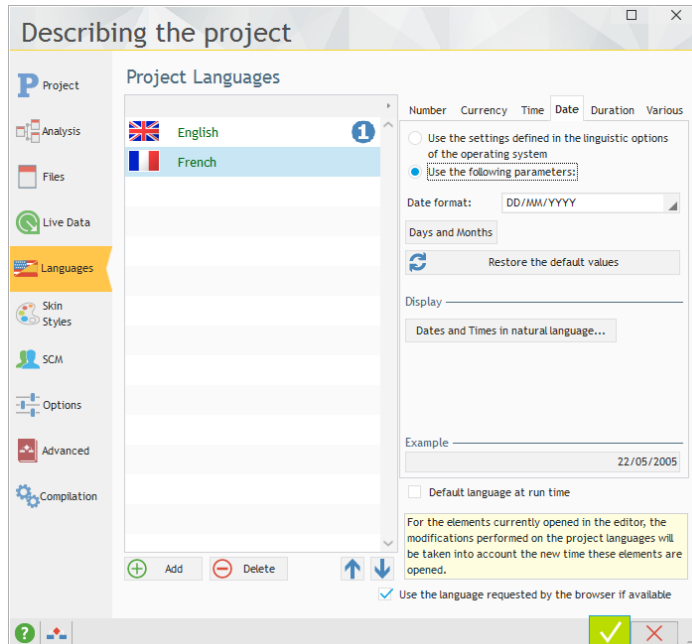
## Choosing the project languages

- 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 the 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 settings defined in the options of the operating system", the parameters used in deployment will be the server parameters (and not the user settings).
  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 are going to ignore them for now. These errors will be processed later in this tutorial.

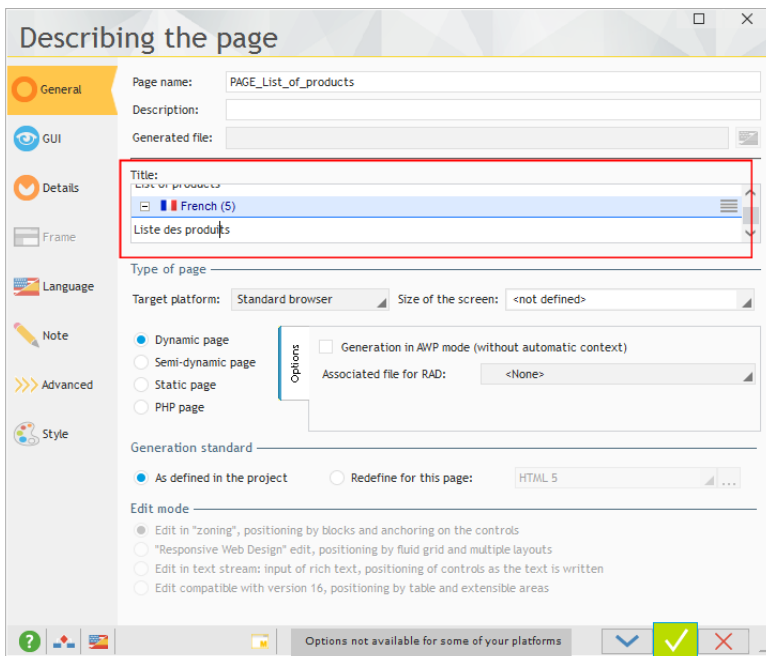
## Localizing the project elements

All the project elements can become multilingual elements: pages, reports, ...

We are going to modify some elements of the "PAGE\_List\_of\_products" page to present the different methods that can be used.

We are going to see how to modify:

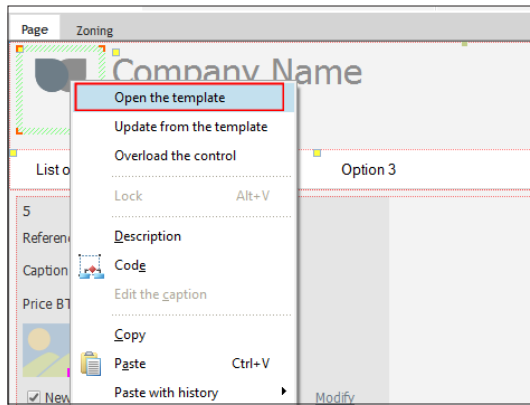
- the image of the logo used in the PAGE\_List\_of\_products page.
  - the captions of the controls found in the PAGE\_List\_of\_products page.
  - the menu options.
  - a message displayed by the WLanguage code.
- Open the "PAGE\_List\_of\_products" page in the editor (double-click its name in the "Project explorer" pane for example). Validate (if necessary) the update of the template.
- First of all, we must check whether the PAGE\_List\_of\_products page is associated with the different languages defined in the project:
1. Display the description of the page ("Description" from the popup menu of the page).
  2. Select the "Language" tab: the two languages selected in the project are displayed.
  3. Select the "General" tab: the title of the page must be translated.




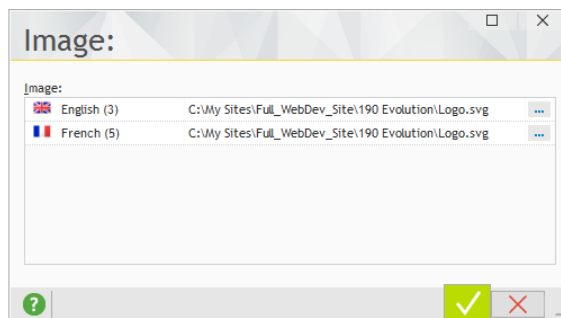
4. Enter "Liste des produits" in the French area.
5. Validate the window.

## Localizing an image

- To change the image of the logo 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 image of the logo 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 appears, enclosed in an orange border.
- 3. Display the description window of the logo (double-click the control).
- 4. In the "General" tab, on the right of the "Image" area, click the button .
- 5. The window for managing the multilingual images is displayed. A different image can be used for each language. This feature is very useful if you are using images containing text.





- 6. Close the description window of the control.
- As we are located in the page template, we are going to take the opportunity 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 the option, or from the page editor directly.

- In our example, we are going to translate the "List of products" option.
  1. Select then click the menu found in the page template.
  2. The menu becomes editable and a yellow border is displayed.
  3. Select the "List of products" option.
  4. Display the description window of the option: display the popup menu and select "Option description".
  5. In the description window, enter the option in French "Liste des produits".
  6. Validate.

The options can also be translated in the page editor directly.


- To translate the "Add a product" option:
  1. On the "Display" pane, in the "Options" group, expand "Language displayed" and select the language that will 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. Enter the caption in French: "Ajouter un produit".
  6. Validate the caption with the ENTER key.
  7. Press the [ESC] key to exit from the edit mode.
  8. Save the page template (  or CTRL S).
  9. Switch the displayed language back to English: on the "Display" pane, in the "Options" group, expand "Language displayed" 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 the 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 the 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 the control ("Description" from the popup menu).
  3. Enter the caption in French: "Modifier".
  4. Validate.
  5. Save the page (  or CTRL S).



## Localizing a programming message

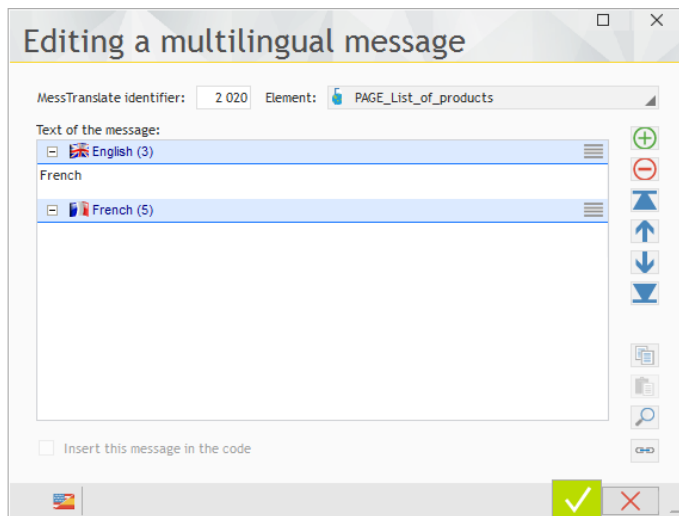
All the messages found in your program can be translated into several languages.


Let's see how to translate a programming message:

1. Display the code editor (press [F2] on the PAGE\_List\_of\_products" page).
2. Enter 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 following window is displayed:



5. This window allows you to translate all the messages of your program into all project languages.
6. In the "French" area, type "Français" and validate.
7. The icon  as well as a digit appear in the code editor. These icons indicate that the multilingual message exists in 2 languages.
8. 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:

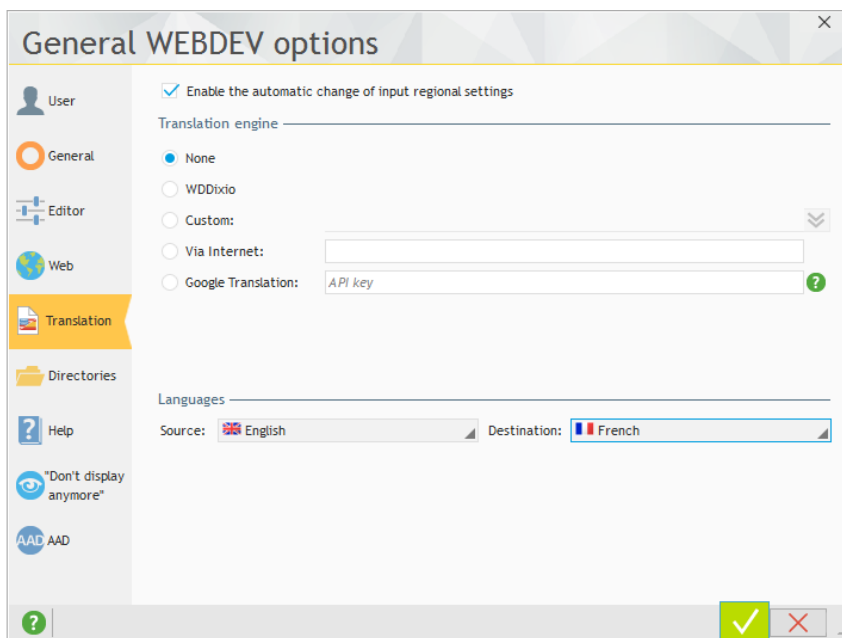
- a direct translation of messages performed in the different editors: this is the feature that was just used to translate some elements of our site. This translation can possibly use a translation tool, Google Translate (providing that you own a license), ...
- an "industrialized" translation performed via an external tool (WDMSG and WDTRAD).

### Direct input of translations

The translations are entered 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 must be used for the translation. You have the ability to use WD-Dixio, 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 the 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 the translated messages back in.

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.

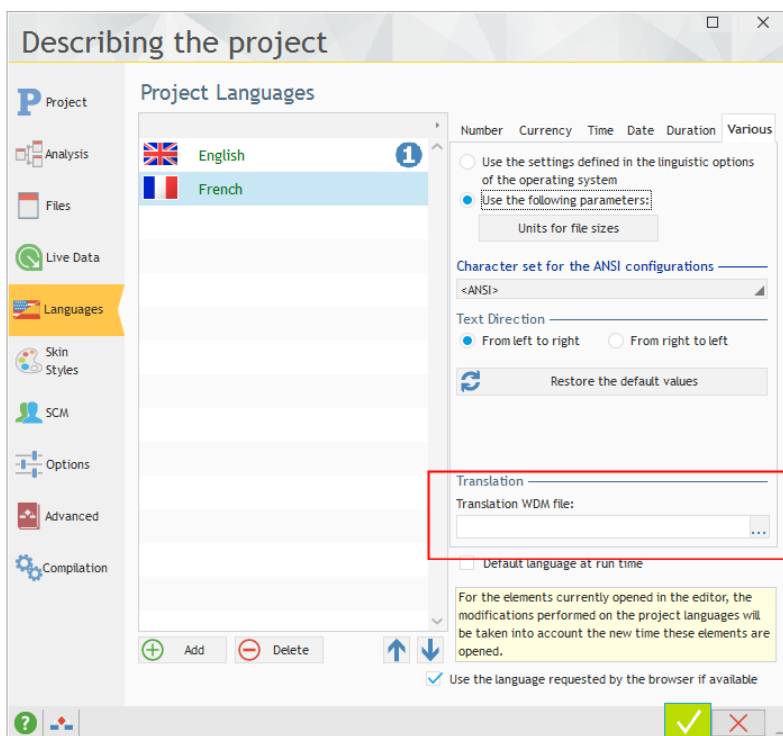
Contact PC SOFT Sales Department for more details about **WDMMSG** and **WDTRAD**.

## Other elements to translate: the framework messages

Various information and messages are found in the WEBDEV framework. For example, the names of the 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 the libraries. To use this file in your application:

- you have the ability to 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 the 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 of 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 is displayed.
  5. Press the [SPACE] key on the keyboard: the caption becomes editable.

6. Enter the "Languages" caption and validate.
7. Select the "Languages" option that was just created.
8. Display the popup menu (right mouse click) and select "Insert a sub-menu".
9. Press the [ENTER] key to edit the sub-menu.
10. Type the caption of the first sub-option: "English".
11. Press the [ENTER] key.
12. Press the [SPACE] key and type the caption of the second sub-option: "French".

► While we are positioned in the page template, we are going to delete the last two menu options:

1. Select "Option 5". Display the popup menu of the option (right mouse click) and select "Delete". The option is deleted.
2. Repeat this operation to delete the option 4.

We are now going to enter the WLanguage code required to change language.

## Programming



- To enter 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. Enter the following code in the server code of the 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. Enter the following code:

```
Nation(nationFrench)
PageUse(CurrentPage())
```


In this code:

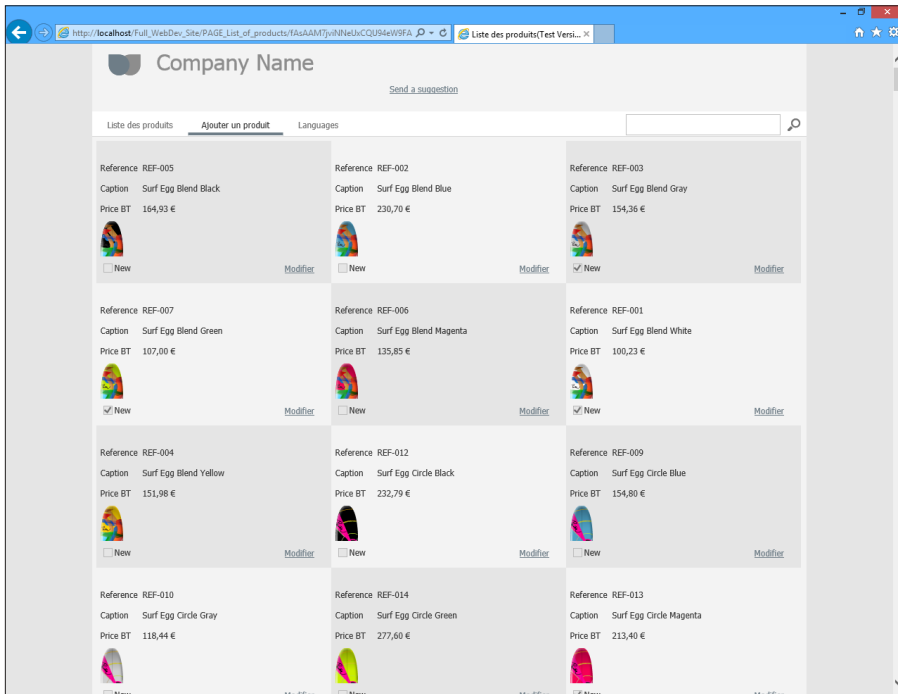
- **Nation** is used to change the runtime language of the site. The constants passed in parameter allow you to specify the language to use.
  - **PageUse** is used to re-display a page (the current page in our case) in order to take the change of language into account.
8. Close the code window.
  9. 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.

## Test of the project

Some application elements being translated, we are now going to 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 min

## 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 important if you are using an Intranet site.



Answers

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 properties of pages.
- The site map path.

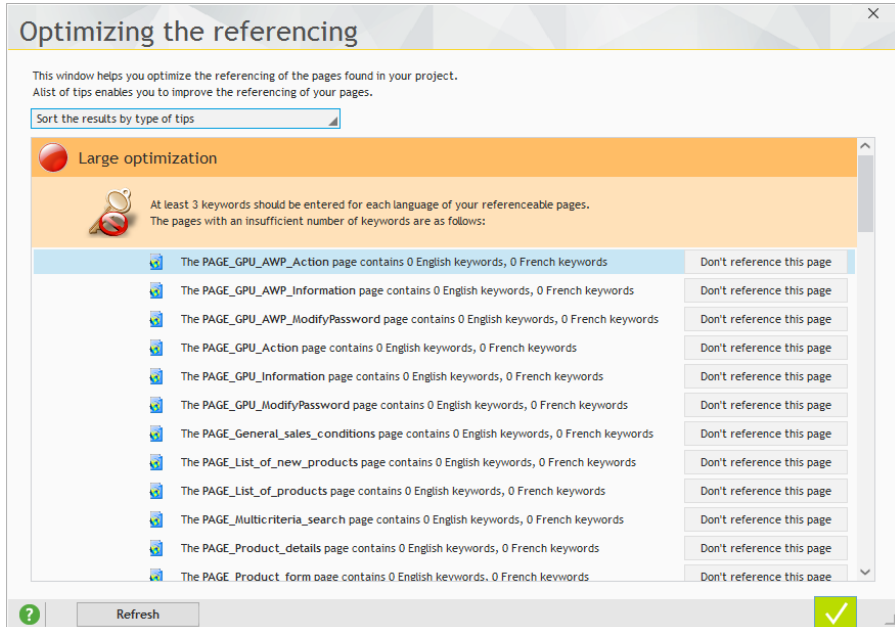
We are going to present these different methods.



## Referencing wizard

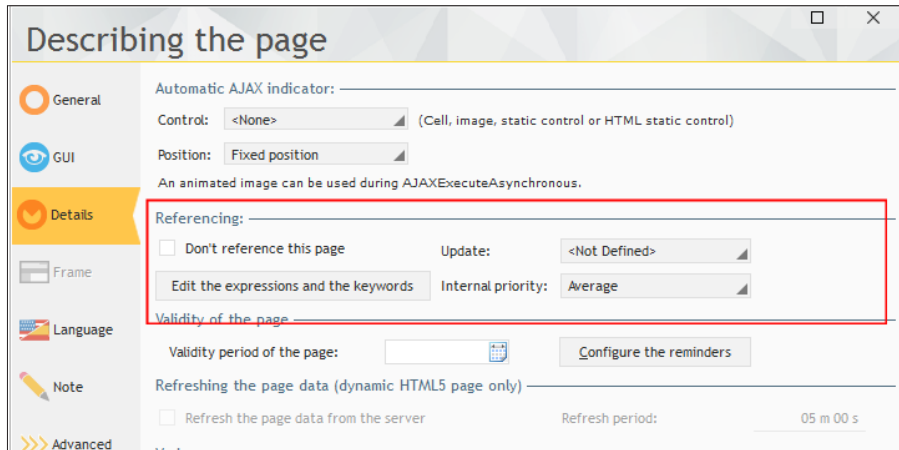
WEBDEV is supplied with a referencing wizard.

- 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 the referencing of the site. 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 title of a page is defined in the description window of the page, for each language.
- **Keywords of pages:** We advise you to associate keywords with a page. To define these keywords:
  - Display the description window of the "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 the keywords".



- **Alternative text:** The alternative texts are texts that are substituted to images when these ones are not displayed (either because the Web user has explicitly requested not to display the images, or because they are not loaded yet, or even because the Web user is using a browser in text mode or a screen for the visually impaired). The alternative texts describe the image and they are indexed by the search engines. Therefore, they must be chosen with great attention. The alternative text of an image is defined in the description window of the Image control, in the "Help" tab.



## Properties of pages

As we just saw, the referencing wizard indicates several modifications that can be performed in the description window of pages:

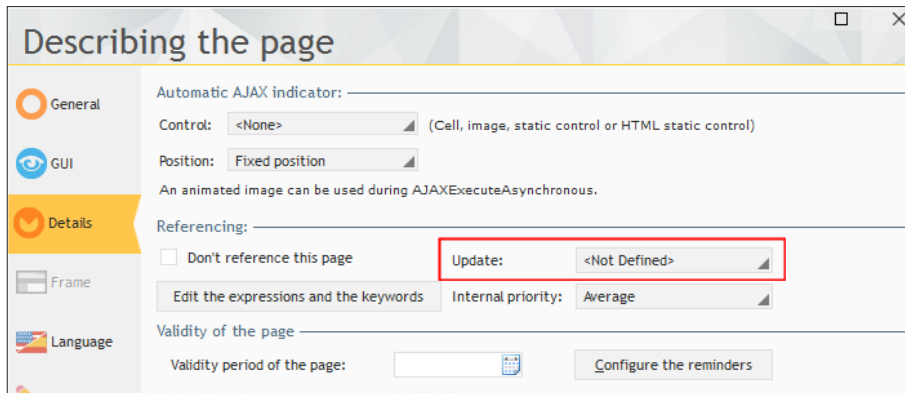
- Title of the page
- Keywords of pages, ...

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 indicates to the search engine the frequency at which the page must be reindexed.



Whenever possible, you should indicate a frequency close to reality:

- If the page changes more often than what is specified, some versions may not be indexed.
- If the page changes less often, the operation performed by the search engine will induce an unnecessary load on the server and the re-indexing may be penalized.

### Importance in the site

This criterion allows the search engine to identify the page that must be proposed first to the Web user when several site pages correspond to 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 inside 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 \_WEB directory of your site and it is named \_sitemap.xml.

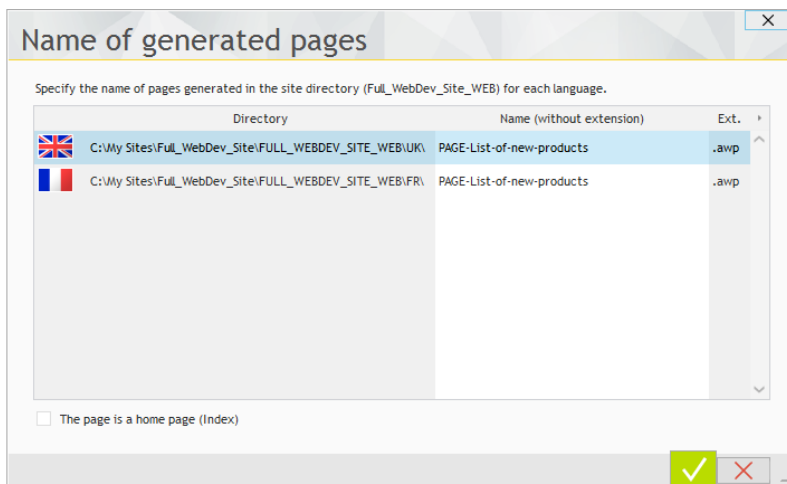
The sitemap.xml file must be supplied to the tools for webmaster of search engines so that they can index the content of your site.

## Physical naming of pages

To improve even more the referencing, you have the ability to define the physical name of the AWP pages so that it is different in every language (and different from the logical name as well).

For example, a page displaying a sales dashboard whose logical name (the name used in programming) would be PAGE\_SalesDash can be generated in English in the 'sales-dashboard.awp' file and in French in the 'tableau-de-bord-ventes.awp' file.

- To configure the generation name of a page:
1. Display the description window of the "PAGE\_List\_of\_new\_products" page.
  2. In the "General" tab, click the  button on the right of the "Generated file" control.
  3. A window used to configure the name of the generated pages is displayed:



4. You can change the name of the generated page and validate.

DEVELOP 10 TIMES FASTER

**PART 6**

**Site deployment**

21





## LESSON 6.1. DEPLOYING A SITE

**This lesson will teach you the following concepts...**

---

- How to deploy?
- Required configuration



Estimated time: 30 min



## 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"
- Deployment 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](http://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"

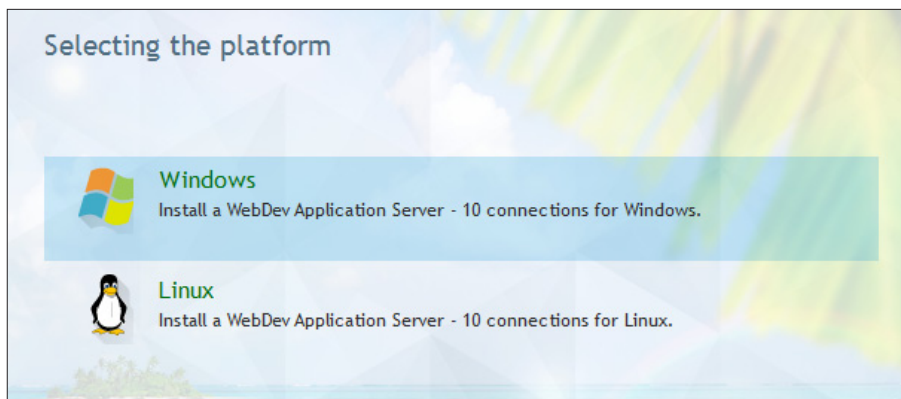
The setup steps are as follows:

1. Accept the license agreement.
2. Choose the "Windows" platform. Go to the next step.

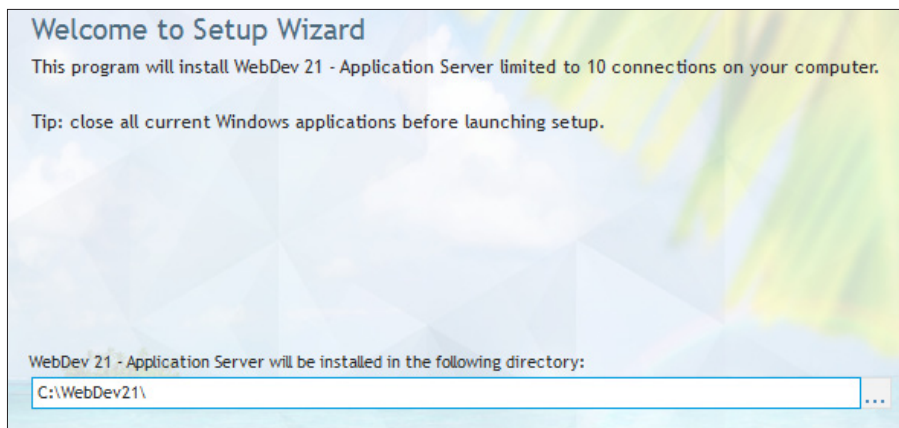


Note

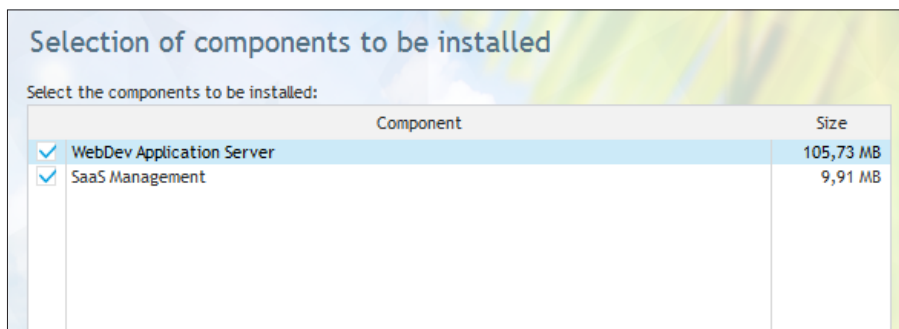
This section presents the deployment on a Windows server. If you want to use a Linux server, see the documentation about the WEBDEV application server for Linux or the online help.



3. Select the setup path of the application server ("C:\WEBDEV21" by default). Validate the directory creation.



4. The next step is used to specify the additional modules to install:
- The application server (mandatory module, cannot be unchecked).
  - The SaaS administrator that is used to manage your SaaS sites (see the online help for more details).



5. Validate this step without modifying the options.  
6. Keep "Host the sites in earlier versions" and go to the next step.  
7. 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).

8. Select the virtual Web servers on which the WEBDEV application server must be installed. The default Web site is sufficient. Validate.  
9. Keep the selected options and check "Start the WEBDEV administrator". Validate.  
10. 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:

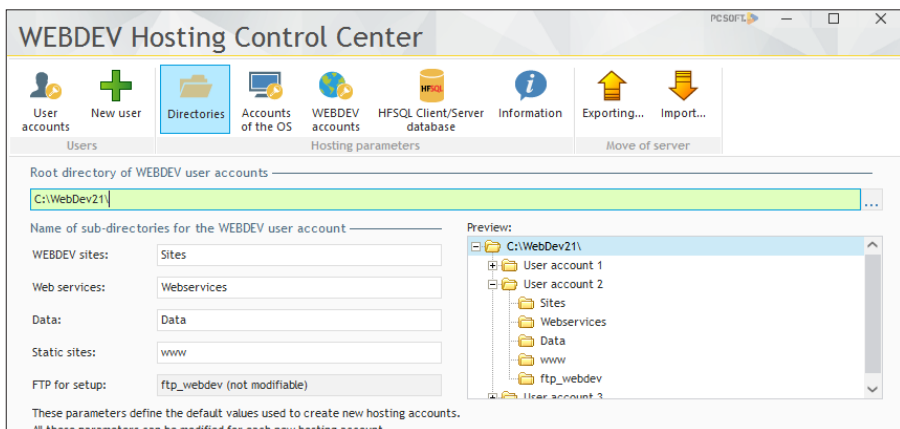
1. Start the Hosting Control Center from the "Start" menu of Windows.
2. Select the "Hosting parameters" tab. This tab allows you to define the parameters that will be used by default for your hosting server via four sub-tabs.
3. In the "Directories" tab, specify where the sites will be installed. Choose the root directory where the sub-directories of the 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.

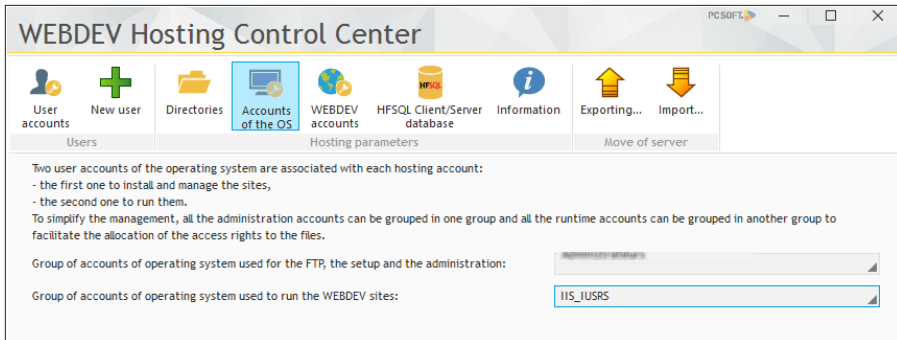
You can choose the name of the sub-directories that will contain the sites, the webservices and the data.



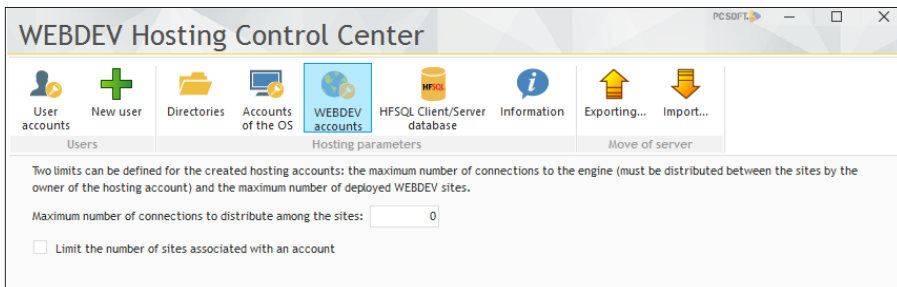
4. In the "Accounts of the OS" tab, define the groups where the Windows users created for the deployment will be assigned.

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).

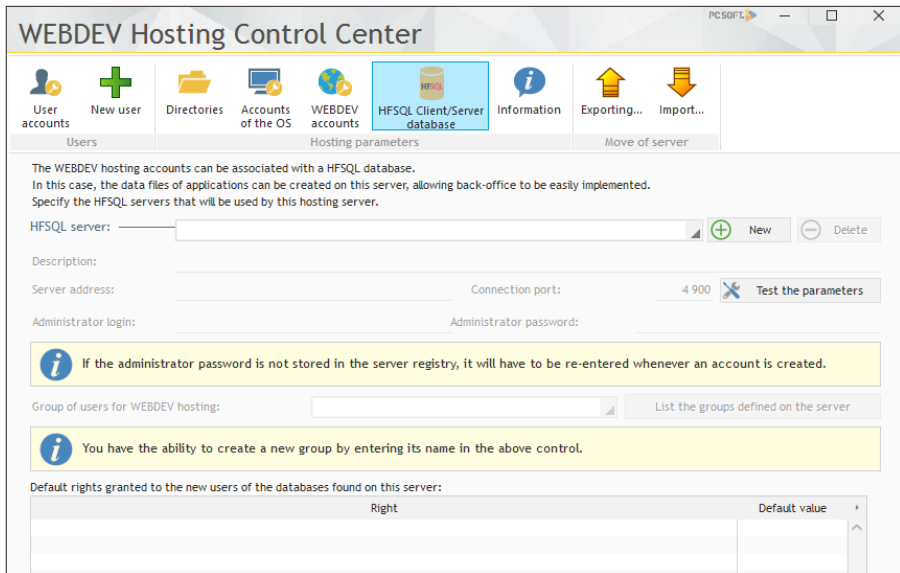


5. The "WebDev accounts" tab is used to specify the limitations of resources that must be applied to the WEBDEV accounts:



- 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. The "HFSQL Client/Server database" is used to configure the creation of a HFSQL database whenever a user is created.



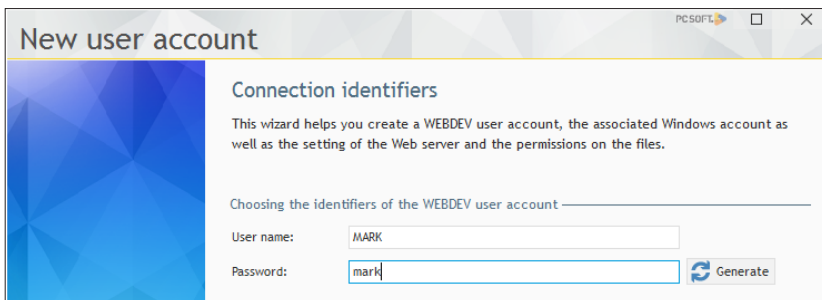
7. Once the hosting is configured, click "Apply".

## Creating a deployment account

The Hosting Control Center can also be used to create a deployment account.

To create a deployment account:

1. Click the "User accounts" tab.
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. Enter the name of the user and his 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.

**New user account**

**Associated Windows accounts**

Each WEBDEV user account must be associated with two local Windows accounts.

Windows account for remote setup (deployment) and management

- ☒ Create a Windows account with the identifiers of the WEBDEV user account (recommended)
- ☐ Create a Windows account by choosing the identifiers
- ☐ Use an existing Windows account

User name:

Password:

Windows account for running sites and webservices

- ☒ Create a Windows account with the identifiers of the WEBDEV user account (recommended)
- ☐ Create a Windows account by choosing the identifiers
- ☐ Use an existing Windows account

User name:

Password:

Navigation: < [checkmark] >

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 site" 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.

**New user account**

**Virtual Web site**

Each WEBDEV user account must be associated with a virtual Web site where its sites and webservices will be published.

- ☒ Create a new virtual Web site
- ☐ Use an existing virtual Web site

IP address	TCP port	Name of Host Header
(All non-assigned)	80	MARK.DocHC.pcsoft.com

Use an existing virtual Web site:

Navigation: < [checkmark] >

8. Go to the next step.
9. Select an FTP site. Go to the next step.
10. The wizard is over. Check all the choices. You have the ability to uncheck some operations if you do not 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. Different 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 having an FTP server. It operates on 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 generates 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 the WDDeploy tool. WDDeploy is a freely distributable tool that is supplied with WEBDEV.
- Deployment via the test hosting service of PC SOFT.
- Deployment in PC SOFT Cloud.

Let's study 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.

- To open this project in WEBDEV:
1. Close (if necessary) the current project in order to display the home window.
  2. In the home window, click "Tutorial" and select "Full WebDev Site (Exercise)".



Answers

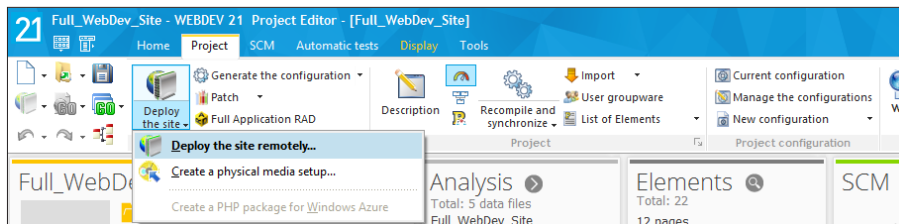
If you did not perform the operations in the previous parts, open the corrected project. This project contains the different pages created in the previous parts. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WebDev site (Answer)".

## Preparing the setup

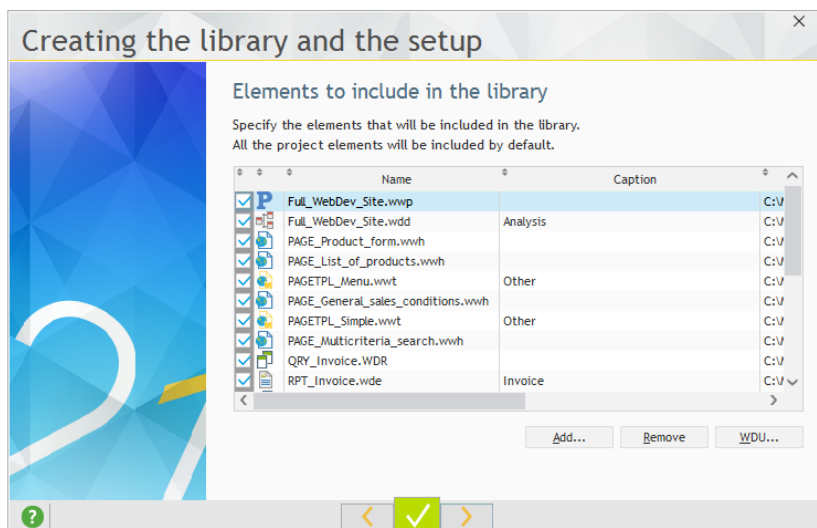
A setup wizard is supplied with WEBDEV ; this wizard allows you to easily install your site at the hosting company (Internet or Extranet site for example) or on one of your servers dedicated to WEBDEV hosting (Intranet site for example).

► 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.

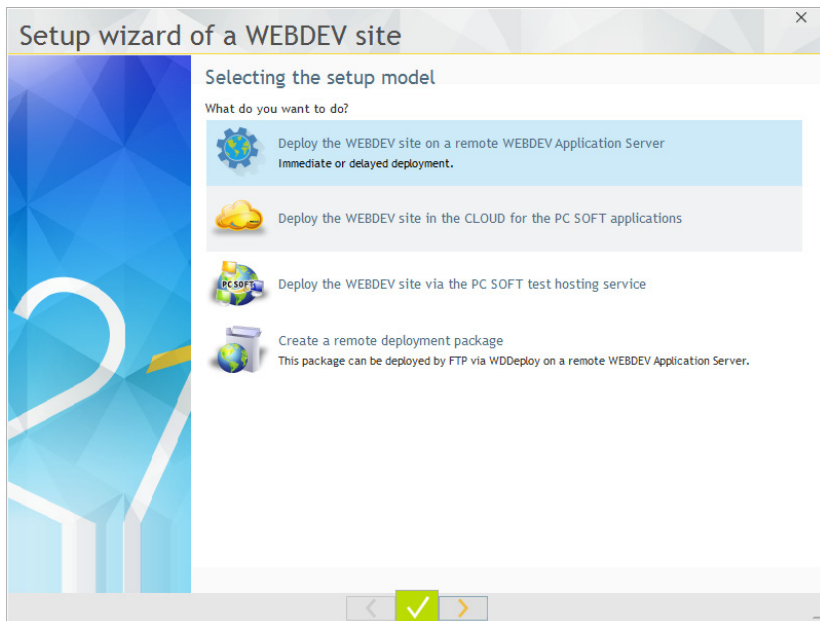
4. Several languages can be included in the library. In our example, we will keep the default options. Go to the next step.



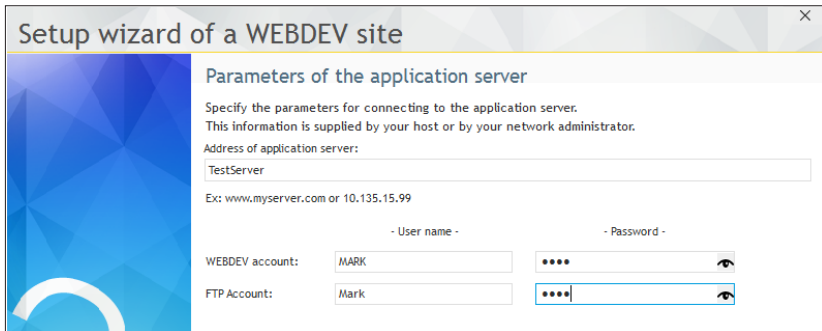
5. The information about the library version is used to enter the elements that will be displayed in the file properties in the Windows explorer. Go to the next step.
6. Don't save the project and validate the library creation.

## 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 plane.



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:
  - Address of the server (in our example, name of the computer where WEBDEV Application Server 10 connections was installed). The name can be:
    - the name of a computer accessible via network ("TESTServer" for example),
    - an IP address (192.168.15.99 for example),
    - an Internet address (www.myserver.eu for example).



- Characteristics of the user account entered by the hosting company in the WEBDEV account manager.
- Characteristics of the FTP account entered by the hosting company when creating an FTP account.



Note

Caution: the user name can be preceded by the name of the domain to avoid confusions.

For example: "mycomputer\test" or "mydomain\test"

3. When the information regarding your WEBDEV account and your FTP account was entered, go to the next step.
4. Enter 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:

Domain associated with the site \_\_\_\_\_

If you own a domain name corresponding to the deployment address, you can specify it here:

If no address is specified, the name of the deployment server will be used.  
 The domain name will be used to build the Sitellap 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:

If two sites use the same directory, their data files will be shared.

Name of deployment profile \_\_\_\_\_

Enter a name for this deployment profile:

< ☒ >

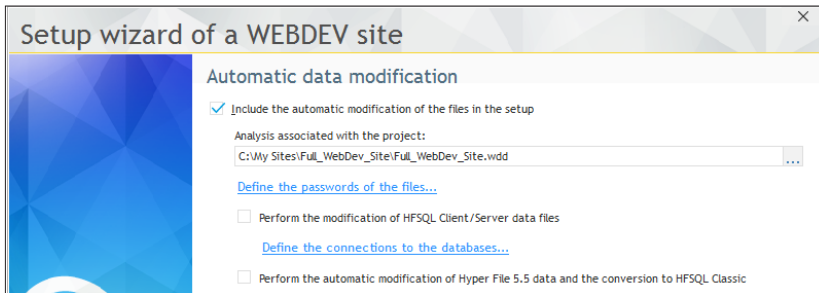
5. WEBDEV establishes the connection and it summarizes the operations performed (number of updated files, number of deleted files, ...). To get the details of operations and ot modify them if necessary, click the "Edit the list of files" button.

### List of files to deploy

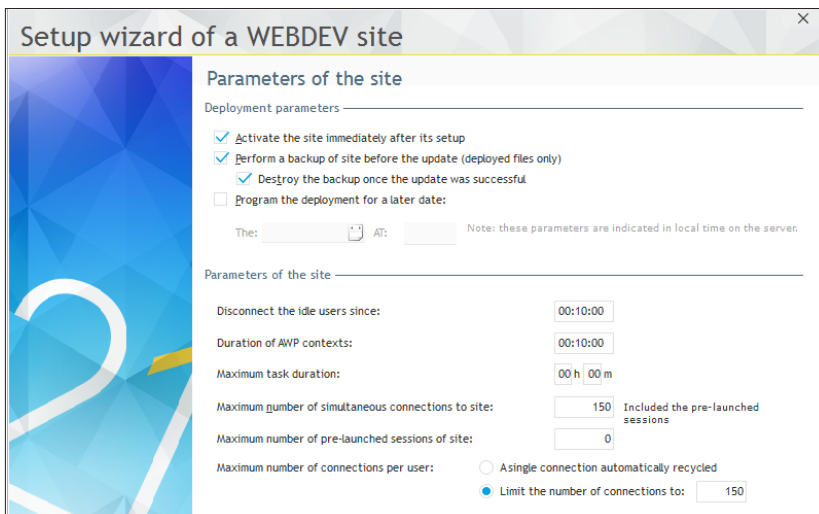
Files of the site	site Full_WebDev_Site on twr34-7	Destination directory
Full_WebDev_Site190EvolutionGrayVolution.css	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css
PAGE-List-of-new-products.asp	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css
PAGE-Product-details.asp	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css
PAGETPL_GPU_Action.asp	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css
PAGETPL_GPU_ModifyPassword.asp	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css
PAGE_GPU_AWP_Action_style.css	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css
PAGE_GPU_AWP_ModifyPassword_style.css	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css
PAGE_List_of_new_products_style.css	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css
PAGE_Product_details_style.css	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css
palette-GrayVolution.css	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css
_sitemap.xml	→ Update	Full_WebDev_Site190EvolutionGrayVolution.css

6. Go to the next step.

7. 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.



8. Specify the parameters of the site:



You have the ability to 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 the Web user if this one has performed no action since the specified duration.

9. By default, your site is enabled immediately after the setup. The users will have no access to your site if "Enable the site at the end of 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 proposes to automatically generate the statistical files for the site. These statistics affect for example the actions performed on the site, the origin of the Web users, ... Keep the options proposed by default and go to the next step.

13. 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 the 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"> <li>• data files (.fic, .ndx, .mmo, .ftx): &lt;data&gt;\&lt;site name&gt;</li> <li>• other files: &lt;site&gt;\&lt;site name&gt;</li> </ul> For a Webservice: <ul style="list-style-type: none"> <li>• data files (.fic, .ndx, .mmo, .ftx): &lt;data&gt;\&lt;site name&gt;</li> <li>• other files: &lt;webservice&gt;\&lt;site name&gt;</li> </ul>

where:

- <site>, <data> and <webservice> are the directories defined when creating the hosting account (see above).
- <site name> is the name of the site.

The distribution of the 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 min

## 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.

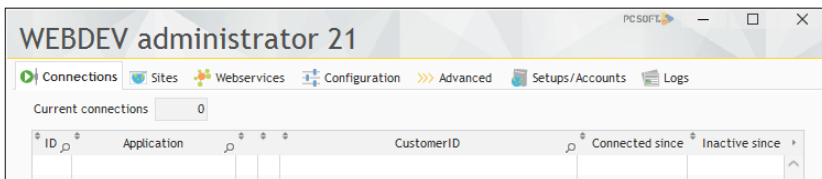


Note

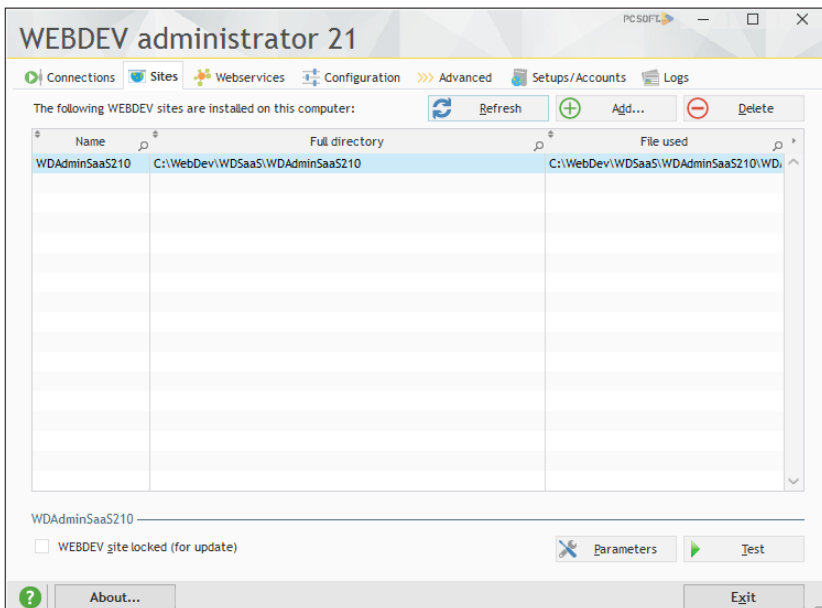
The WEBDEV administrator can be used only if you can directly access the Web server on which your site is deployed.

This application includes seven parts, represented by seven tabs:

- Connection: Displays the list of users connected to the sites and allows you to disconnect a user.

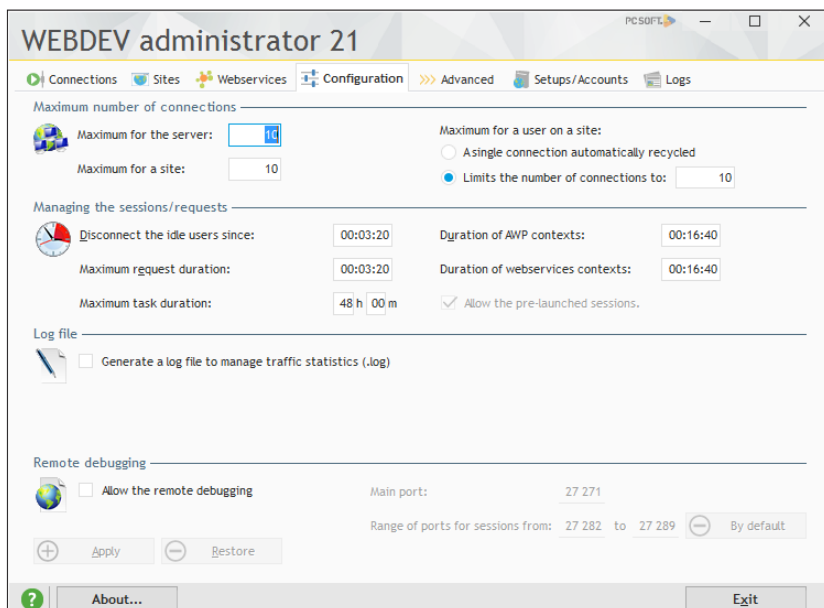


- Site: Lists the sites deployed on the server as well as their parameters:

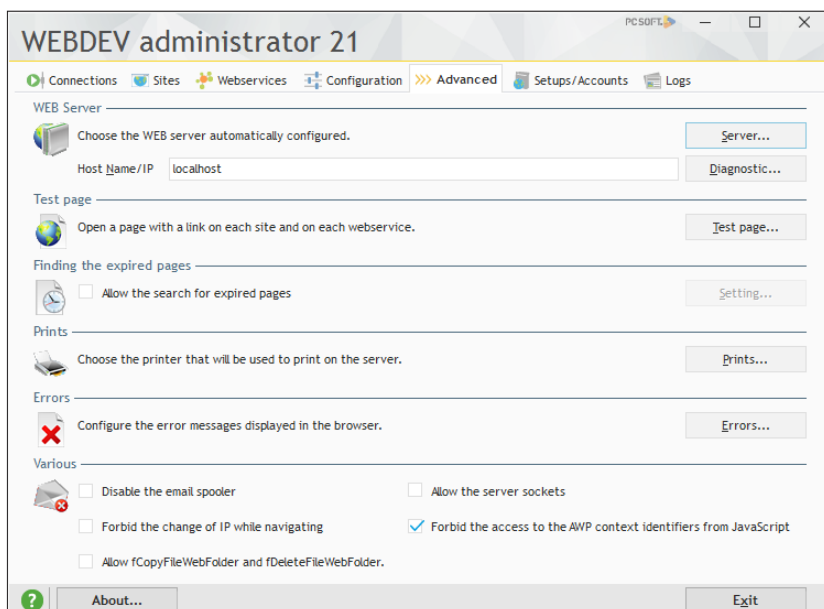


- Webservices: Lists the webservices deployed in the server as well as their parameters.

- Configuration: Used to manage the server configuration.

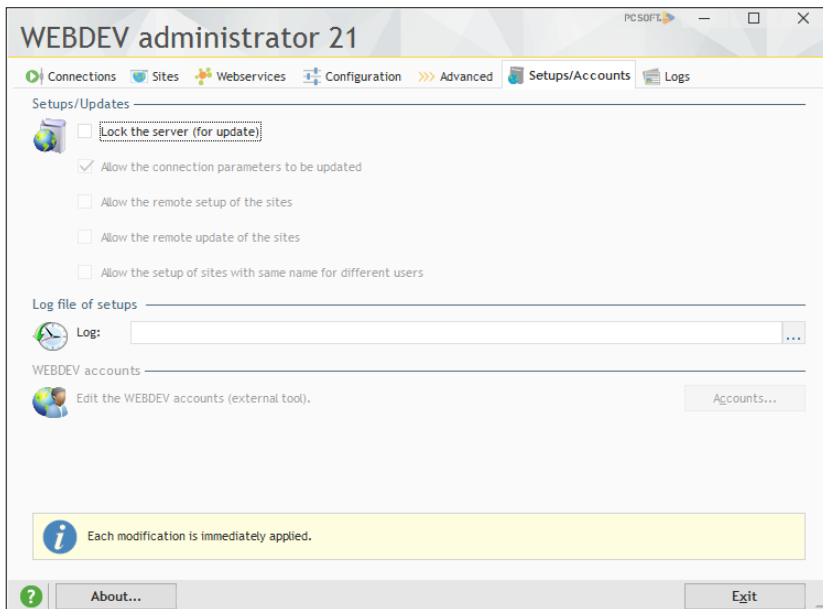


- Advanced: Configures the additional server parameters such as print management, email spooler, ...





- **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 installed elements of a site. It is very useful before contacting the Technical Support if errors occur during the deployment.



- **Logs:** Used to see the logs for the sites or the 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.

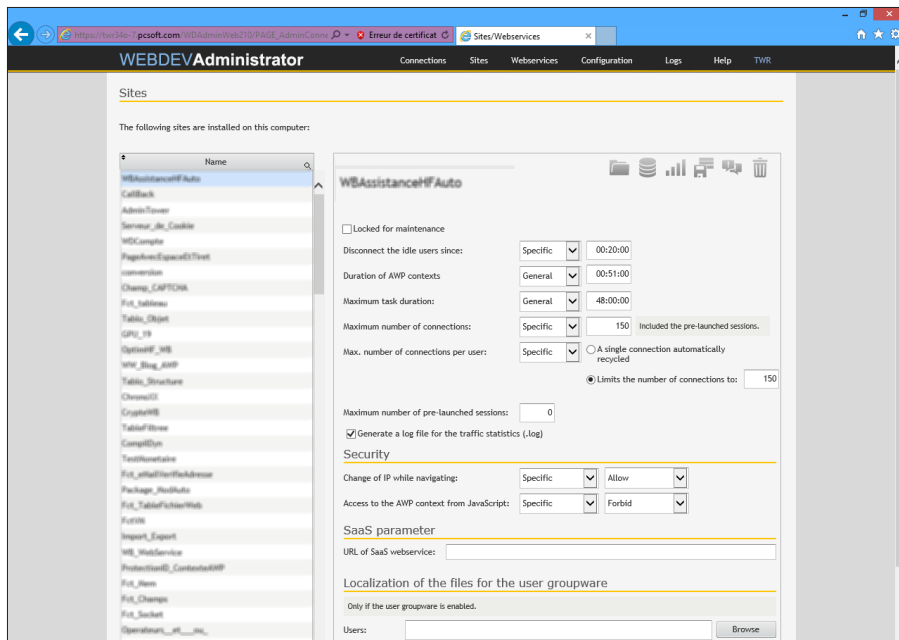
**Note:** This tool is not available for the WEBDEV application server 10 connections.

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 WAdminWeb210:
  1. Start the WEBDEV administrator (WAdminWeb210) on the deployment computer.
  2. Open your favorite browser (Internet Explorer for example) on your computer.
  3. Enter the following URL in the address bar of the browser (while respecting the case):

```
http://computer/WAdminWeb210
```

where "computer" is the name of the deployment server.



DEVELOP 10 TIMES FASTER

## PART 7

### Specific Web features

# 21





## LESSON 7.1. 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.

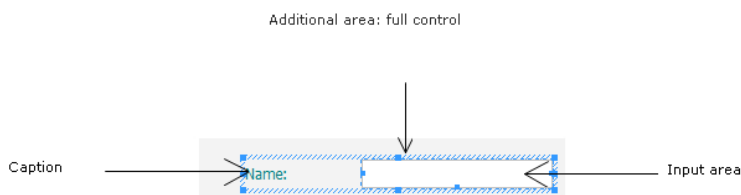
## Discovering the styles

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 the 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 area of the control.

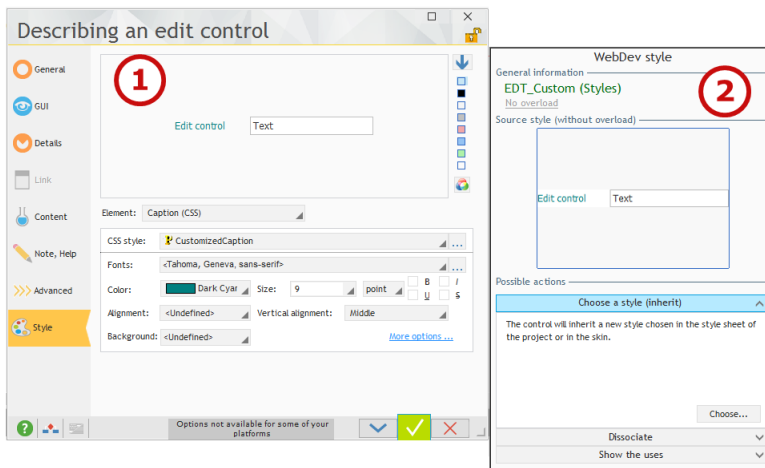
## Discovering the styles via an example

- To discover the styles in WEBDEV, a simple example was prepared for you:
1. Start WEBDEV 21 (if not already done). Close (if necessary) the current project in order 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)".

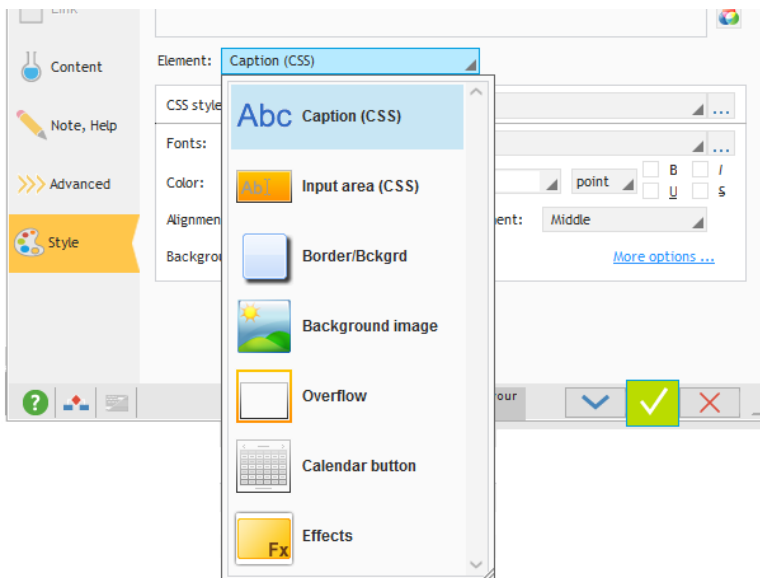
3. Open the "PAGE\_Styles" page in the editor.
4. Display the description window of the edit control "Name" (display the popup menu of the control and select "Description").
5. Select the "Style" tab. This tab is used to define the style characteristics of the control.
6. The following window is displayed:



This window allows you to manage the CSS styles (section 1 of the window) and the WEBDEV styles (section 2 of the 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:
  1. In the "Style" tab of the description window of the control, in the "Element" combo box, choose the element with the requested style.

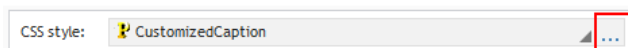


Note: All the elements marked "(CSS)" can use a CSS style directly: all you have to do is select its name.

2. The "CSS style" combo box is used to choose the CSS style of the element: this style will be applied to the element.

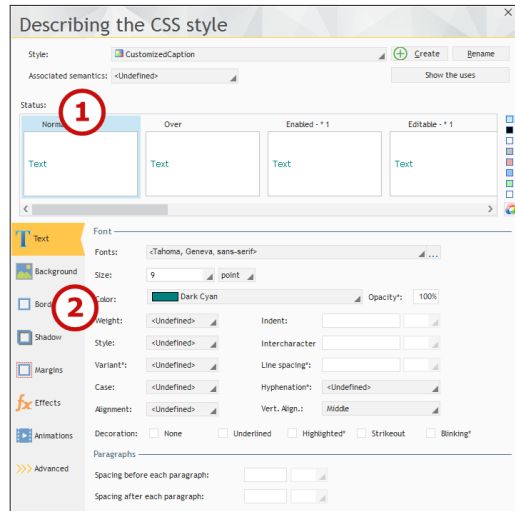
## Editing the CSS style of a control

- To create or edit a CSS style:
  1. In the "Style" tab of the description window of the control, click the "..." found beside the "CSS style" combo box.





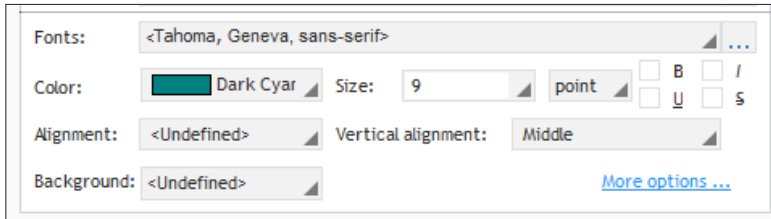
2. The window for editing the CSS styles is opened.



3. 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 the 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":
  1. Select the style "MyCustomStyle\_InputArea" in the combo box of section 1 if necessary.
  2. Select the "Normal" status (the first status of section 1).
  3. Click the "Background" tab: the background color is white.
  4. Select the "Editable" status (the 4th status of section 1).
  5. In the "Background" tab, the background color of the input area is pastel yellow. **The CSS style contains the style for each one of the states.**
  6. Close the description window of the CSS style.

- The "Style" tab found in the description window of the control is redisplayed. In the bottom section of the window, you have the ability to quickly access the style elements that are frequently modified:



The screenshot shows a style configuration window with the following options:

- Fonts:** A dropdown menu showing "<Tahoma, Geneva, sans-serif>" and a small menu icon.
- Color:** A color selection box showing "Dark Cyar" and a small menu icon.
- Size:** A text input field showing "9" and a small menu icon.
- point:** A text input field showing "point" and a small menu icon.
- Alignment:** A dropdown menu showing "<Undefined>" and a small menu icon.
- Vertical alignment:** A dropdown menu showing "Middle" and a small menu icon.
- Background:** A dropdown menu showing "<Undefined>" and a small menu icon.
- More options ...** A blue link.
- Formatting icons:** A set of icons for Bold (B), Italic (I), Underline (U), and Strikethrough (ABC).

- 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.

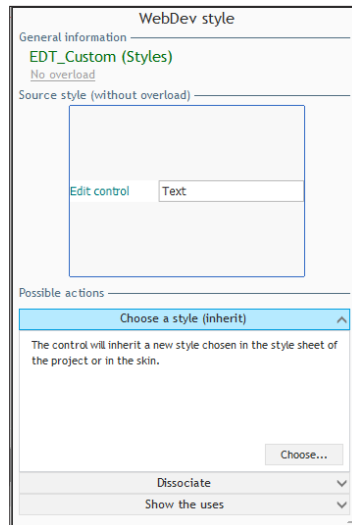
Let's now discover the WEBDEV styles.

## 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 area of the control.

- To manage the WEBDEV styles, use the panel displayed on the right of the "Style" tab of the control:



The name of the WEBDEV style appears at the top of this window. You can:

- Choose an existing style,
- Add the style of the current control to the style sheet of the project.  
This option allows you to re-use this style in other controls of the project.
- Dissociate this control.  
This option prevents the future style modifications from being applied to this control. This option is not recommended: we advise you to use the 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 the "Style" tab.
- Select "Bold" in the bottom area.

This modification will not dissociate the WEBDEV style from the control: an overload of the Bold property will be performed 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

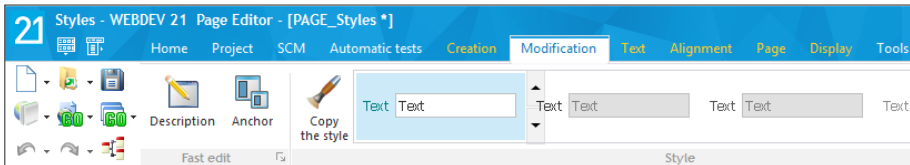
To see the overloaded elements, all you have to do is click at the top of the panel for managing the WEBDEV styles on the link that indicates the number of overloads performed for the control.

## Implementing styles: a practical example

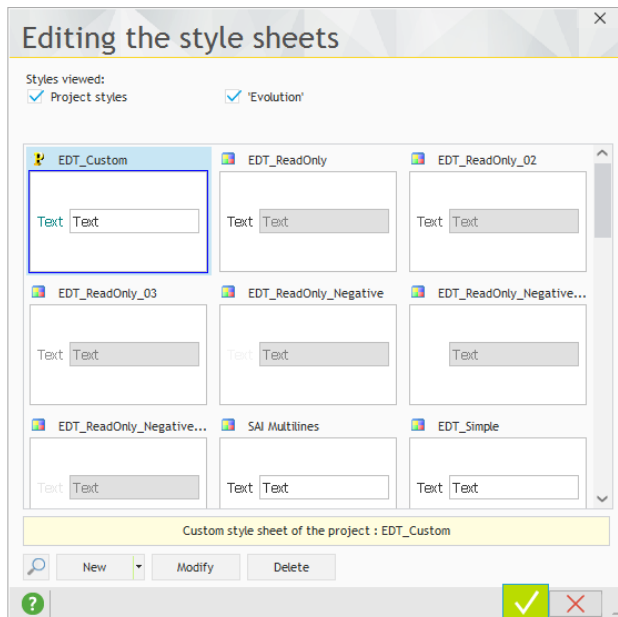
Enough theory, let's get down to work. We are going to create an edit control and to modify its styles in the "PAGE\_Styles" page of the "Styles" project.


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 the control (as already seen).
  - Via the "Modification" pane of the ribbon: a preview of the different WEBDEV styles available for the control is displayed.




- By displaying the popup menu of the 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 the control caption changes: the style was applied.
- ▶ Save the page (CTRL S) and run the test of the page (  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 the created control.
  2. In the "Style" tab, choose the element "Input area (CSS)".
  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 the editable inside area was overloaded.
  8. Validate the description window of the control.
- ▶ Save the page (CTRL S) and run the test of the page (  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 the input area in order to change the color of the text typed. This modification will be performed for all the edit controls of the page.
  1. Display the description window of the created control (ALT + Enter).
  2. In the "Style" tab, choose the element "Input area (CSS)".
  3. Click the [...] button on the right of the name of the 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 the control.
- ▶ Save the page (CTRL S) and run the test of the page (  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

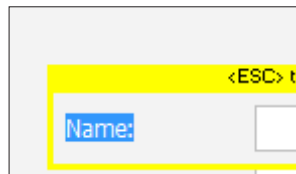
You also have the ability to define rich text in the elements!

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.

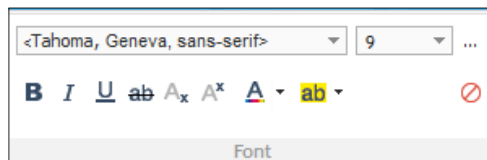
Let's take a simple example: adding a red asterisk to specify that a control is a mandatory control.

► In our example:

1. Select the "Name" control.
2. Press the SPACE or ENTER key on the keyboard.
3. The text of the caption becomes editable.



4. Add a star at the end of the caption.
5. Select this star.
6. On the "Text" pane, in the "Font" group:



- Click **A\*** to move the star in exponent.
- Change the color of the text to red with the button **A ▾**.

## LESSON 7.2. THE COOKIES

**This lesson will teach you the following concepts...**

---

- What is a cookie?
- How to manage the cookies?



Estimated time: 30 min

## What is a cookie?

A cookie is an easy way to temporarily store an information on the computer of the 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 the Web user when its lifetime is exceeded.

Caution: using cookies is possible only if the browser of the Web user is configured to accept the cookies.

A cookie is used to store on the computer of the 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 the 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 unencrypted 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 the Web user (in most cases, in the Internet "cache" of the browser) during a specific duration. The cookie is created by the browser or by the server.

The following elements are required to store an information in a cookie:

- Name of the cookie, used by the site to identify the cookie.
- Text of the cookie, corresponding to the information registered 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 the 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 link found below the name of the unit example.
  5. The page corresponding to the unit example is displayed in the editor.
  6. Save the page (  or CTRL S)

## How to use the cookies?

---

The cookies can be used according to two different modes:

- The cookies used in Server code.
- The 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**: used to send a cookie when the HTML page is displayed in the browser of the Web user.
- **CookieRead**: used to retrieve the value of a cookie saved on the computer of the Web user.

The "PAGE\_Cookies" example page presents an example for reading and writing a cookie in server code and in browser code.

See the online help for more details.

## LESSON 7.3. SECURE TRANSACTION AND PAYMENT

**This lesson will teach you the following concepts...**

- Securing the information and the pages via TSL/SSL
- Secure payment



Estimated time: 30 min

## 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.

The information does no longer flow via the HTTP protocol but via the HTTPS protocol (<https://customers.mywebdevsite.com/customers> for example).

### 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 (keyword: "SSL") for more details.

### 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 entering the 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 (keyword: "SSLActive") for more details.

## Secure payment

---

### Overview

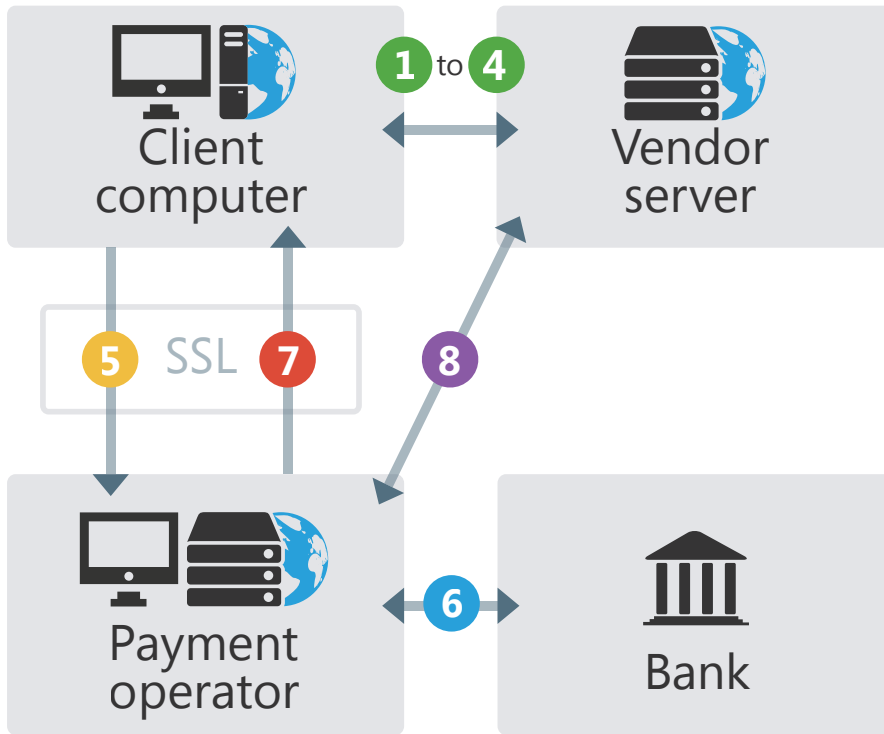
Most of the business sites allowing the Web users to perform an online purchase are using a system for online payment by credit card.

The secure payment is an essential feature for a business site. The payment solution must reassure the Web user (the "customer") and must guarantee the payment to the business site.

The data exchanged during this transaction must be secured (via the SSL protocol that was described in the previous paragraph for example).

Several payment solutions are available (PayBox, ATOS, CyberMut, CyperPaiement, SPPlus, ...).

The principle for secure payment may slightly change from a provider to another one but it is overall the same:



- |  |  |
|--|--|
| <b>1</b> View the site<br>Fill the basket              | <b>5</b> Enter the credit<br>card number |
| <b>2</b> Identification<br>(input of personal details) | <b>6</b> Bank authorization              |
| <b>3</b> Check the<br>order                            | <b>7</b> Response of the bank            |
| <b>4</b> Redirection to the<br>secure payment          | <b>8</b> Result of the transaction       |

- 1 to 4. Preparing the order on the business site: the Web user places his order on the site. During the payment operation, the business site transfers the information used to identify the order (vendor number, invoice amount, ...) to the payment operator.

- 5 to 7. Entering and checking the credit card number: the Web user enters his credit card number in a secure payment page. The data transmission is protected via SSL to ensure the confidentiality of data. The business site does not know the credit card number entered by the Web user.
- 8. Back to the business site: the payment operator indicates to the business site whether the payment was validated, canceled or refused.

**Caution !**

Important: the field of online payment 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. Implement the business site.

### Using the component supplied with WEBDEV: "Secure 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 check 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 min

## Overview

---

An anchor is used to define the behavior of a control or area when resizing the browser. Will the control be enlarged? Will it move?

This allows you to define the behavior of pages during the display on specific resolutions (tablets, phones, ...).

The anchors can be easily implemented via an option found in the popup menu of controls.

We are now going to use the anchors via an example.

We are going to create a project containing a page used to implement the 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 accessed at any time to check the validity of the 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. Save the page (  or CTRL S). The title of this page being "Anchor", its name "PAGE\_Anchor" is automatically proposed. Validate the backup window.



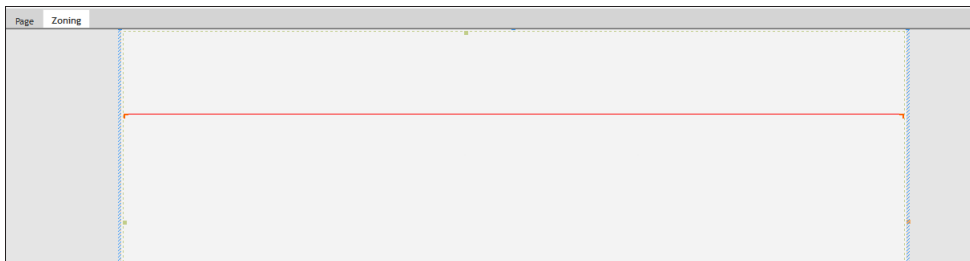
We are now going to divide our page into 3 areas:

- A header (at the top on the entire width).
- A menu on the left (on the entire height).
- A content area in the middle.

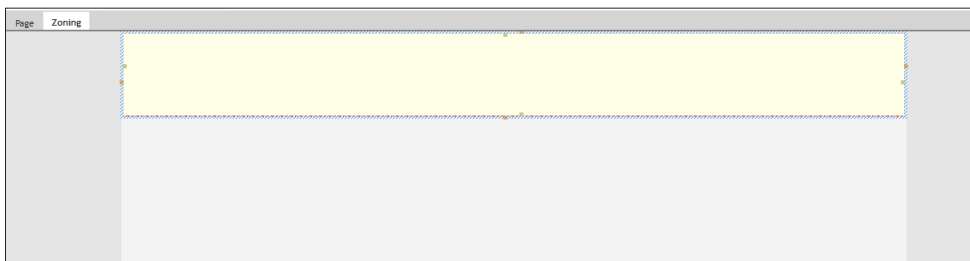
Each area will have a specific behavior when resizing the page.

To create the areas, we will be using the zoning feature.

- ▶ To create a first title area at the top of the 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 the page and draw a horizontal line. The area is created when the mouse button is released.

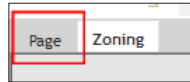


- ▶ To view this area, we are going to associate it with a background color:
  1. Display the description window of the area: select "Description of the area" from the popup menu.
  2. In the "Style" tab, select the "Border/Bckgrd" element if necessary and select a background color (tooltip yellow for example).
  3. Validate the description window of the area.

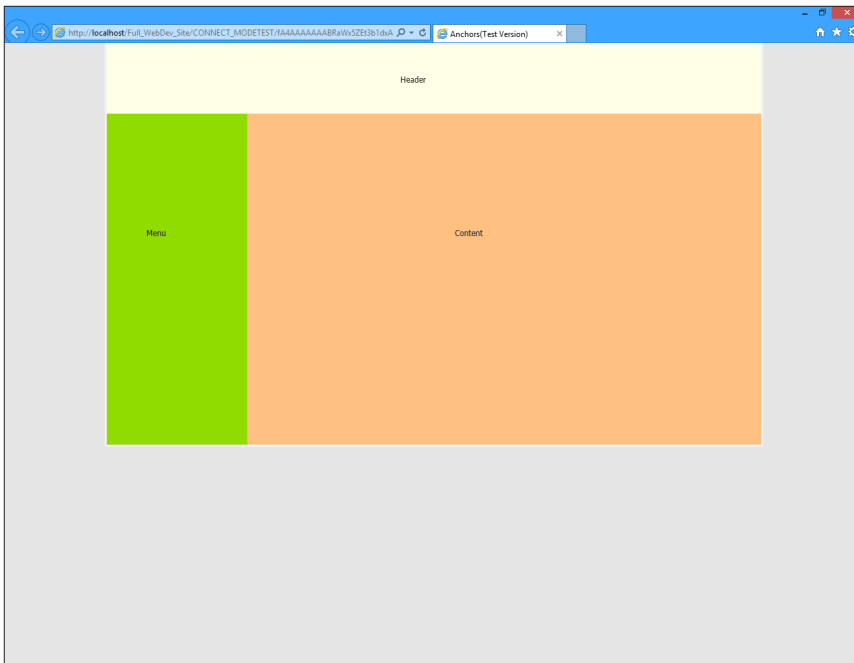


- ▶ We are going to perform the same operations to split the bottom area of our page:
  1. On the "Page" tab, in the "Edit" area, expand "Zoning" and select "Split the layout".
  2. The mouse cursor turns into a pen.
  3. Click in the middle of the bottom area and draw a vertical line.

- To view these areas, associate the green color with the bottom area on the left and the orange color with 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 in the yellow area. The Static control is automatically created.
  4. Modify the caption ([SPACE] key on the keyboard): the caption of this control 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 test of the page (🚀 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 anchoring

First, we are going to implement the anchoring of areas then the anchoring of controls in the areas.

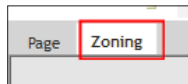
We want to get the following behavior when the browser is enlarged:

- 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 want them to remain centered in relation to their area.

We are now going to perform all the necessary operations.

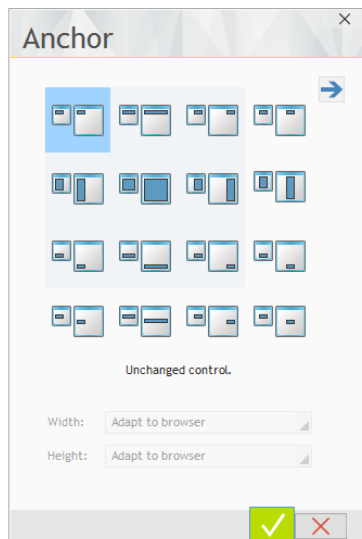
► To implement the anchoring of areas:

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 managing the anchors is displayed.

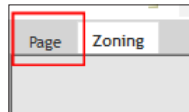





4. The area must be enlarged in width: select "Width" (  ) and validate.


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 managing the anchors. The area must be enlarged in width and in height: select "Width and height" (  ) and validate.

► To implement the anchoring of controls:

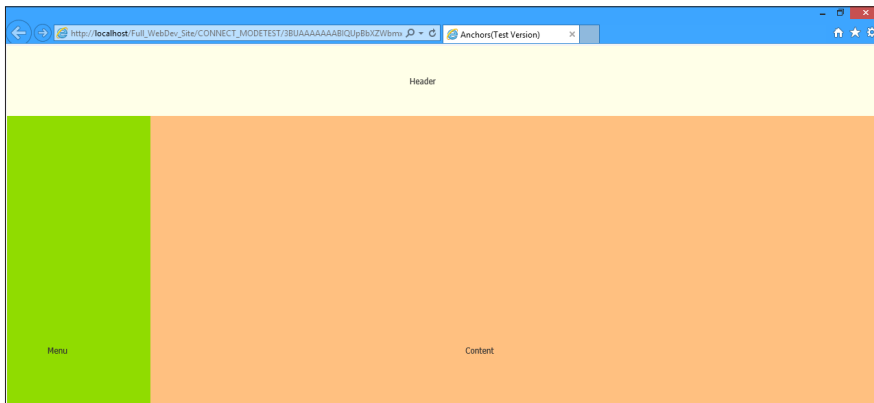
1. Display the page in Page mode: click the "Page" tab at the top of the page.



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" static control must be centered in height (  ).
  - The "Content" static control must be centered in width and in height (  ).

► Save the page (CTRL S) and run the test of the page (  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.

DEVELOP 10 TIMES FASTER

## PART 8

Useful  
features of  
WEBDEV

# 21





## LESSON 8.1. 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 min

## Overview

---

An internal component is a grouping of project elements. This grouping 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 the objects, procedures or methods made visible by the creator of the component.

Creating a component is **child's play**.

How to proceed? Do as usual, 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 a newsletter in a Web site.

This component includes:

- a page, used to view the list of persons who subscribed to the newsletter,
- 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 are going to use 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)".




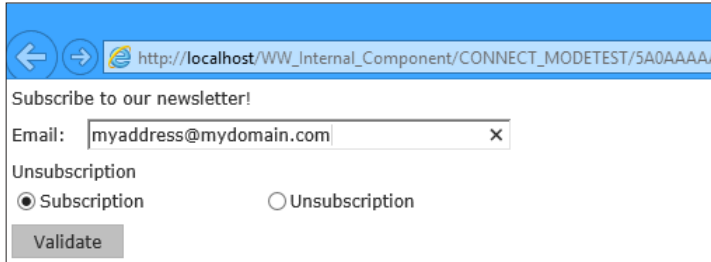
Notes

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



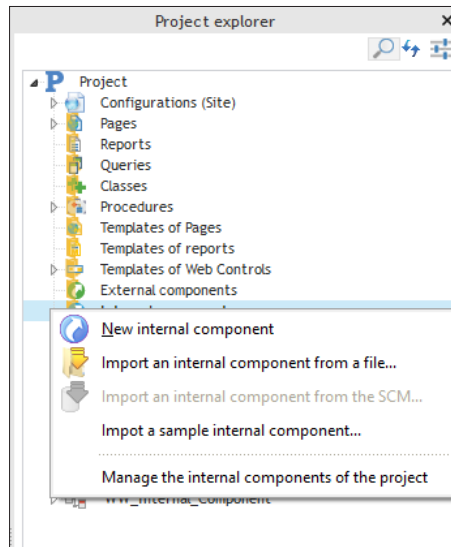
- We are going to test of the control template of this project:
1. Display the Web control template named "TPLC\_NewsletterSubscription" (double-click its name in the project explorer).
  2. Run the test of the Web control template (  among the quick access buttons).
  3. Enter an email address.



4. Click "Validate" to add the address into the data file.
5. Close the browser and go back to the editor.
6. The content of the data file can be checked by WDMaP (accessible in the "Tools" pane of the ribbon).

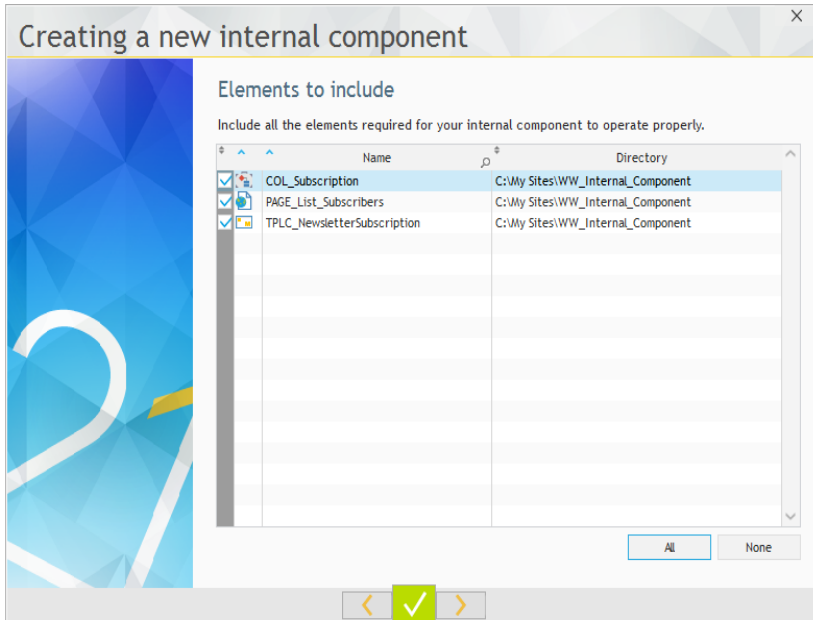
We are now going to create our internal component.

- To create an internal component:
1. In the project explorer:
    - Select the "Internal components" folder.
    - Display the popup menu (right mouse click).

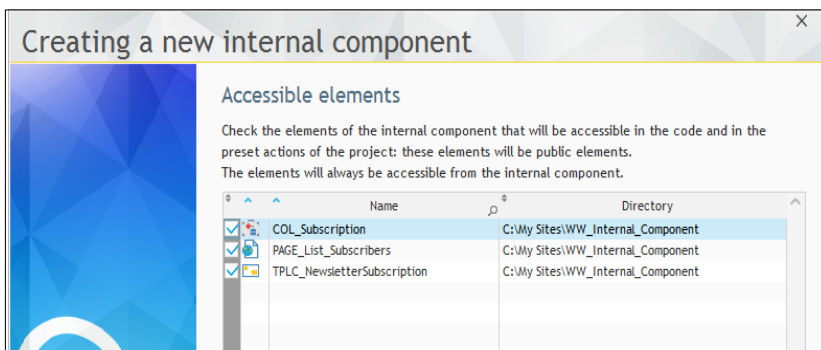


- Select "New internal component".
- The wizard for creating an internal component starts.

2. Go to the next wizard screen.
3. Identify your component: enter the name "InternalComponentNewsletterSubscription". The caption of the component is automatically proposed.
4. Go to the next step.
5. Select the elements that must be included in the component. In our case, all the elements must be selected.

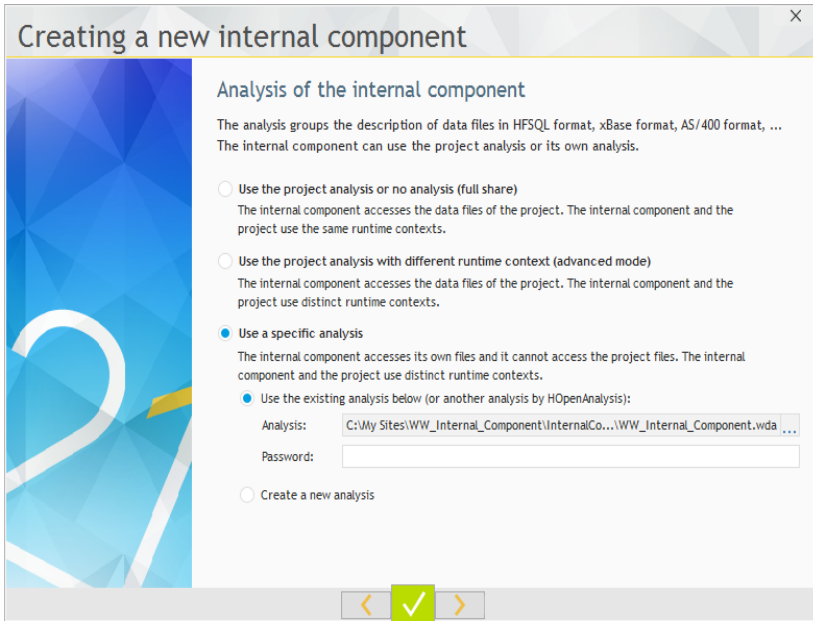


6. Go to the next step.
7. The wizard asks you to select the component elements that will be accessible from the client project.  
In our case, all the elements must be selected.



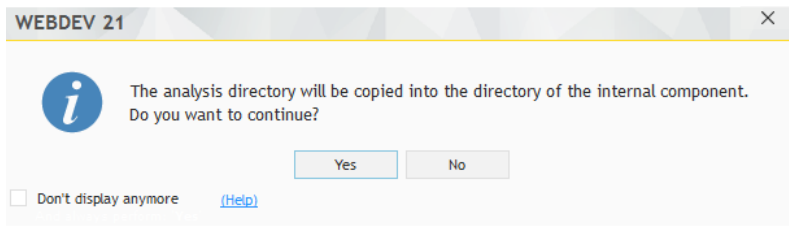
8. Go to the next step.

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 the project).

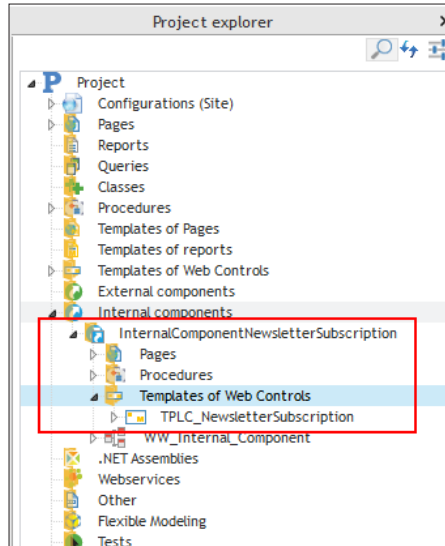
WEBDEV proposes to copy the analysis directory into the directory of the internal component.



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 for the 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 (keyword: "Internal component") for more details.

In our example, we are going to import the internal component into the "Full\_WebDev\_Site" project.

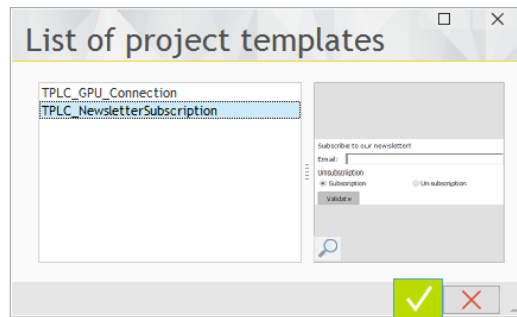
### ► Open the "Full\_WebDev\_Site" project.

1. Close the current project if necessary. The home window is displayed.

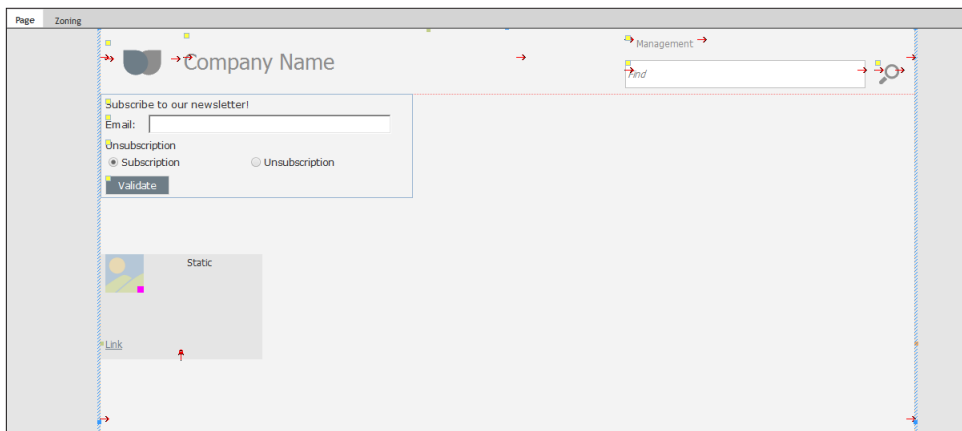
2. In the home window, click "Tutorial" and select "Full WebDev site (With pages)". The project is loaded.

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)".

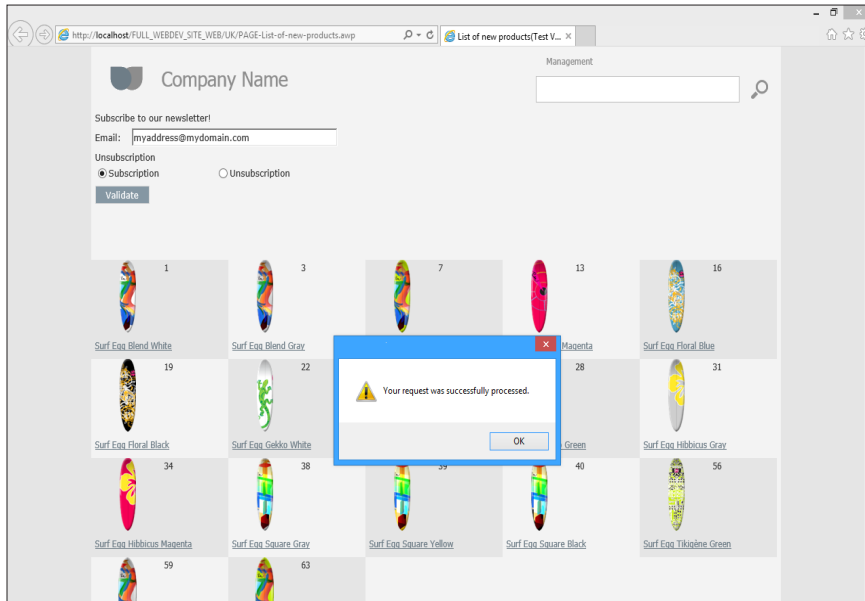
- ▶ Include the internal component in the project.
  1. In the ribbon, on the "Project" pane, in the "Project" group, expand "Import" and select "Import an internal component .. From a file".
  2. Select the "InternalComponentNewsletterSubscription.wci" file (found in the "WW\_Internal\_Component\InternalComponentNewsletterSubscription" sub-directory of the 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. Select the "Creation" pane of the ribbon and in the "Containers" group, click "Control template".
  3. The window for selecting the template to instantiate is displayed.



4. Select the template coming from the internal component and validate.
5. 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 test of the page (Go among the quick access buttons).
  2. Enter an email address.
  3. Click "Validate".
  4. A message confirms the registration.



## LESSON 8.2. 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 min

## 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:

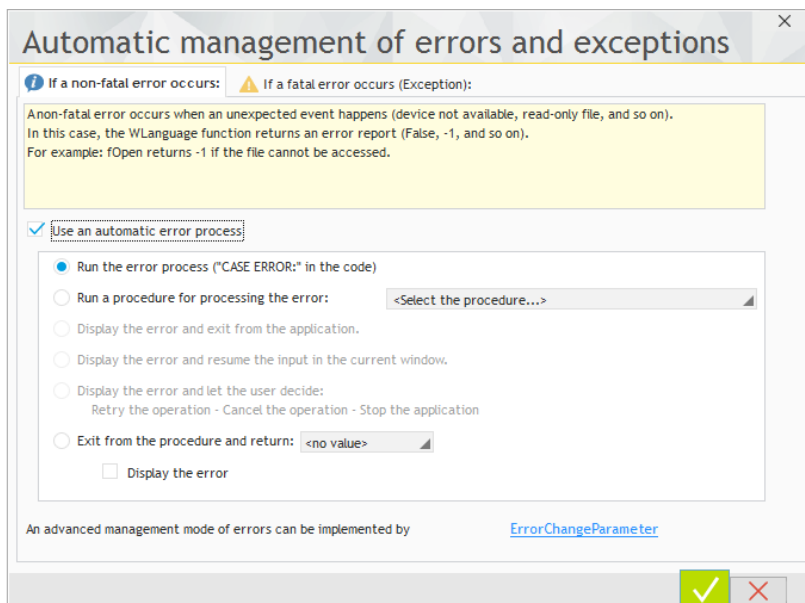
- a return value of error is returned by the function (**fOpen** returns "-1" if the specified file was not opened for example).
- the error is detected by WLanguage (the **ErrorDetected** variable is set to True) and the error details are returned by **ErrorInfo**.

This second operation can be automatically managed by 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 "If error: by program" link in the code editor:



- by programming with **ErrorChangeParameter**.



## Types of affected errors

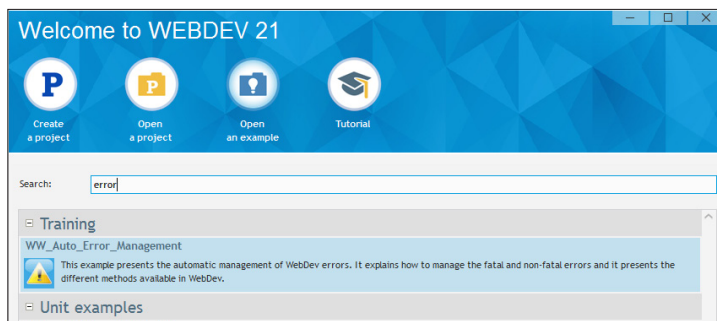
Two types of errors can occur in WLanguage:

- the "non-fatal" errors (or 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 development problems (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.

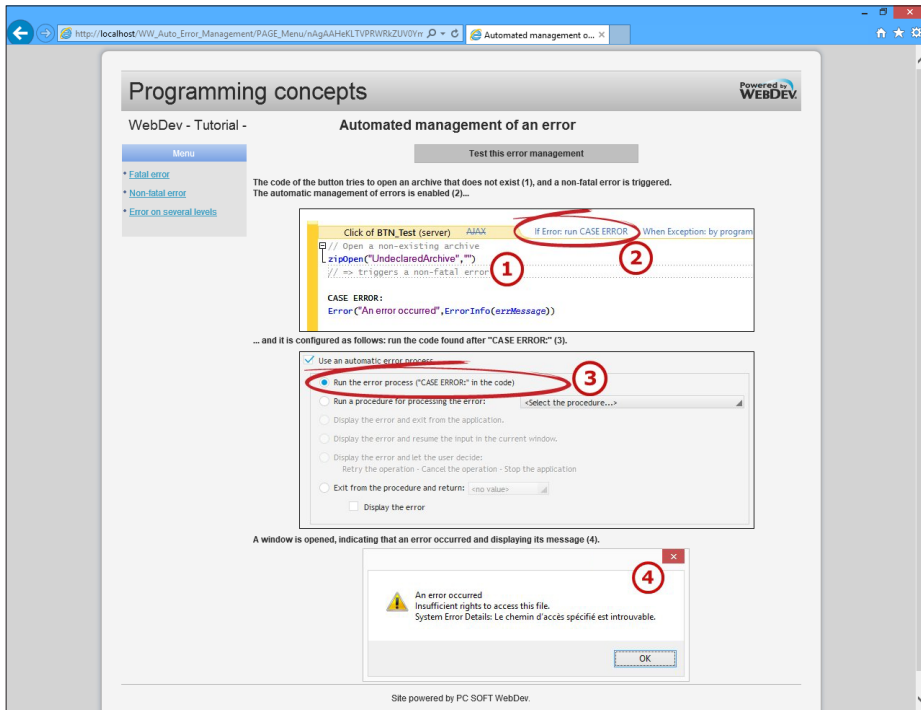
## Automatic management of errors: a training example

- 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 full 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.

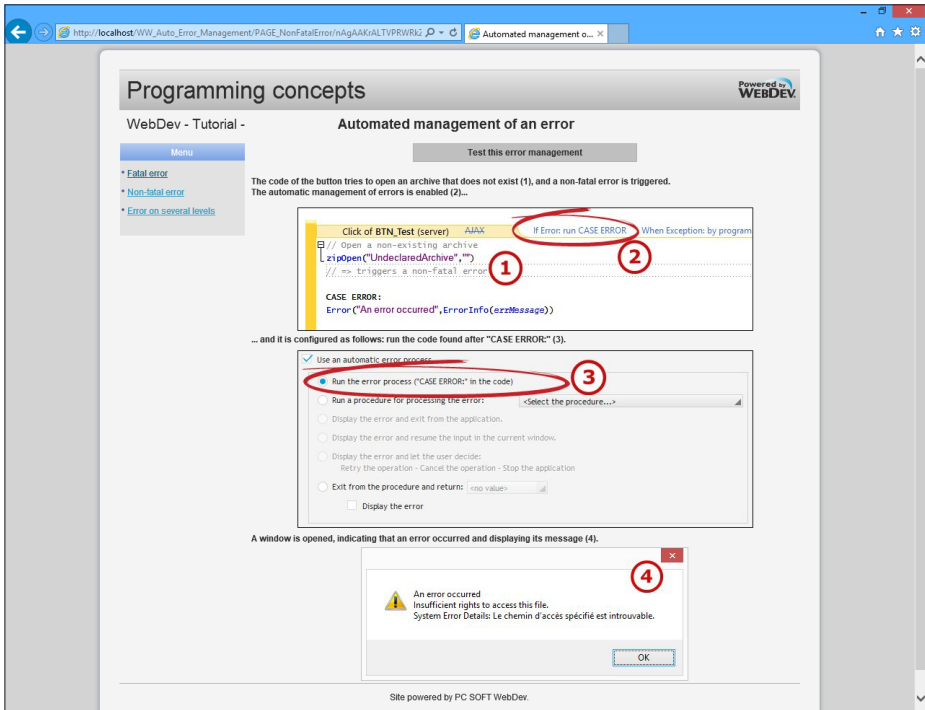
- ▶ This project presents the management:
  - of a non-fatal error (opening an archive that does not exist).
  - of a fatal error (division by 0).
  - of an error on several levels.
- ▶ Run the project test by clicking  (among the quick access buttons).
  1. Click "Non-fatal error".
  2. The following 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.



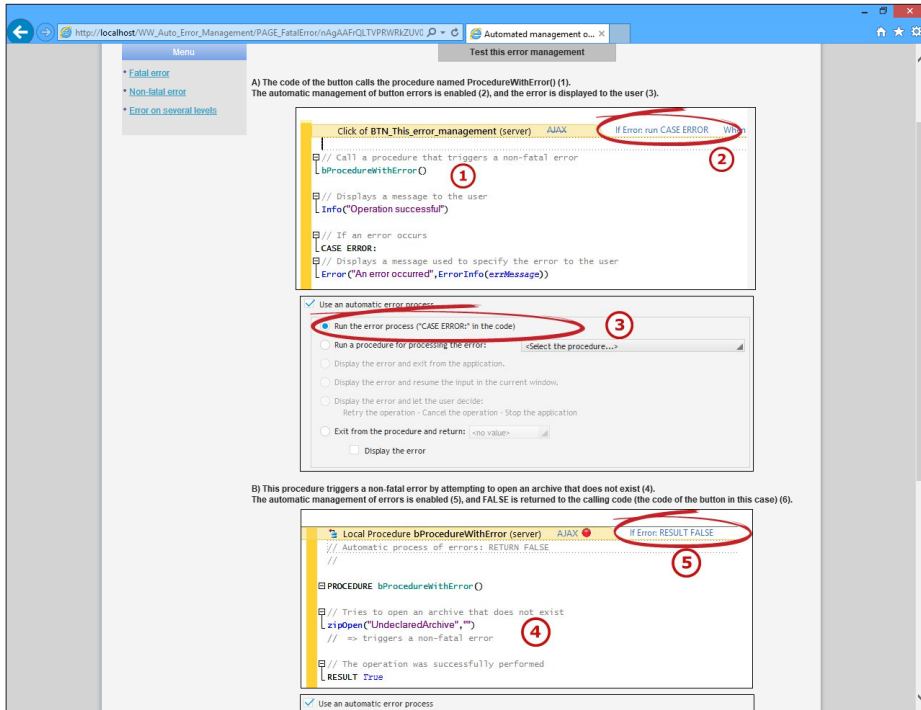
2. Click the "Test this error management" button.

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 (keyword: HFSQL, Managing the errors) for more details.

## LESSON 8.3. 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 min

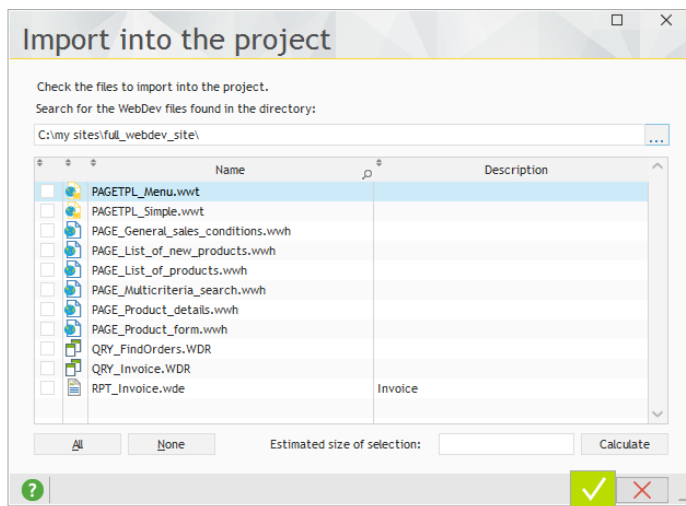
## Importing elements

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 the 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.

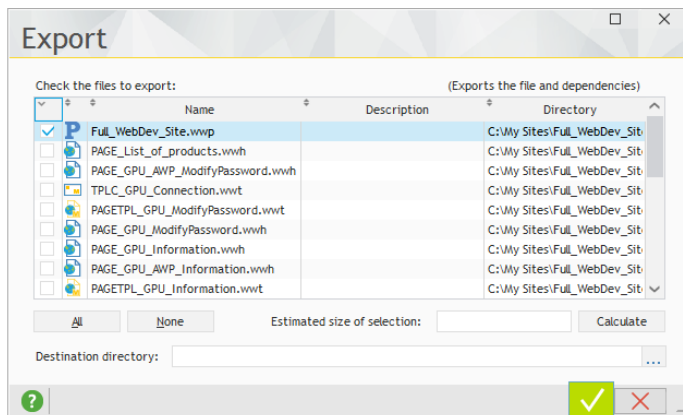


Caution!

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 the 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 the 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 without 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

You are interested in a page found in a non-WEBDEV site? 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. It is not a feature for "grabbing" Internet sites.

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 (an existing project or a new project for example).
  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 the HTML page. This page can be a file found on your computer ("..." button) or an Internet address (<http://www.mywebdevsite.eu/mapping.html> for example). 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.



DEVELOP 10 TIMES FASTER

## PART 9

# Optimizing and debugging a project

# 21





## LESSON 9.1. OVERVIEW

**This lesson will teach you the following concepts...**

---

- Why optimize a site?
- Example project



Estimated time: 5 min

## Overview

---

Your site is done. It operates. You want to deploy it.

Have you thought of using the WEBDEV tools to optimize your site? WEBDEV proposes several tools and features allowing you to quickly optimize your site and to easily avoid the predictable bugs.

This part 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 in order to display the home window.
- ▶ Open the "WW\_Optimization" project.  
To do so, in the home window, click "Tutorial" and select "Optimize a project".  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Optimize a project".

## LESSON 9.2. PROJECT AUDITS

**This lesson will teach you the following concepts...**

---

- What is an audit and what is its purpose?
- Starting and analyzing the static audit
- Starting and analyzing the dynamic audit



Estimated time: 20 min

## What is an audit?

The audits provide a set of features used to automatically improve the quality and the performances of a project and to follow the conditions in which it is implemented.

Two types of audits are available:

- **The static audit.** The static audit performs a detailed analysis of a project and elements. This audit is performed from the project editor.
- **The dynamic audit.** The dynamic audit analyzes 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 a feature of the environment 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".



Notes

The static audit of the project can also be started from the project dashboard, via the "Static audit and Compilation" widget.

To do so, you must:

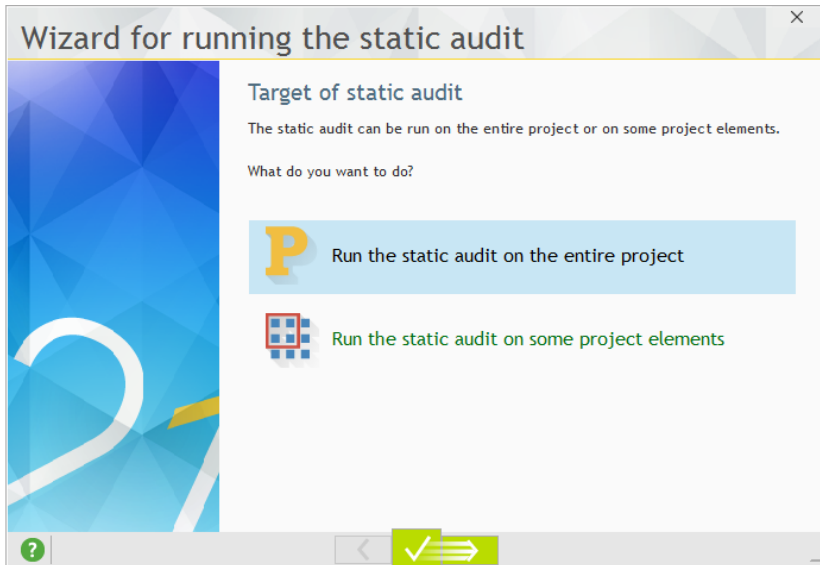
- enable the Widget if necessary (click the link "Click here to re-enable").
- click the arrow.

### Static audit and Compilation



2. The wizard starts. We are going to define the target of the static audit.

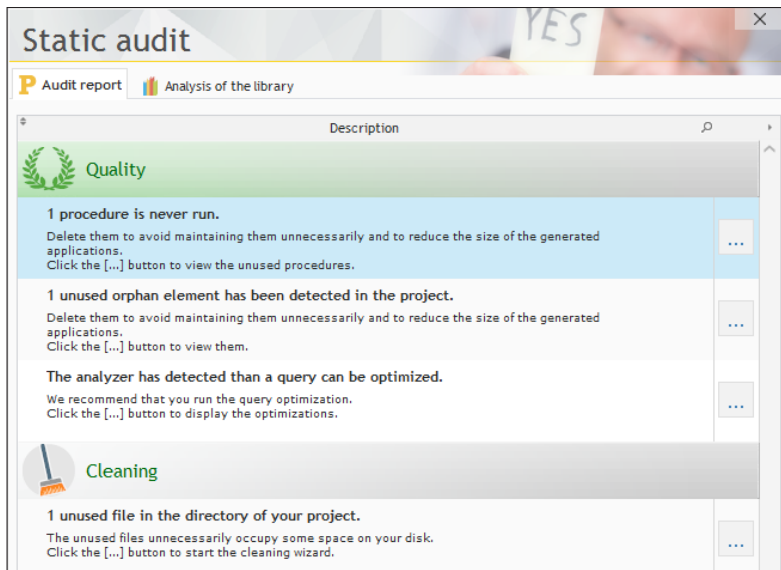




3. Select "Run the static audit on the entire project".

4. Validate the wizard.

5. The audit report is displayed:



The static audit includes:

- The static audit of the project.
- The audit of the content of the application library.
- Let's study the topics presented by this report.

## Procedure not run

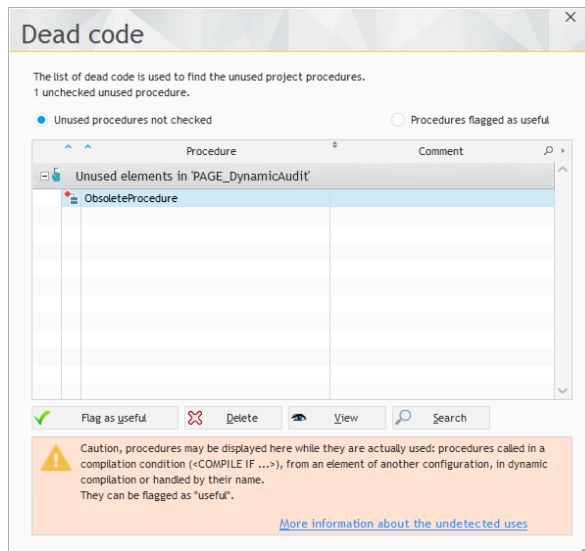
In our project, the audit indicates that a procedure is never run.

In large 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 used to list the dead code is displayed.



2. The "ObsoleteProcedure" procedure is never called. This window is used to:

- specify that the procedure is still used ("Mark as used" 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 the 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.

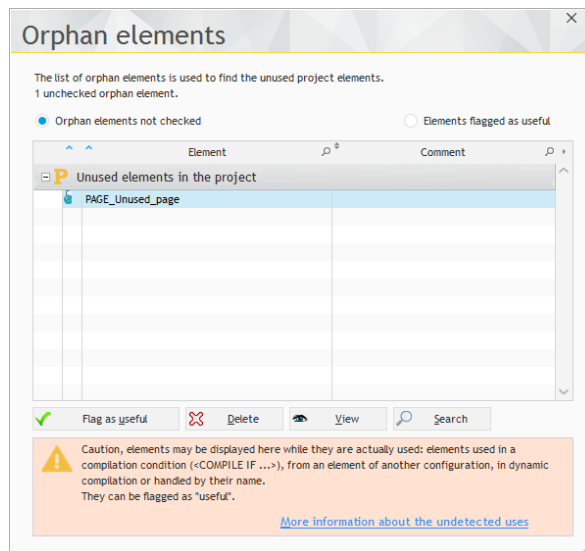
## Orphan element

The audit indicates 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 used to list 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 used" 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 the 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.

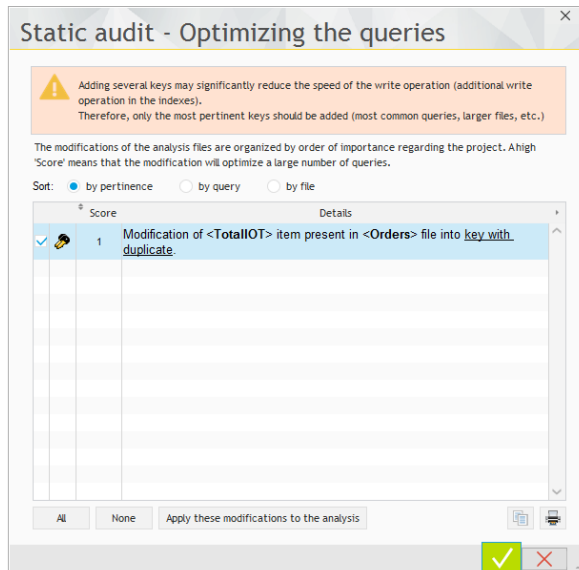
## Query to optimize

The audit indicates that our project contains a query to optimize.

This optimization is used to improve the execution speed of the query. The optimization of a query is based on the following principle: a wizard defines the composite keys that must be modified and/or added in the analysis associated with the project.

► To optimize the query:

1. Click the [...] button to get more details. The window used to list the modifications to perform is displayed.



2. The wizard proposes to modify an analysis item by changing it into a key with duplicates.
3. We are going to apply this change: click the "Apply these modifications to the analysis" button.
4. A window is displayed, asking you to confirm the modification of the analysis. Validate this window.
5. The data model editor is opened and the modification is performed. A message indicates that the analysis must be generated. Click "Yes" to confirm the generation of the analysis.
6. The project is using data so an automatic modification of the data files is required. Validate the different screens of the wizard for data modification (by keeping the default options).
7. The window of the static audit is automatically refreshed.

## 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 analyze the execution of the application. The audit is used to detect problems such as:

- Excessive memory consumption,
- Slowness of the 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.

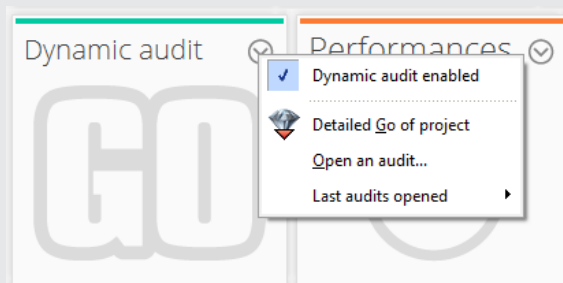
The dynamic audit and the project test will be started at the same time.

► To start the dynamic audit on the "WW\_Optimization" project:

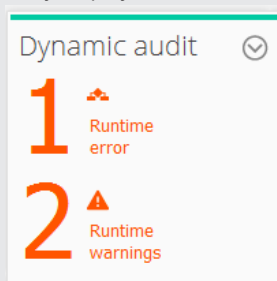
1. In the ribbon, on the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Debug the project while the audit enabled". The test of the project is run.

The dynamic audit of the project can also be started from the project dashboard, via the "Dynamic audit" widget. To do so, you must:

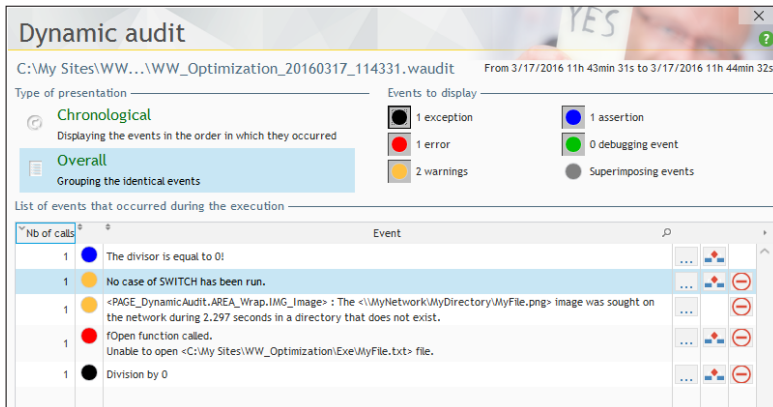
- enable the Widget if necessary (click the link "Click here to re-enable").
- expand the arrow and select "Detailed Go of project".



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. Note: 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 the 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. In this case, the chronological border allows you to see the position and importance of problems.
  - choose the type of problem to display (error, assertion, ...). This allows you for example 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 directly 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 in order to analyze the result. See the online help (keyword: "Dynamic audit") for more details.

## LESSON 9.3. PERFORMANCE PROFILER

**This lesson will teach you the following concepts...**

---

- Overview
- Starting the performance profiler
- Analyzing the result



Estimated time: 15 min



## Overview

---

The performance profiler (also called Profiler) is a tool used to check and optimize the execution time of the processes found in your site.

**Its 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 execution times of each one of the processes 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 directly:**

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 WEBDEV editor, all you have to do is exit from your site.

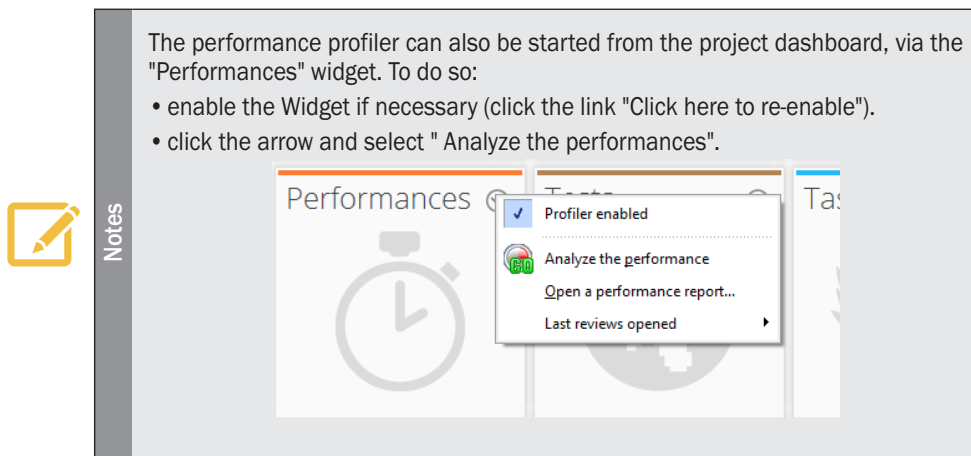
The performance profiler displays the result of the analysis. This result is saved in a WPF file.

- **from one of your processes in WLanguage**, via the following functions:

<b><i>ProfilerStart</i></b>	Starts "collecting data" for the performance profiler.
<b><i>ProfilerEnd</i></b>	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.

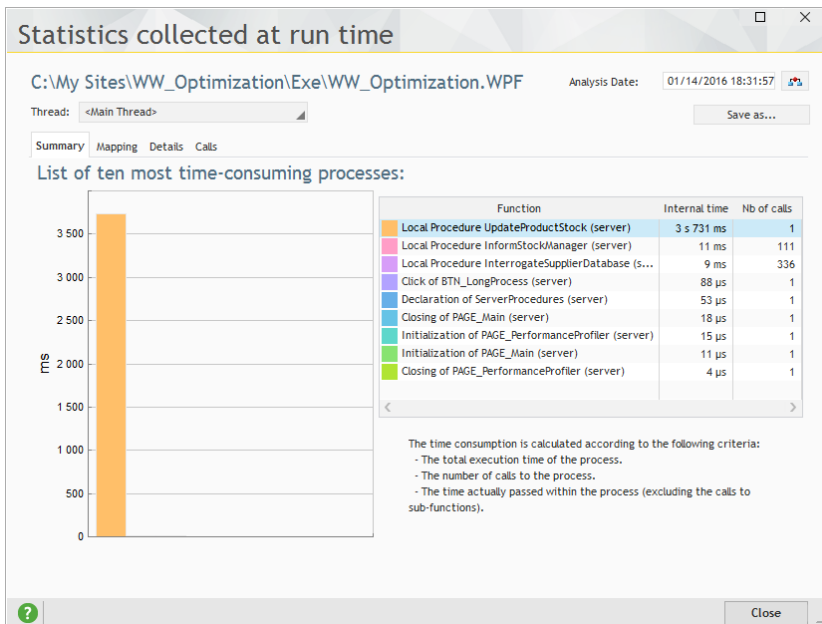
- The first method will be used in our example. To start the performance profiler on the "WW\_Optimization" project:
  1. In the ribbon, on the "Project" pane, in the "Audit and performances" group, expand "Analyze the performance" and select "Analyze the performance".



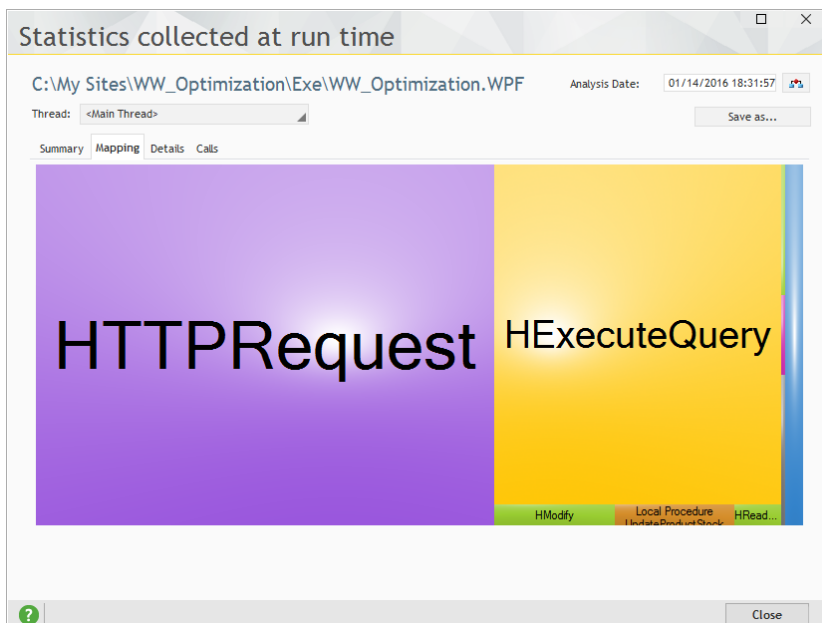
2. The test of the project is run.
3. Click the "Test page of performance profiler" button.
4. Click the "Process to analyze" button.
5. Validate the information window and stop the project test. The report window of the performance profiler is displayed.

## Analyzing the result

- Let's study the report window of the 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.

Statistics collected at run time

C:\My Sites\WW\_Optimization\Exe\WW\_Optimization.WPF Analysis Date: 01/14/2016 18:31:57

Thread: <Main Thread> Save as...

Summary Mapping Details Calls

Function	Total time	Internal time	Nb of calls	Time 1 call	code %	Parent
Execution completed	4 s 463 ms	613 ms	1	4 s 463 ms	13.7 %	
Click of BTN_LongProcess (server)	3 s 753 ms	88 µs	1	3 s 753 ms	0.0 %	PAGE_Performan
Local Procedure UpdateProductStock (server)	3 s 753 ms	3 s 731 ms	1	3 s 753 ms	0.9 %	PAGE_Performan
HTTPRequest()	2 s 291 ms	2 s 291 ms	1	2 s 291 ms		PAGE_Performan
HExecuteQuery()	1 s 351 ms	1 s 351 ms	336	4 ms		PAGE_Performan
Connecting to the 'WW_Optimization' site	94 ms	91 ms	1	94 ms	96.8 %	
HModify()	35 ms	35 ms	336	105 µs		PAGE_Performan
HReadNext - FOR EACH	13 ms	13 ms	336	40 µs		PAGE_Performan
Local Procedure InformStockManager (server)	11 ms	11 ms	111	105 µs	33.7 %	PAGE_Performan
Local Procedure InterrogateSupplierDatabase (serve	9 ms	9 ms	336	27 µs	79.1 %	PAGE_Performan
ListAdd()	6 ms	6 ms	111	58 µs		PAGE_Performan
HSavePosition - FOR EACH	4 ms	4 ms	1	4 ms		PAGE_Performan
PageDisplay("PAGE_Main")	2 ms	1 ms	1	2 ms		
sending the 'PAGE_PerformanceProfiler' page to the b	2 ms	2 ms	1	2 ms	100.0 %	
Random()	1 ms	1 ms	336	5 µs		PAGE_Performan
sending the 'PAGE_Main' page to the browser (intern:	1 ms	1 ms	1	1 ms	100.0 %	
StringBuild()	1 ms	1 ms	111	11 µs		PAGE_Performan
HReadFirst - FOR EACH	125 µs	125 µs	1	125 µs		PAGE_Performan
Declaration of ServerProcedures (server)	53 µs	53 µs	1	53 µs	100.0 %	ServerProcedur

☒ Show the WLanguage functions

Calls Print

Close

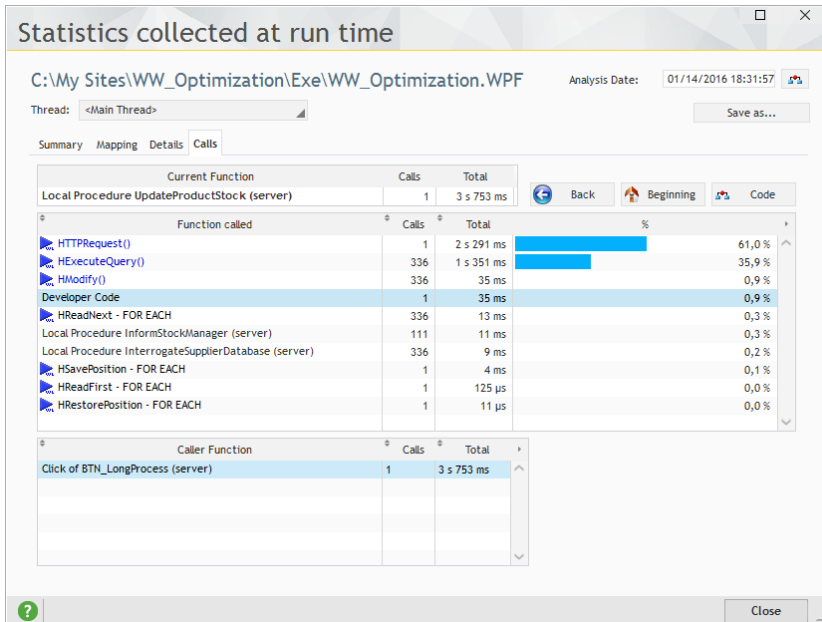
The following information is displayed for each process:

- Function: Function, process or procedure run.
- Total Time: Execution time of the function.
- Internal time: Execution time due to the engine.
- Nb calls: Number of calls made to the function (procedure or process).
- Time 1 call: Time for running 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 displays the call to **HTTPRequest** in the list of elements taking the longest time.

Select this line. We are going to check 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")
```

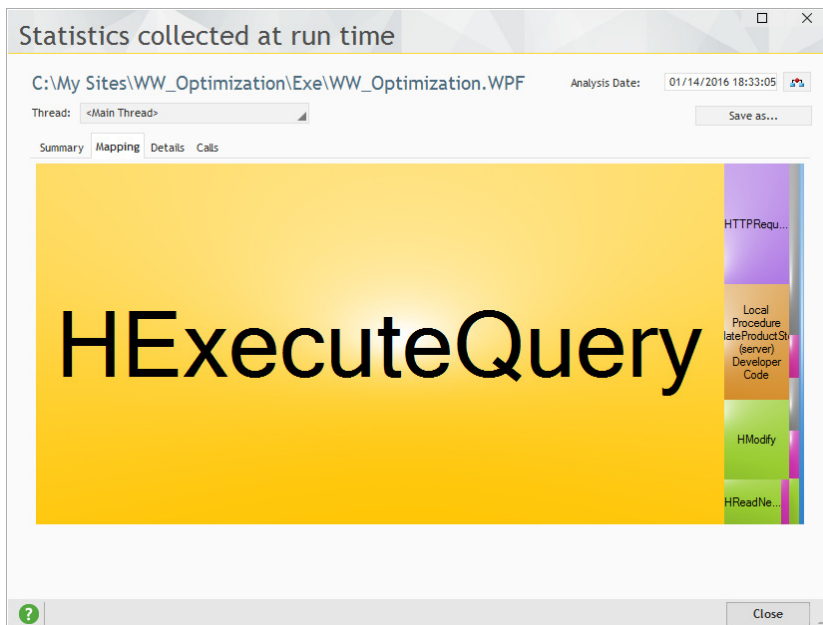
The slowdown is caused by the fact that the address specified for **HTTPRequest** is not accessible.

- Let's check the operating mode of the application by optimizing this code:
1. Replace the code line containing **HTTPRequest** by the following code line:

```
// Checks the accessibility of the supplier server
HTTPRequest ("http://windev.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 performances" and select "Analyze the performances".
  2. The test of the project 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 the 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 9.4. DEBUGGING A PROJECT

**This lesson will teach you the following concepts...**

---

- Overview
- Using the debugger



Estimated time: 15 min

## 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 becomes 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 the "Process to analyze" button: click in front of the first code line of the 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 the "PAGE\_PerformanceProfiler" page ( among the quick access buttons).

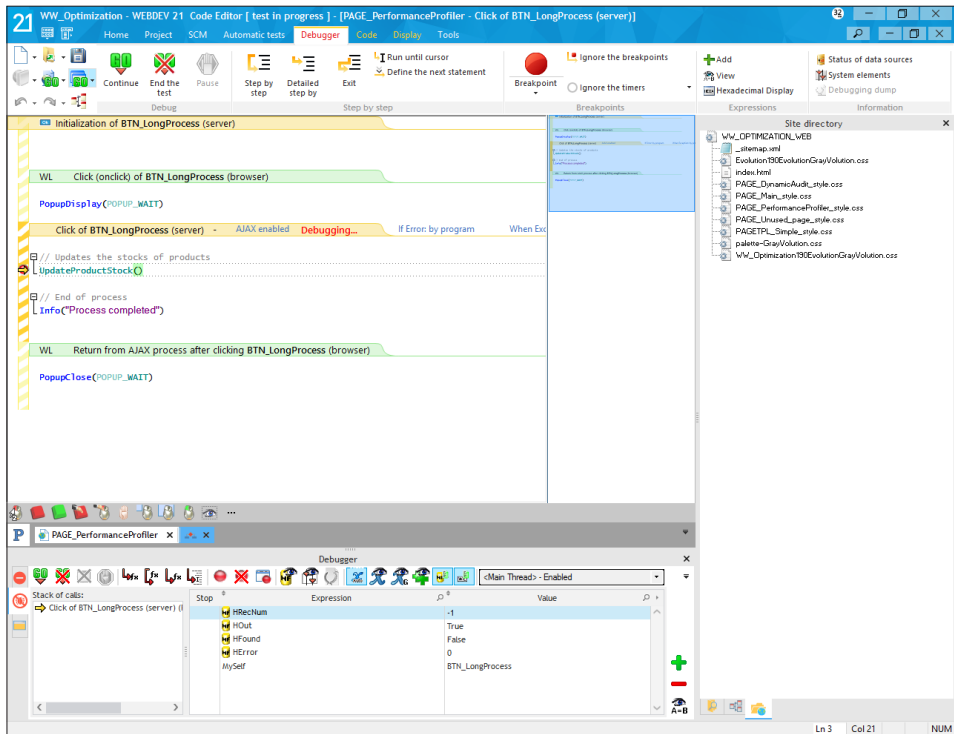


Note

Several methods can be used to start the debugger. See the online help (keyword: "Debugger, Run a test") for more details.

4. Click the "Process to analyze" button. The debugger starts: the code editor appears 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 the 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 in order to follow their evolution (this feature will be presented later).

We are going to perform some operations in the debugger to discover its capabilities.

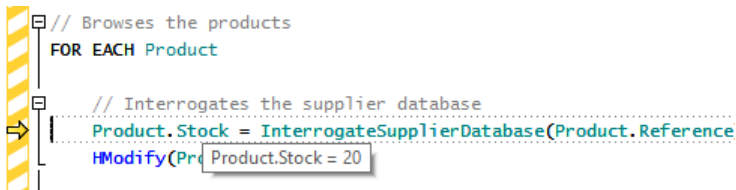
- First of all, we are going to run the different statements step by step and see the content of variables:

1. Press the [F7] key (or press the "Detailed step by step" button). The current line is automatically run by going into the sub-procedures if necessary.

Note: To avoid going into the sub-procedures, press the [F8] key or click the "Step by step" button.

The values of the variables are modified (if necessary) in the "Debugger" pane (displayed at the bottom of the screen).

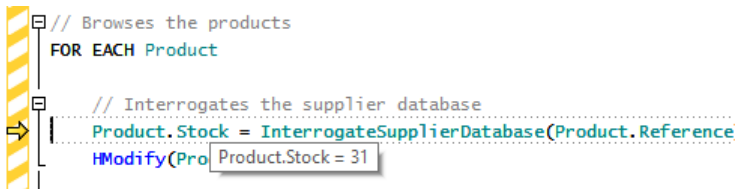
2. Keep pressing the [F7] key until you reach the line "Product.Stock = InterrogateSupplierDatabase ...", hover "Product.Stock" with the mouse cursor. A tooltip is displayed with the value of the expression:



3. The value of "Product.Stock" is displayed in a tooltip. This value corresponds to 20 because the code line was not run.

4. Press the [F8] key to run the line.

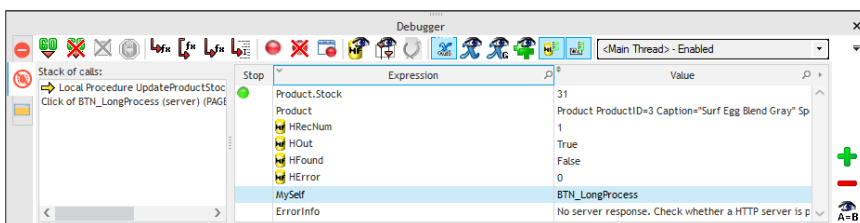
5. Hover "Product.Stock" again. The value of "Product.Stock" displayed in the tooltip corresponds to the result of the `InterrogateSupplierDatabase` procedure.



- 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 while 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 to 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:

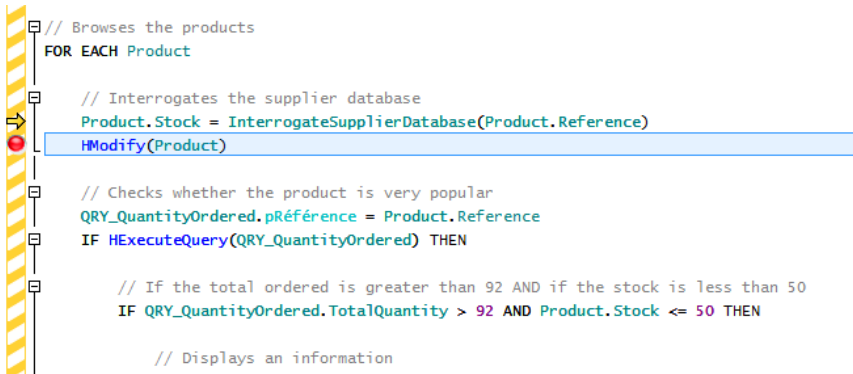
```
IF HExecuteQuery(QRY_QuantityOrdered) THEN
```

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.



The screenshot shows a code editor with a vertical margin on the left. A yellow and black hatched area is visible in the margin. A red bullet (breakpoint) is placed on the line containing **HModify(Product)**. The code is as follows:

```
// 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

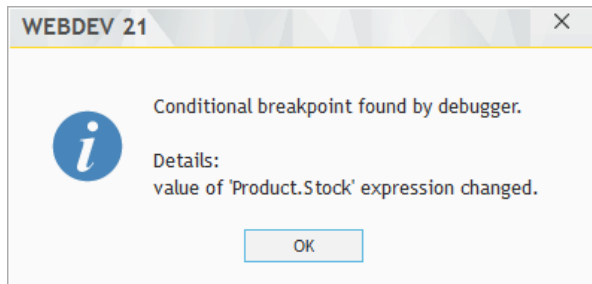
        // If the total ordered is greater than 92 AND if the stock is less than 50
        IF QRY_QuantityOrdered.TotalQuantity > 92 AND Product.Stock <= 50 THEN

            // Displays an information
```

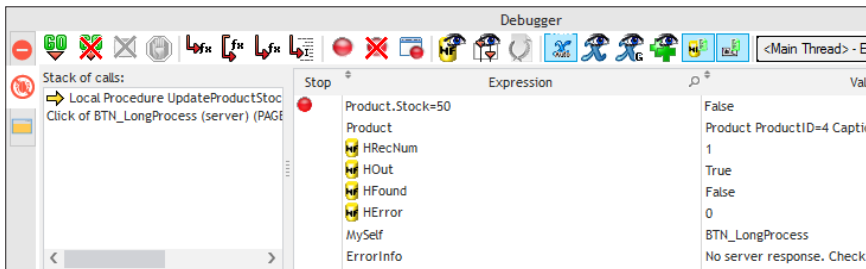
2. Press the [F5] key (or click the "Continue" button found in the ribbon). The code is run until it reaches the breakpoint. The arrow used to identify the current line moves until it reaches the breakpoint.

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 the [F5] key to continue the test.
  4. A message is displayed, indicating that the value of the "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 the Product.Stock variable is equal to 50.
- That's it, we presented the main features of the debugger. To stop the test in the debugger, click "End the test" found in the ribbon.

DEVELOP 10 TIMES FASTER

**PART 10**

**Managing  
a HFSQL Client/  
Server database**

21





## LESSON 10.1. INTRODUCTION

**This lesson will teach you the following concepts...**

---

- Principle of Client/Server
- Why switch an application to HFSQL Client/Server?



Estimated time: 5 min

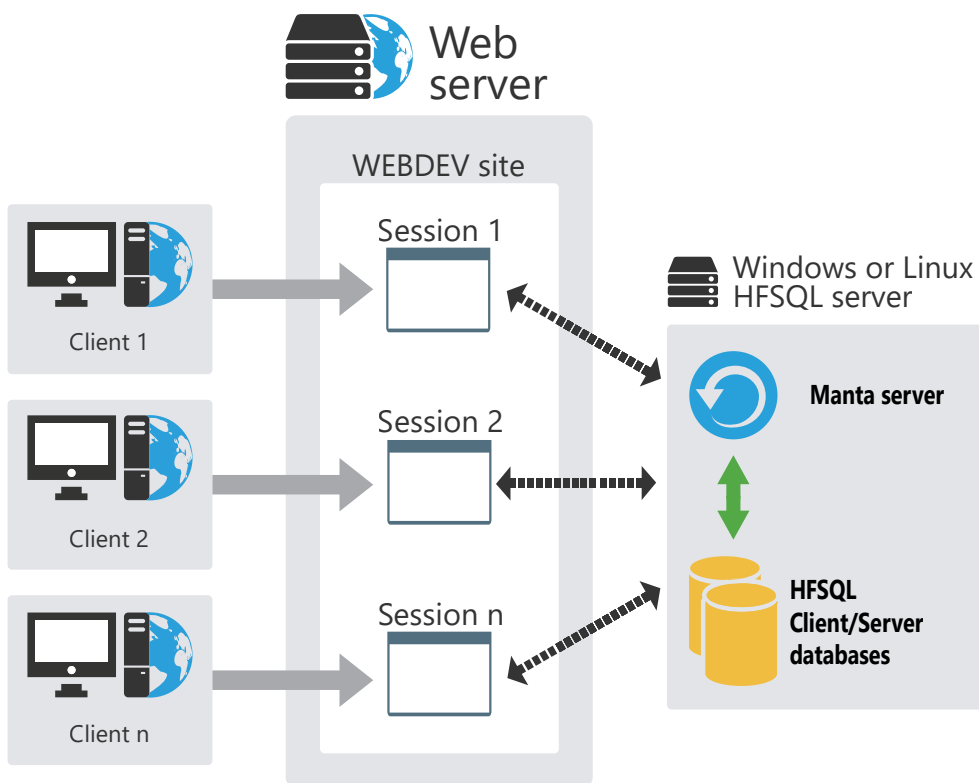
## Overview

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 the security of data,
- allows you to 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 a HFSQL Classic database so that it can use a 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 the native multi-user mode: the performances are optimized in multi-user mode.

## LESSON 10.2. 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 min

## 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 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 CD. If you do not own this CD, the setup of the HFSQL server is also available from our Web site ([www.windev.com](http://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 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 name and port of the server. The port 4900 will be used by default.



Notes

Don't forget to open this port on the firewall in order to connect to the HFSQL server from another computer.

8. Install the HFSQL Control Center if this one 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 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 the 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 ("HDescribeConnection") for more details.

## 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 this adaptation in the data model editor.
- perform this adaptation from 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

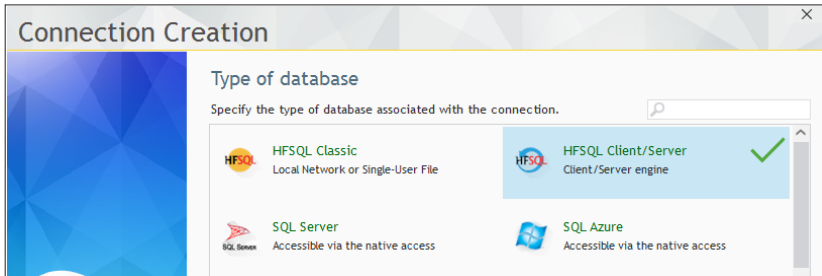


Answers

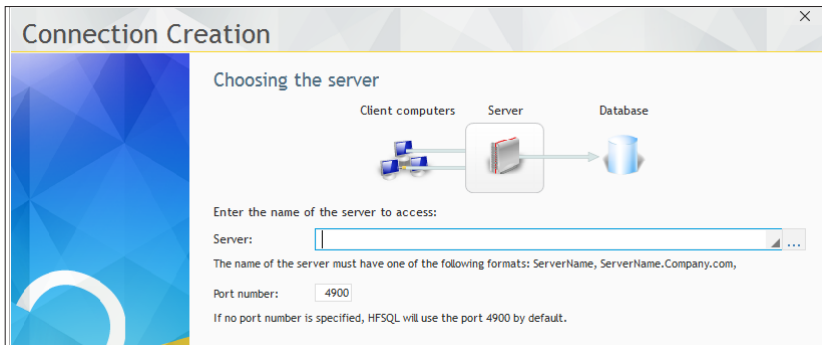
If you did not perform the operations in the previous lessons, you can follow this lesson by opening a corrected project: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Full WebDev Site (Answer)"

► 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.

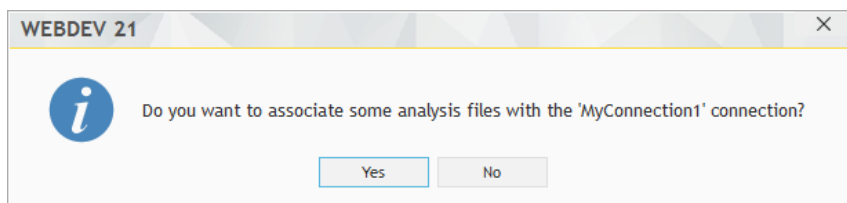


5. In the following planes, specify:



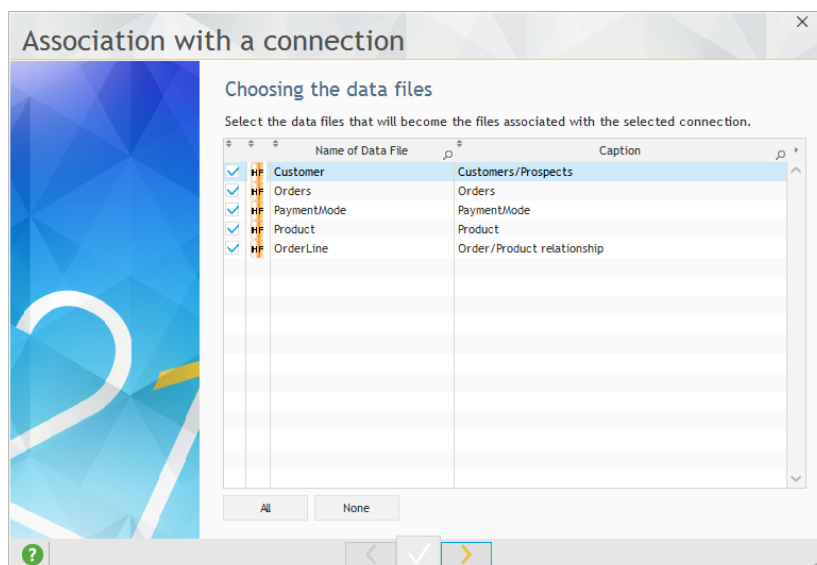
- the name of the server on which the HFSQL Client/Server server was installed (localhost if it was installed on your computer for example) and the port number. Go to the next screen.
  - the name of the user and his password (leave this information empty to use the administrator). Go to the next screen.
  - the name of the database ("Full\_WebDev\_Site" in our example). Go to the next screen.
6. Enter the name of the connection (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.



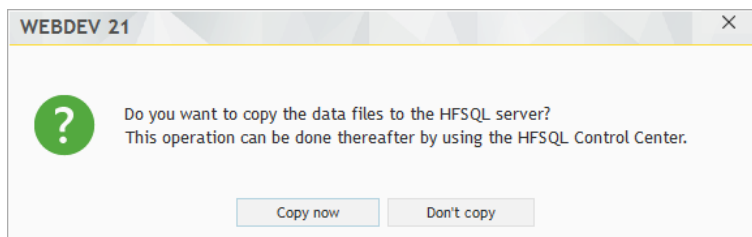
8. Click "Yes".

9. In the next screen, select all the proposed files:

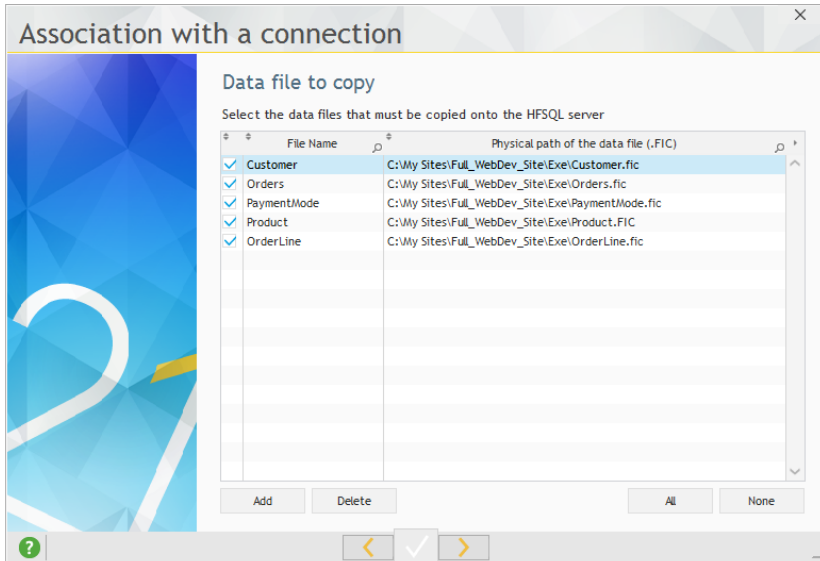


Go to the next step.

10. Then, the wizard proposes to copy the data files onto the server. Validate ("Copy now").

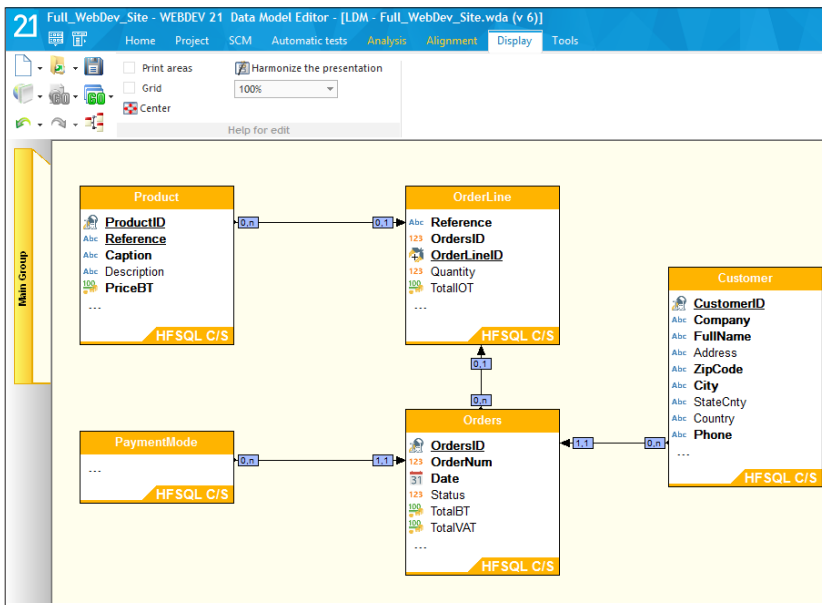


11. Select the data files that will be copied onto the server: in our case, all the data files found in the EXE directory.



12. Go to the next step and validate.

13. The data files of the analysis are automatically changed to HFSQL Client/Server data files and associated with the selected connection.



**14. 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: tips**

- Check the code of your project: in HFSQL Client/Server, *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 the HFSQL Classic mode: all you have to do is specify the path of the directory containing the HFSQL Classic data files.

**15.** You have adapted the development project. 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 10.3. 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 min

## 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 the computers (setup of a 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.

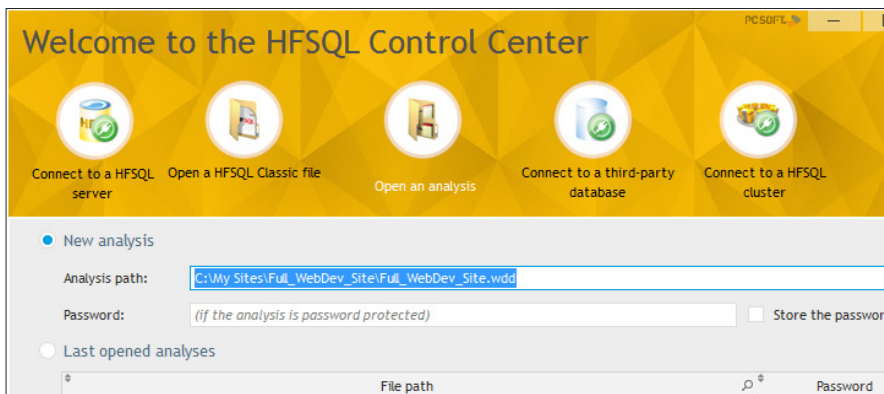
## The HFSQL Control Center

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.

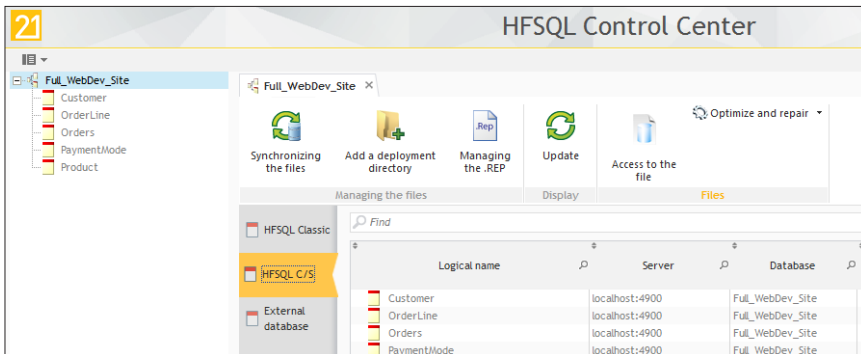
First, we are going to start the HFSQL Control Center from the WEBDEV project.

- 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 of the HFSQL Control Center is displayed. The analysis of the current project is automatically selected.



3. Validate the screen. The HFSQL Control Center is displayed. This start mode allows you to see the different files linked to the analysis of the current project.

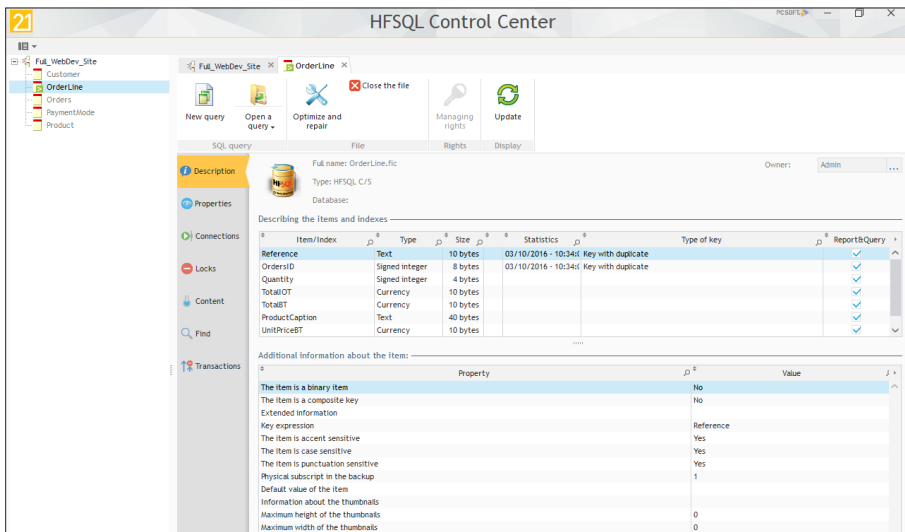
4. Click the vertical tab named "HFSQL C/S" if necessary. The list of data files in HFSQL Client/Server format is displayed.



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 view 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 the 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.

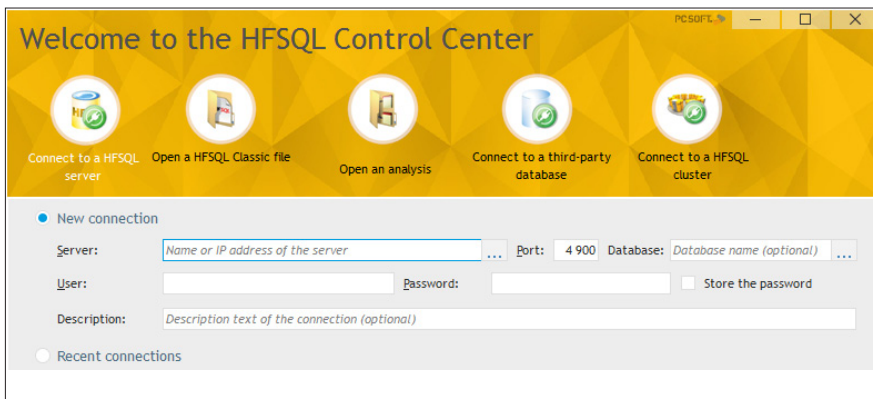
Do not confuse the management of rights for the Client/Server databases with the user groupware that was presented in a previous lesson.

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:

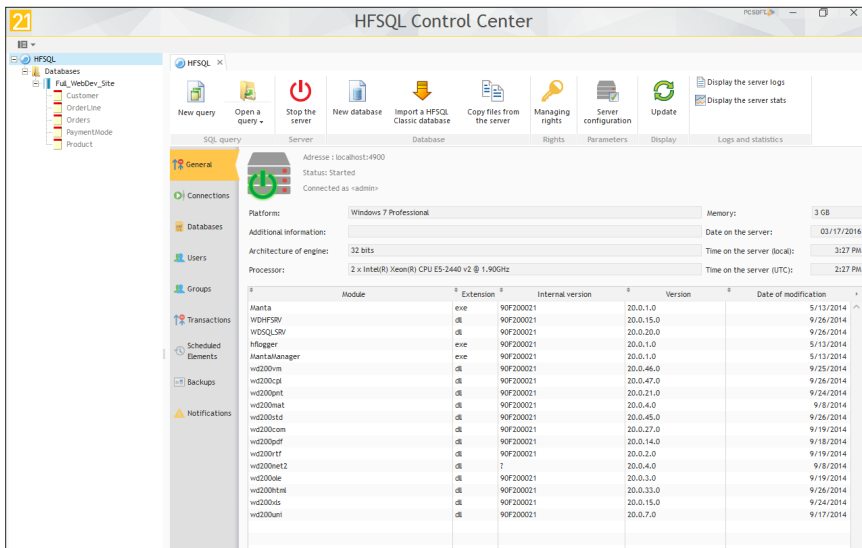
1. Expand the menu  found at the top left of the HFSQL Control Center and select "Connect to a HFSQL server".
2. The home window is displayed in the HFSQL Control Center.



3. The option "Connect to a HFSQL server" is selected by default. Specify the characteristics of the server installed in the previous lesson.

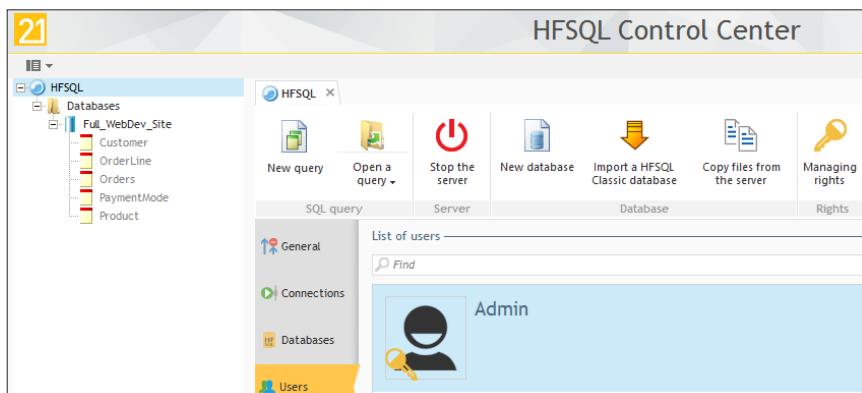
4. The characteristics of the HFSQL server are displayed in the different panels:

- in the panel on the left, the computer, the name of the HFSQL server as well as the list of databases found on this server are displayed.
- in the right section of the screen, a new tab is displayed, allowing you to see the characteristics of the 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. Enter the following information for example:

The screenshot shows a 'Properties of a user' window. The 'User name' field contains 'Test'. The 'Password' and 'Confirm password' fields are masked with dots. The 'Account enabled' checkbox is checked. The 'Full name', 'Email', and 'Phone' fields are empty. The 'Description' field is empty. The 'Photo of the user' field is empty. The 'Groups' section is at the bottom, with 'Access restrictions by IP address' selected. It shows two empty list boxes for 'Server groups' and 'User Groups'. The window has a green checkmark button and a red X button at the bottom right.

(use "Test" as password for example).



Notes

Several characteristics can be noticed:

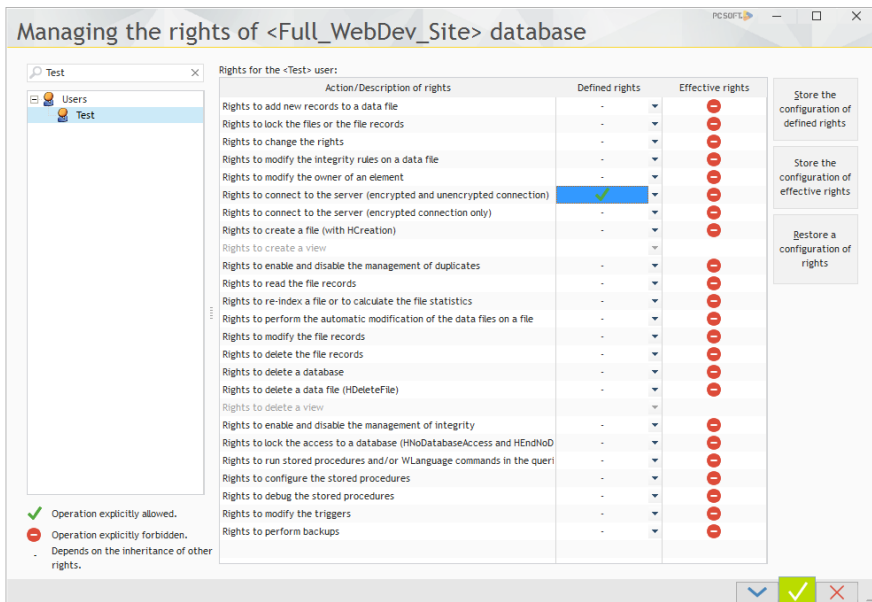
- **Super User:** The users defined as "Super user" are allowed to perform all the actions on the server, on the databases and on all the files.
- **Account enabled:** If this option is not checked, the user exists but he is not enabled (users on holiday for example).
- **Password expiration:** A password can be valid for a given number of days only. (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:

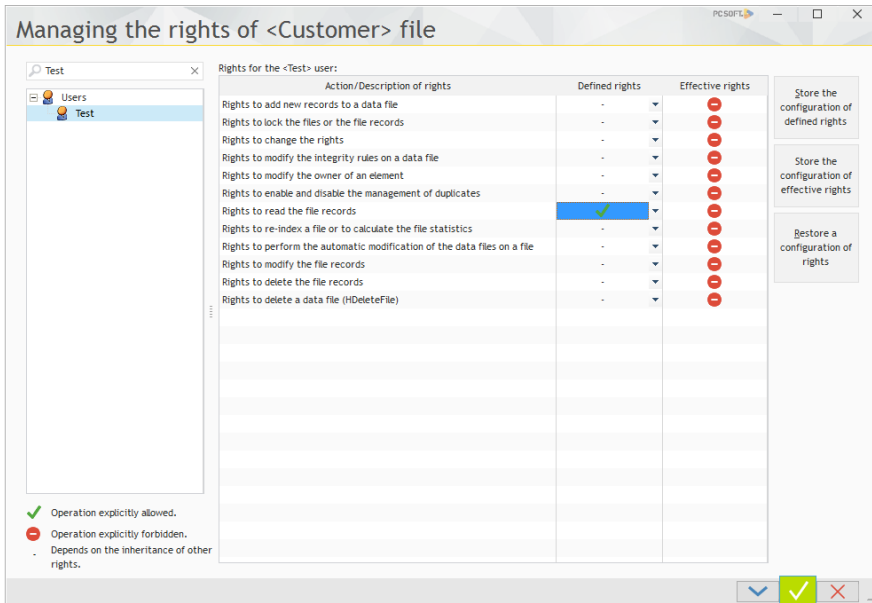
1. In the HFSQL Control Center, double-click the "Full\_WebDev\_Site" database. A new tab appears.
2. On the "Full\_WebDev\_Site" pane, in the "Rights" group, click "Managing rights".
3. Select the "Test" user in the list of users.
4. In the list of rights, for the action named "Rights to connect to the server (encrypted and non-encrypted connection)", click the "Defined rights" column and select the green checkmark.



5. Click the "Apply" button found at the bottom of the window (blue arrow). The rights become effective.

6. Close the window for managing rights.

- To grant the 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 "Customer" pane, in the "Rights" group, click "Managing 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 the window (blue arrow). The rights become effective.
6. Close the window for managing rights.

Similarly, 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, click "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 the 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 administrator of the database.



DEVELOP 10 TIMES FASTER

**PART 11**

**Source Code  
Manager**

21





# LESSON 11.1. SCM

**This lesson will teach you the following concepts...**

---

- Overview
- The Source Code Manager



Estimated time: 15 min

## 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, ...).

WinDev is supplied with a Source Code Manager named "SCM" used to share the source codes of different projects between developers and to find out the full history of the modifications performed (in the code, in the interface, ...).

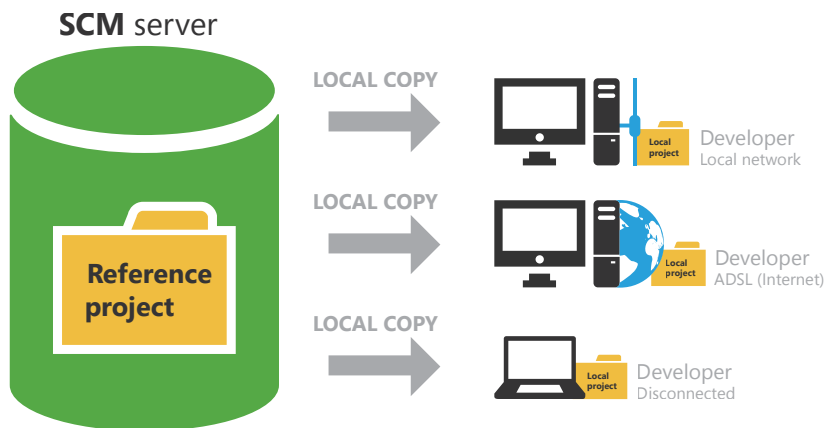
## SCM (Source Code Manager)

### Principle of SCM

The Source Code Manager is used to store and share the projects and their elements.

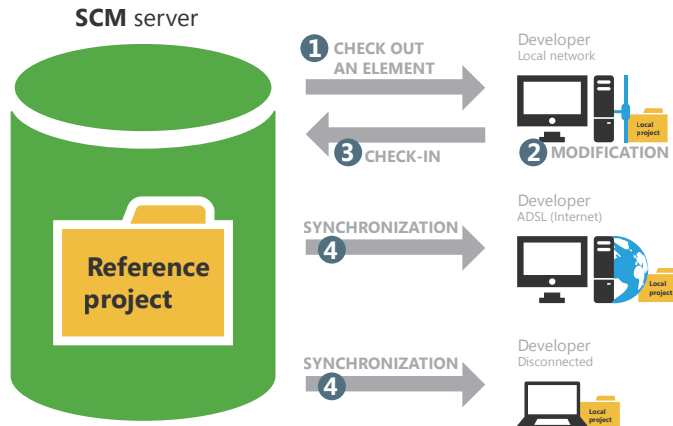
The principle is as follows:

- 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).
- When 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, a 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 the SCM.
- when importing a project into the 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 source codes of projects from anywhere and at any time. See the [www.pcscloud.net](http://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, all you have to do is connect as administrator to the tool for SCM management: on the “Management” pane, in the “Backups” group, click “Full backup”.

## LESSON 11.2. INCLUDING A PROJECT IN THE SCM

**This lesson will teach you the following concepts...**

---

- Adding a project into the SCM.
- Opening a project from the SCM.
- Configuring the SCM.



Estimated time: 15 min



## Including a project in the SCM

### Adding the project into the SCM

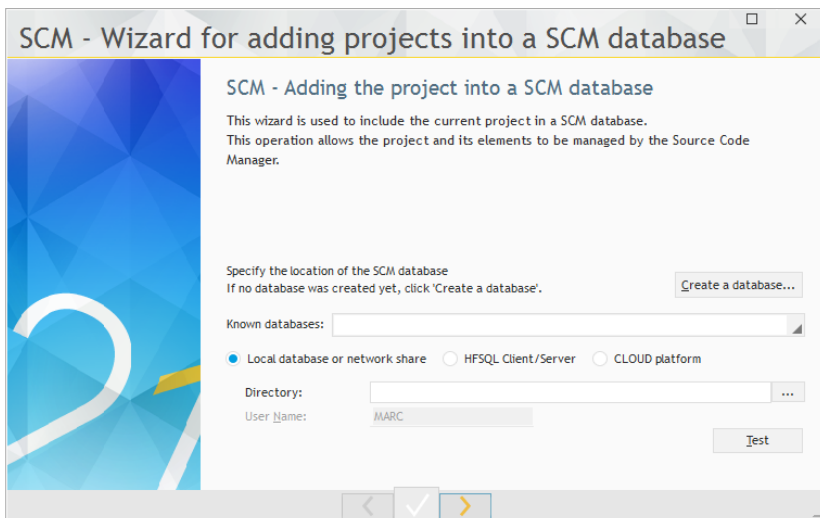
To use an existing project with the Source Code Manager, all you have to do is include this project in the SCM database.



Answers

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 the SCM starts:

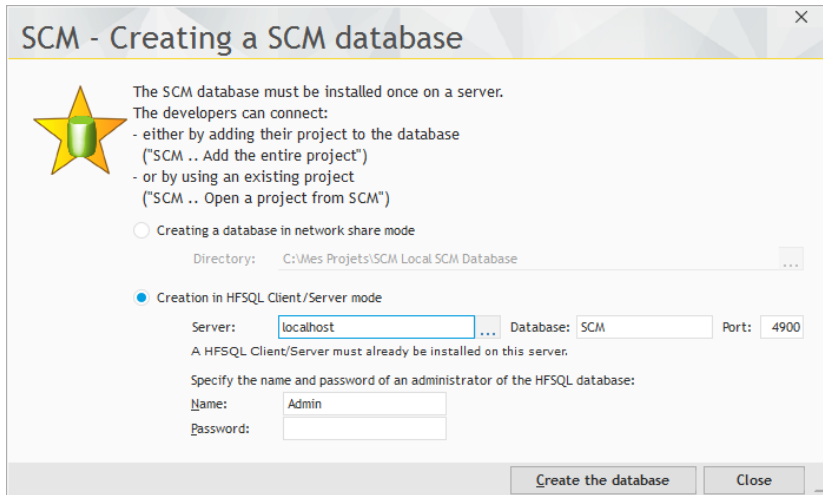


The SCM database was not created yet. We are going to create one.

Note: 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 "Installing a local HFSQL server", page 361.

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.

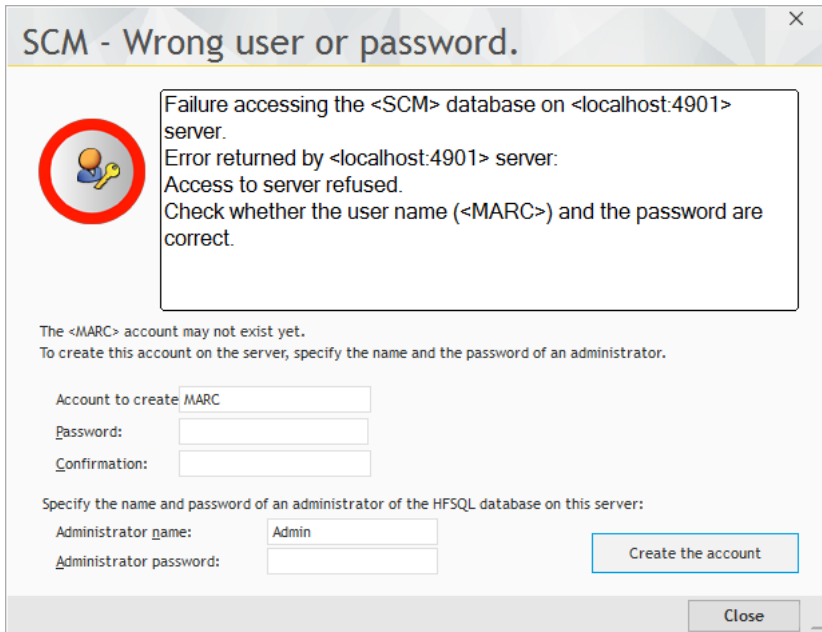


Notes

If the SCM database is in HFSQL Client/Server format, then 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 name of the server, its port.
  - The name of the administrator, his possible password.
6. Validate the creation of the 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.

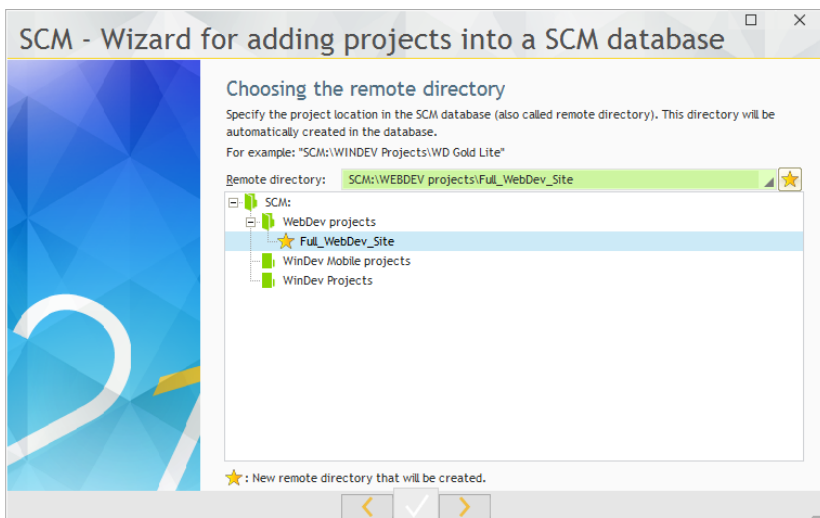
8. Go to the next step. The current user is not saved in the database of users of the HFSQL server.



The dialog box titled "SCM - Wrong user or password." contains a red circle with a key icon. The text inside the dialog reads: "Failure accessing the <SCM> database on <localhost:4901> server. Error returned by <localhost:4901> server: Access to server refused. Check whether the user name (<MARC>) and the password are correct." Below this, it says: "The <MARC> account may not exist yet. To create this account on the server, specify the name and the password of an administrator." There are three input fields: "Account to create" with "MARC" entered, "Password:", and "Confirmation:". Below these, it says: "Specify the name and password of an administrator of the HFSQL database on this server:" with "Administrator name:" containing "Admin" and "Administrator password:". A "Create the account" button is on the right, and a "Close" button is at the bottom right.

We are going to create this user: click the "Create the account" button.

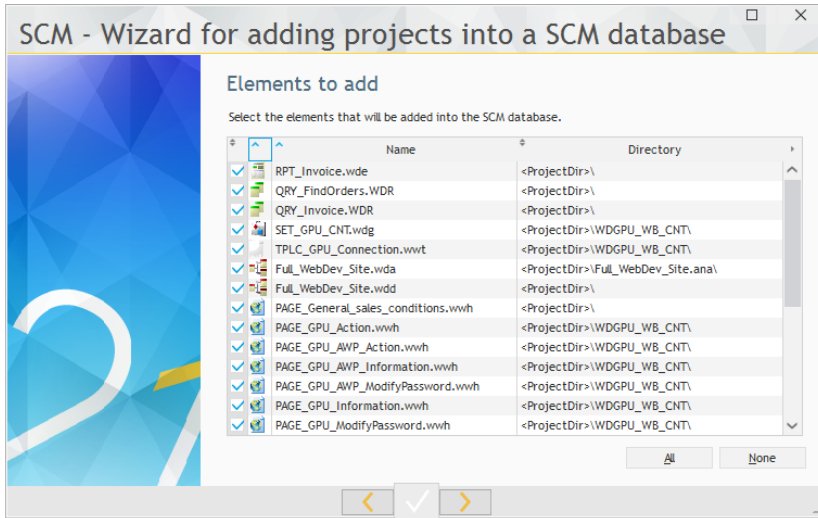
9. The wizard proposes to place the project in the "WebDev projects" sub-directory of the SCM database.



The wizard dialog box titled "SCM - Wizard for adding projects into a SCM database" shows the "Choosing the remote directory" step. It instructs the user to "Specify the project location in the SCM database (also called remote directory). This directory will be automatically created in the database." and gives an example: "For example: 'SCM:\\WINDEV Projects\\WD Gold Lite'". The "Remote directory:" field contains "SCM:\\WEBDEV projects\\Full\_WebDev\_Site". A tree view on the left shows the directory structure: "SCM:" > "WebDev projects" > "Full\_WebDev\_Site" (selected and marked with a star). Other items in the tree are "WinDev Mobile projects" and "WinDev Projects". A legend at the bottom indicates that a star means "New remote directory that will be created." Navigation buttons (back, forward, and a checked "OK" button) are at the bottom.

We are going to accept this location. Go to the next step.

**10.** The wizard asks you to select the project elements that must 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 must be added into the SCM database. These dependencies correspond to all the external elements required by the project (images, style sheets, ...).

We want to add all the project dependencies. Go to the next step.

**12.** Validate the inclusion of the project in the SCM. The project and all its elements have been added into our SCM database.



Notes

### Sharing the 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 applied to the other projects.

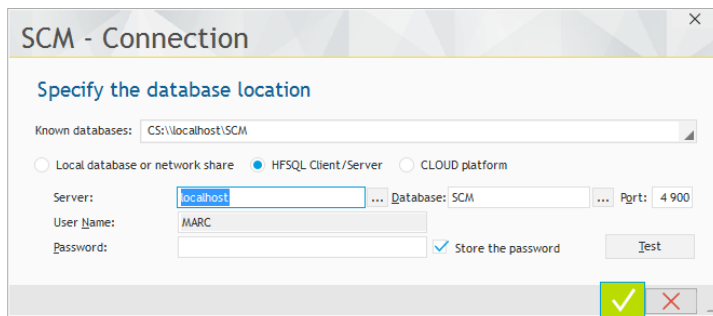
## Opening a project from the SCM

In our example, the project is included in the SCM and it can be used directly.

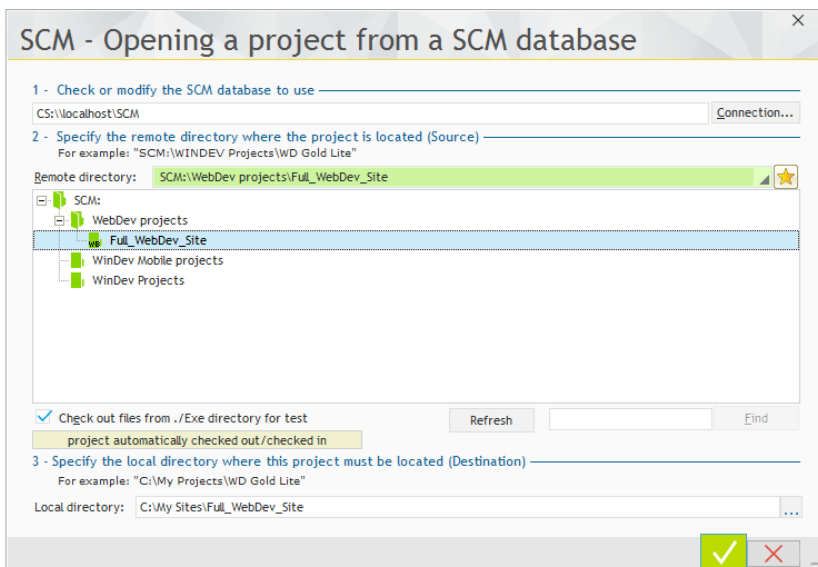
In a real case, in order for other developers to work on a project found in the Source Code Manager, they must retrieve a copy of this project locally.

To do so, the following operations must be performed:

1. Open your 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 the 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 the 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 method for opening a project managed by SCM is identical to the method for opening a project not managed by SCM: all you have to do is open the project (".WWP" file) corresponding to the local copy.

## Configuring the SCM

Before you start working on the elements of the project found in the SCM, you must configure the check-out mode of the project elements.

When working on the elements of a project found in the 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 the SCM users.

**WEBDEV proposes two modes for checking out the project elements:**

- **the standard mode:** if you display a 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 panel).
- **the automatic mode:** if you try to modify a 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 the SCM is used with a slow Internet connection.

**The automatic check-out will be used in this tutorial.**

- To make sure that the automatic check-out is disabled, 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 elements during the first modification".

## LESSON 11.3. 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 min

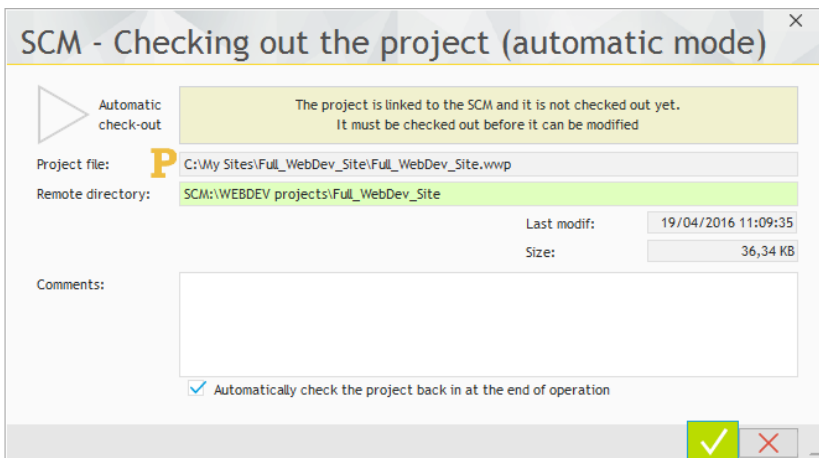
## Overview

We are now going to work with the SCM in real conditions by performing the following operations:

- Modify a project parameter.
- Modify a project page.

## Modifying a project parameter

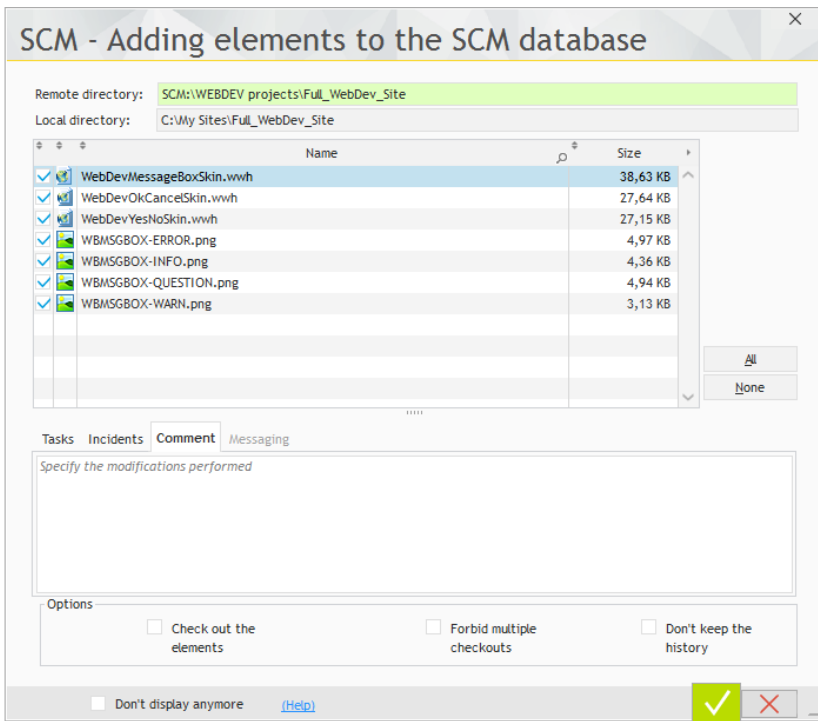
- ▶ We are going to modify the project by asking to display the skin on the dialog boxes:
  1. Display the project description: in the ribbon, on the "Project" pane, in the "Project" group, click "Description".
  2. Click the "Skin/Styles" tab.
  3. Select "Apply the skin to the dialog boxes".
  4. Validate the description window of the 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 so the project must be checked out.



2. The option "Automatically check the project back in at the end of the operation" is used to automatically check the project back in after its modification. Keep this option.
3. Validate this window.



4. A new window appears. This window is used to add elements and their dependencies into the project found in the SCM database. Indeed, the option that was selected adds several pages into the project:



5. Validate this window.

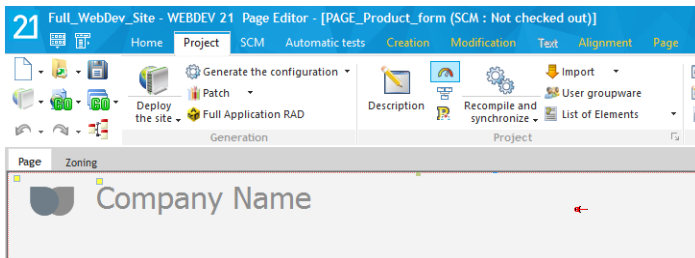
6. The description window of the project is closed: the project is automatically checked back in the SCM database.


## Modifying a project page

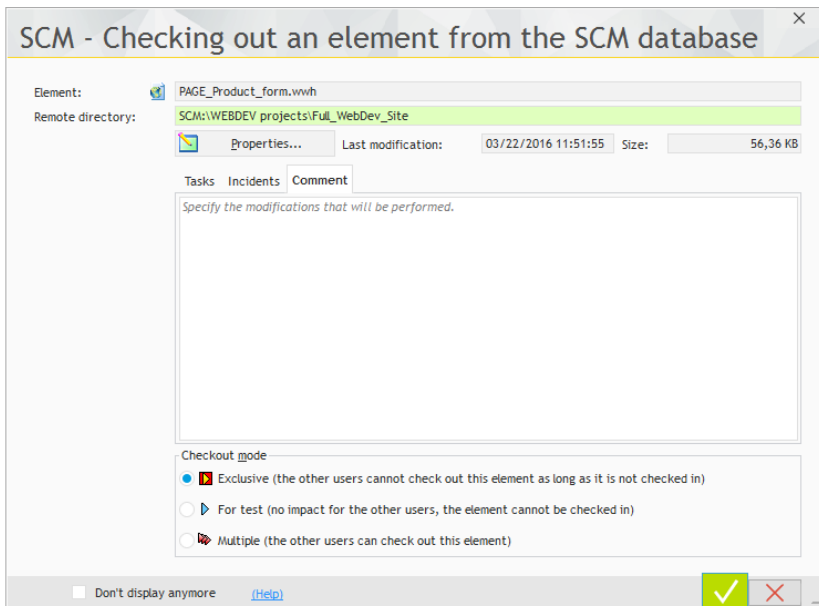
We are now going to modify the "PAGE\_Product\_form" page. We are going to move the "Cancel" and "Validate" buttons.

To modify a project element, this element must be checked out.

- To modify the "PAGE\_Product\_form" page:
  1. Select the "PAGE\_Poduct\_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 the SCM pane (  ).
4. The check-out window is displayed:



5. The SCM proposes three check-out modes:
  - **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 the other users. In this case, the differences between the different versions of the element 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. Enter 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.

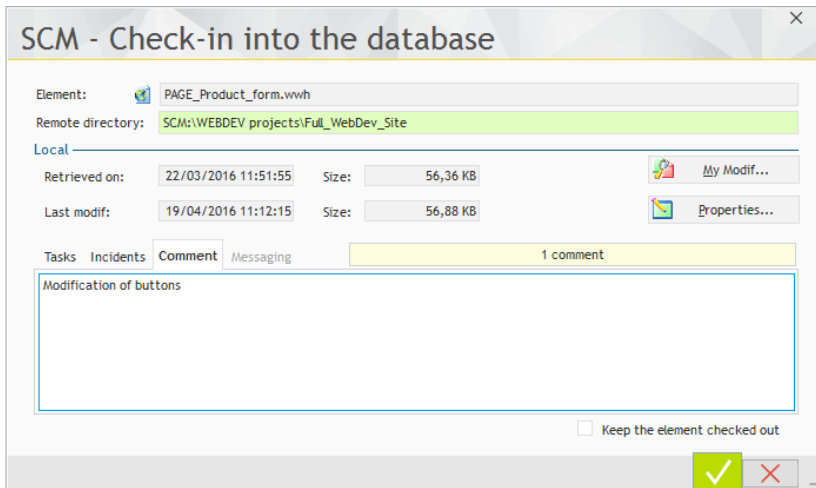
This allows you to make an application evolve by keeping a steady version in the SCM database.

- ▶ 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).
- ▶ Test 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 the 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 code section 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").



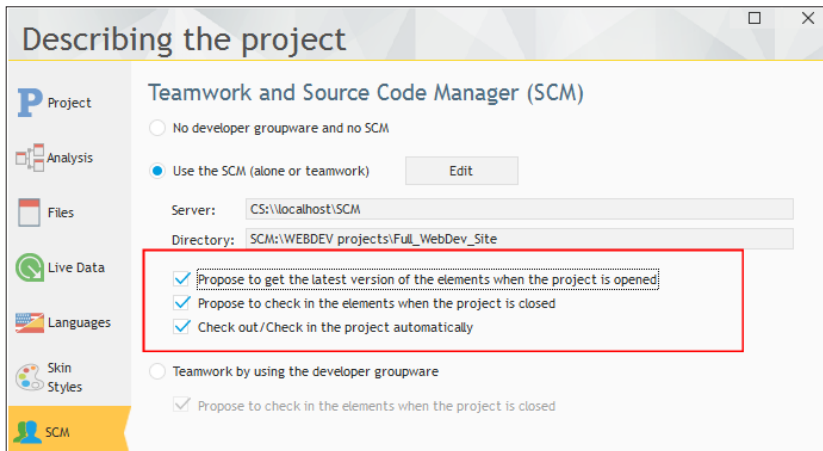
Notes

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 the tasks, the corrections of bugs, ...

4. Validate the check-in.
5. The page is displayed in the editor.

## Synchronizing the project

Several options can be used to configure a project handled by the SCM. These options are grouped in the "SCM" tab of the 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 retrieve the latest version of the project elements.
- *Propose to check in the elements when the project is closed.*  
When closing the project, this option is used to display the list of elements that are currently checked out in order for some of them (or all of them) to be checked back in. By default, the checked-out elements are not checked back in when closing the project.
- *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 the modified elements back in.

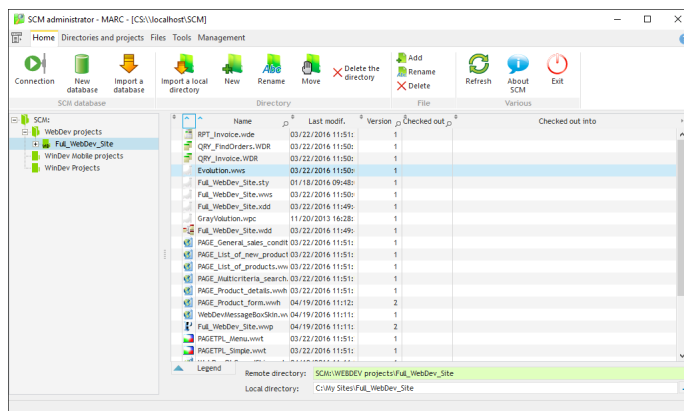
In mobile mode, two solutions are available for checking out elements:

- 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 the 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.

The SCM administrator is used to directly handle the different projects included in the source code manager.

- manage the SCM databases (creation, connection to a SCM database).
- manage the files and directories found in a project of the SCM database (add, delete, rename, ... files and directories).
- manage the different files of 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 "Database" group, click the "Manage" button. All the project elements are listed in the administrator.



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

To stop using the SCM on a project:

1. Display the description window of the 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 the project.





DEVELOP 10 TIMES FASTER

## **PART 12**

# Appendices

# 21





# APPENDICE 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

---

### Native access (connector)

Method for connecting to a database from a program.

### Ajax

AJAX (Asynchronous Javascript and XML) is used to refresh the data modified in an HTML page without having to redisplay the entire page.

### Assignment

Operation that consists in assigning a value in a variable or in a control. For example:

```
// Assign the value MOORE to the CustomerName variable  
CustomerName = "MOORE"
```

The = sign is the assignment operator.

### Alignment

Way to organize the controls in a window or in a page. For example, center a control in a page, define the same width for several controls, ...

### Skin

Style book of a WEBDEV site.

Element in which is defined the graphic layout of the pages found in a WEBDEV project.

### Analysis

Description of the structure of the data files and their relationships.

### Anchoring

Mechanism that consists in defining positioning or resizing rules in order for the content of a window or page to automatically adapt when a window or a browser is resized.

### AWP

Page format directly addressable (fixed URL), referenceable, without automatic context.

### Database

Element containing the data of a program. The data is organized in files or tables.

**Block (report)**

Element constituting a report. For example, a Page Header block, a Page Footer block, a Body block.

**Control (window or page)**

Graphic element used to build the GUI of a program or site.

**Class (OOP)**

Element defined in Object Oriented Programming. A class groups methods (actions) and members (data).

**Classic (file)**

Type of access to a HFSQL file. A HFSQL file is a Classic file when it is directly accessed in its directory.

**Key (file)**

Characteristics of a file item. A key item is used to optimize the searches and the sorts in a data file.

Synonym: index

**Client/Server (file)**

Type of access to a HFSQL file. A HFSQL file is a Client/Server file when it is accessed by connecting to a server containing this file via the HFSQL engine.

**Project code**

Code run when starting a program or a site.

**External component**

Software brick that is used to export one or more business rules in order to re-use them.

**Internal component**

Container that groups elements found in a project (window, page, query, report, class, ...) in order to allow and simplify a share with another project.

**Project configuration**

Description of the output format of a project ; Windows executable, Linux, JAVA, ...

**Context (Page)**

Section stored in memory on a Web server that was used to build a page that is visible in a browser.

**Integrity constraint**

Rule to respect associated with a file item in order to insure the consistency of data in a database.

**CSS**

Language for describing the styles of the different elements found in an HTML page.

**Data binding**

Method that is used to associate a graphic element of the GUI to a data (variable, item).

**Deployment**

Action that consists in installing a program on the user computer.

**Editor**

Program allowing you to create a project element (window editor, page editor, ...)

**Report**

Element of a project that defines a print to perform.

**Event-driven**

Type of programming. A user action on a window or on a page induces a code to run. The code of the action to run is entered in the event representing the user action.

For example, the "Button click" event corresponds to a mouse click performed by the user on this button.

**AAF**

Automatic Application Features

**Style sheet**

Contains the list of styles used in a project.

**File (Table)**

Element that constitutes a database. A file is used to store the data entered in a program.

For example, a CUSTOMER file will contain the customer information that was entered in a program.

**SCM**

Source Code Manager. Tool for organizing, sharing the project sources, managing rights, ...

**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.

**User groupware**

Tool for describing the access rights to the GUI for the users of a program or site. For example, prevent a user from being able to click a "Delete" button according to his login or to his group.

**Homothetic**

Method for resizing an image in order to display it maximized without deforming its content.

**HTML**

HyperText Markup Language.

Language used to describe the elements of a Web page.

**GUI**

Graphical User Interface. Description of the windows or pages that constitute a program.. This is what the user can see when using the program.

**Index (file)**

Synonym: key

**Javascript**

Programming language that is used by the browsers to perform calculations and processes in an HTML page.

**Link (analysis)**

Used to describe the nature of the relationship or the common point between 2 analysis files. From a link, consists in describing the integrity rules that must be respected 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 used only 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.

**Popup menu**

Drop-down menu containing the possible actions depending on the location where the right mouse click was performed and on the type of the element on which this right click was performed.

**Method**

Procedure belonging to a class used to act on the data (members) of the class.

**Control template**

Container of one or more controls (with processes) that can be indefinitely re-used in pages.

Main characteristics of a template: if the initial template is modified, all the modifications are automatically applied to the different uses of the template.

**Page template**

Container representing a standard page that can be applied to a page or to several pages of a project.

Main characteristics of a template: if the initial template is modified, all the modifications are automatically applied to the different uses of the template.

**Report template**

Container representing a standard report that can be applied to a report or to several reports of a project.

Main characteristics of a template: if the initial template is modified, all the modifications are automatically applied to the copies.

**n-tier**

Programming method that is using layers. Each layer is independent and it can be changed without impacting the other ones.

Benefit: Simplified maintenance

Drawback: Difficulty and development time.

**Page**

Element of a WEBDEV project where the graphical interface of the site is defined. In most cases, a site includes several pages linked together.

**Parameter (window, page, procedure, method)**

Element expected in a window, page, procedure or method when it is called. Each value passed in parameter must be assigned to a variable.



**OOP**

Abbreviation of Object-Oriented Programming.

**Popup**

Type of window (or page). A popup is a window (or a page) that is opened on top of a window (or page). You can still view the content of the window (or page) underneath while performing the input in the popup.

**Private**

Variable or procedure that can only be used in the current element.

**Procedure**

Project element containing the code of a process to run.

**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.

**Project**

Element that groups all the elements that constitute a program or a site. A project contains for example an analysis, pages, reports, queries, ...

**Property (control, window, ...)**

Keyword representing a characteristic of an element. The properties are used to handle and modify the characteristics of the project elements by programming.

**Public**

Variable or procedure that can be used from all the elements.

**RAD**

Acronym for Rapid Application Development

Fast development method of a program from an analysis (description of data files).

**Application RAD**

Fast development method of a program from a program template.

**Query**

Element written in SQL language that is used to access (in read-only or in read/write) the content of a database in SQL format.

**Item**

Element that belongs to the structure of a data file or table (of an analysis). For example, a CUSTOMER file can include the FirstName and LastName items.

**Break (report)**

Mechanism that consists in grouping and separating the data according to 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 whenever the city changes.

**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 a database.

**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 type of font, the size of characters, the color of characters, etc.

**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.

**Table (control)**

Graphic element found in a window or in a page. A table control includes one or more columns and rows.

**Array**

Type of variable that contains several values. The values can be accessed via a subscript. The [ ] characters are used from the language to access the array elements.

**Variable**

Element that is used to store a value in memory in a program. Several types of variables exist. Each type corresponds to the nature of the value that we want to store. 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 used to define the splitting of a Web page.

## APPENDICE 2. PROGRAMMING CONCEPTS

**This lesson will teach you the following concepts...**

---

- The different types of variables
- Main statements of WLanguage
- The main operators of WLanguage
- Procedures and functions
- Processing the strings, numeric values and currencies
- Processing the dates and times



Estimated time: 1 hour

## Introduction

---

This lesson allows you to discover the programming concepts of WLanguage by presenting the following topics:

- Declaring the different types of variables,
- Main statements of WLanguage,
- Procedures and functions,
- Processing strings,
- Processing numeric values,
- Processing dates and times.

## Declaring the different types of variables

---

### The different types of variables

WEBDEV proposes several types of variables (boolean, integer, real, currency, string, date, time, duration, datetime, variant, array, structure, ...).

The syntax for declaring a variable is very simple: all you have to do is specify the name of the variable and its type:

```
VariableName IS VariableType
```

Some examples:

```
Subscript is int
SupplierName is string
PriceTable is array of 10 currencies
I, J, K are int
Counter is int = 120
B1 is boolean = False
```

See the online help (keyword: "Data types") for more details

### Declaring the variables and their scope

Two types of variables can be handled in WLanguage:

- The global variables.
- The local variables.

The global variables can be global to a project or to a page. All you have to do is declare these variables:

- in the server code for initializing the project (server variables global to the project). These variables can be used in the server code of all the processes of the project and project elements (pages, reports, ...).
- in the server code for declaring the global variables of a page (server variables global to a page). These variables can be used in the server code of all the processes of the page and page elements.
- in the server code for declaring the sets of procedures (server variables global to a set of procedures).

**Caution !**

The global variables declared in server code are available in the browser code of the page for the following types only: boolean, integer, real, string, simple array or associative array.

The modifications made in browser code to a global variable (assignment of value for example) are not carried over in server code.

- in the browser code for loading the page (browser variables global to a page). These variables can be used in the browser code of all the processes of the page and page elements.

**Caution !**

The browser variables global to a page are not available in the server codes of the page.

All the variables declared elsewhere are local variables.

**Caution !**

**Local variable: specific feature in browser code**

In browser code, only the following types can be used to declare the local variables: boolean, integer, real, string, date, time, datetime, duration, structure, simple array or associative array of one of the previous types.

**Caution !**

Most developers are tempted to declare all their variables as "global variables" in a project. This type of programming is quite "easy". All the variables being global variables, they can be handled from any process.

But this type of programming often causes overwritten variables and exception problems.

Therefore, we advise you to declare a limited number of global variables and to declare local variables when necessary.

## Main statements of WLanguage

WLanguage is a 5GL made of:

- WLanguage functions.
- WLanguage properties.
- WLanguage keywords.
- WLanguage statements.

You will discover all these statements as you progress through this tutorial.

The main statements (used to create conditions and loops or to enter comments) will be presented in this lesson.

- To understand a concept, let's move on to practice. The "WW\_Programming\_Concepts" project contains examples for each presented concept.
1. Start WEBDEV 21 (if not already done).
  2. Close the current project if necessary: on the "Home" pane, in the "General" group, expand "Close" and select "Close the project".
  3. In the home window, click "Tutorial" and select the project named "Programming concepts".
- Tip: on the "Home" pane, in the "Online help" group, you also have the ability to expand "Tutorial" and to select "Programming concepts".

### Conditional statements (server code and browser code)

WLanguage allows you to manage the conditional statements such as:

- **IF, ELSE, END** to run a test on a condition.
- **SWITCH, CASE, END** to perform one or more actions according to the result of a test on a condition.


#### Some examples

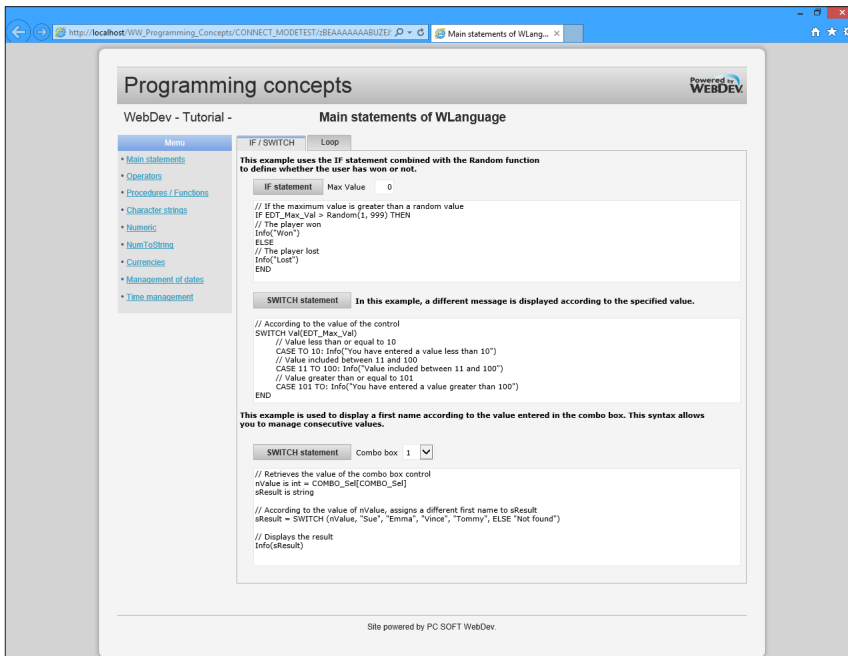
```
IF EDT_Maximum > Random(1, 999) THEN
    Info("Congratulations, you won!")
ELSE
    Info("Bad luck, you've lost!")
END
```

```
SWITCH Day
CASE "Monday"
    // First day of the week
CASE "Wednesday"
    // Third day of the week
CASE "Friday"
    // Fifth day of the week
CASE "Saturday", "Sunday"
    // It's the weekend
OTHER CASE
    // It's another day
END
```

## Practical example

Now that you are familiar with the main conditional statements, let's run a real test!

- ▶ Open the "PAGE\_Statements" page (double-click its name in the project explorer for example). This page presents several examples.
  - ▶ Run the test of this page (  among the quick access buttons).
1. Click the "IF/SWITCH" tab.



2. Test the different examples. The code run is displayed in the page.
3. Close the browser.

## Loop statements (server code and browser code)

WLanguage proposes several methods for managing the loops:

- **FOR, END** for a given number of iterations.
- **WHILE, END** for an undefined number of iterations whose exit condition is checked at the beginning of the loop.
- **LOOP, END** for an undefined number of iterations whose exit condition is checked inside the loop. The **BREAK** statement is used to exit from this loop.



Notes

WLanguage also proposes FOR EACH and END loops used to browse the control elements, the character strings, the records of a data file, ... These advanced loops will be presented later in this tutorial.



## Some examples

```
i is int = 0
WHILE i < Maximum
  i ++ // Equivalent to i=i+1
END
```

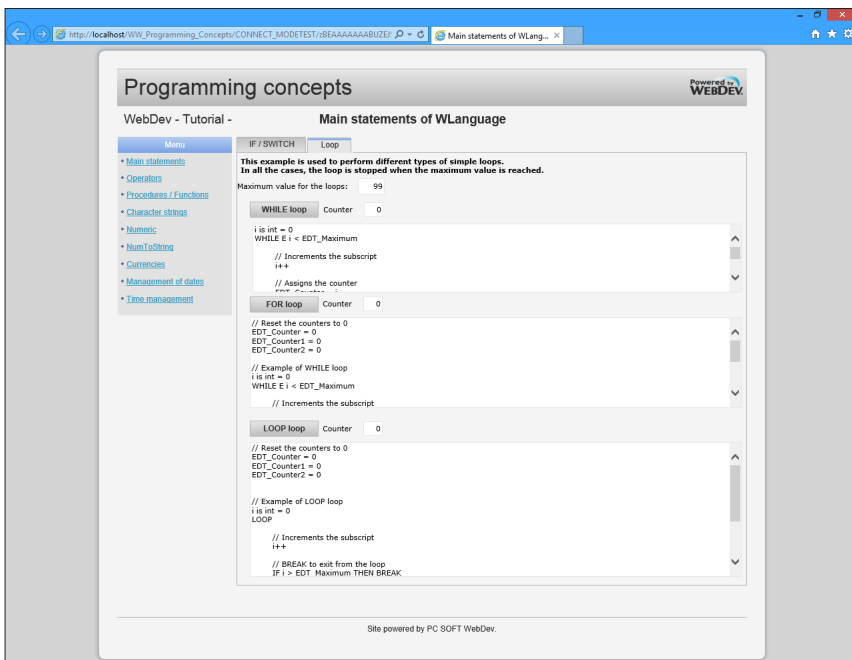
```
i is int = 0
LOOP
  i ++ // Equivalent to i=i+1
  IF i > Maximum THEN BREAK
END
```

```
FOR i = 1 TO 100 // There is no need to declare i
  Cnt ++ // Equivalent to Cnt=Cnt+1
END
```

## Practical example

Now that you are familiar with the main loop statements, let's see the operating mode at run time by checking the "PAGE\_Statements" page again.

- Run the test of the "PAGE\_Statements" page (  among the quick access buttons).
- 1. Click the "Loop" tab.



- 2. Test the different examples. The code run is displayed in the page.
- 3. Close the browser and go back to the editor.

## Comments

To enter comments in the code, the code line must start with `//` (two `/` characters).

Example:

```
// This is a comment line
```



Tip

Several code lines can be set in comment ; to do so, select the code lines with the keyboard (or with the mouse) and press the [CTRL] / keys (on the numeric keypad).

To perform the reverse operation (to remove the comments), select the code lines with the keyboard (or with the mouse) and press the [CTRL] [SHIFT] / keys (on the numeric keypad).

## Main WLanguage operators

WLanguage includes several types of operators:

- the logical operators (AND, OR, NOT, ...),
- the arithmetic operators that are used to add, subtract, ...
- the comparison operators that are used to compare values. Some of them have been used in the previous paragraph (`>`, `<`, `=`, ...),
- the binary operators that are used to perform operations on the binary values,
- the assignment and exchange operators (`=`, `<=>`),
- the indirection operators that are used to build the name of a control from an expression.
- the operators on the character strings (that will be presented in “Processing strings (server code and browser code)”, page 422).
- ...

We are going to present the use of the logical operators, comparison operators and indirection operators. The other types of operators will be presented when necessary in this tutorial. See the online help (keyword: "Operators") for more details.


### The logical operators

WLanguage includes 5 logical operators that are used to perform logical operations and to build conditions:

- AND and `_AND_` that correspond to the logical multiplication.
- OR and `_OR_` that correspond to the logical addition.
- NOT that corresponds to the logical negation.

### Practical example

Let's see an example to understand the subtlety of these operators.

- ▶ In the "WW\_Programming\_Concepts" project, open the "PAGE\_Operator" page.
- ▶ Run the test of this page (  among the quick access buttons) and test the logical operators.


### The comparison operators

WLanguage includes several comparison operators that are used to perform several processes:

- equality operators: equality (=), flexible equality (~=) or very flexible equality (~~).
- comparison operators: different (<>), less and greater than (<, <=, >, >=) or starts with ([=, [=~, [=~~).

### Practical example

An example page is available for checking the different use cases of the comparison operators.

- ▶ In the "WW\_Programming\_Concepts" project, open the "PAGE\_Operator" page.
- ▶ Run the test of this page (  among the quick access buttons) and test the comparison operators.

### The indirection operators

The "indirection" mechanism is used to build the name of a control, the name of a file item or the name of a variable from a string expression.

This allows you to create generic processes independent of the names of controls, variables, file items...



Notes

The indirection is performed by the operators { }.

To optimize the speed of your applications, the type of the element handled should be specified when using the indirection syntax.

Therefore, if you are using a control, the corresponding type will be *IndControl*.

Some indirection examples:

- Example of simple indirections:

```

{"NAME",indControl} = CustName
// is equivalent to NAME=CustName
{"NAME",indControl} = {"CU.CUSTNAME"}
// is equivalent to NAME=CU.CUSTNAME
{"PAGE_CUST.NAME",indControl} = CustName
// is equivalent to PAGE_CUST.NAME=CustName
{"PAGE_CUST"+"NAME",indControl} = CustName
// is equivalent to PAGE_CUST.NAME=CustName

```

- Example of indirection with a procedure:

```
ControlName is string
ControlName = "EDIT1"           // EDIT1 is the name of the
control
// Call to a procedure used to make a control invisible
INVISIBLE(ControlName)

PROCEDURE INVISIBLE(ControlName)
{ControlName,indControl}..Visible = False
```

**Example**

See the unit example named "Using the indirections", supplied with WEBDEV, for more details. This example can be accessed from the home window of WEBDEV (CTRL <).

## Procedures and functions

### Definition

When a process is called several times in a project or in a page, it may be interesting to create a procedure containing this process. Then, all you have to do is call the procedure whenever required.

The procedures and the functions are available in programming:

- The **functions** return a result.
- The **procedures** are used to run a specific process.

WEBDEV allows you easily manage the two types of processes: in WEBDEV, there is no difference between a procedure and a function. This is the reason why we shall talk about "procedure" in the rest of this lesson.

You have the ability to create "local" procedures and "global" procedures.

**Caution !**

The procedures in server code are not accessible in browser code and the procedures in browser code are not accessible in server code. However, a server procedure can be run from a browser code via the validation of a button or via **AJAXExecute**.

### Local procedure

A "local" procedure is linked to a single page.

A local procedure can only be used in the processes of the page where it was declared (and in the processes of the controls found in this page). It is part of the page.

## Global procedure and set of procedures

The "global" procedures are stored in "sets of procedures". Each "set of procedures" is a file that contains all the global procedures that are associated with it.

For example, a set is used to group the procedures according to a topic: CalcVAT, CalcShippingCost,

...

In a project corresponding to a dynamic site, two types of sets of procedures can be used:

- A set of server procedures, file whose extension is ".WDG".
- A set of browser procedures, file whose extension is ".WWN".

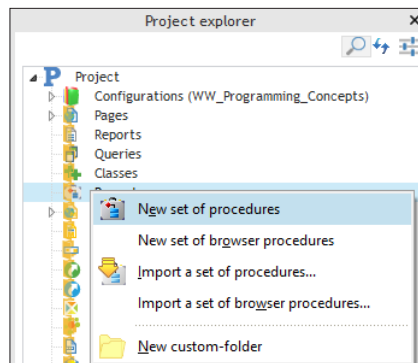
You can create as many sets of procedures as necessary in a project.

A set of procedures is used to:

- share the global procedures between several developers, for the same project.
- share the global procedures between several projects. Indeed, the same set of procedures can be used in several projects.

► To create a set of procedures:

1. Select the "Project explorer" pane.
2. Select "Procedures".
3. In the popup menu (right mouse click), select:
  - "New set of procedures" to create a set of server procedures.
  - "New set of browser procedures" to create a set of browser procedures.



4. In the window that is displayed, specify the name of the set of server or browser procedures and validate.
5. The processes associated with the set of procedures are displayed in the code editor.
6. Save the set of procedures (CTRL S).
7. The new set of procedures is displayed in the project explorer.

Then, you will have the ability to create global procedures (this creation is presented in details in the rest of this lesson).

## How to decide whether a procedure is global or local?

To choose whether a procedure must be global or local, ask yourself the following question: "Will the procedure be used in this page only or can it be called from another page?".

- If the procedure is called "from this page only", the procedure can be "local".
- If the procedure can be called "from several pages", the procedure must be "global".

## About passing parameters

A procedure can manage parameters. The parameters can be mandatory or optional parameters. The "mandatory" parameters are always defined before the "optional" parameters. The declaration of an "optional" parameter is performed by assigning a default value when declaring the parameter. For example:

```
PROCEDURE MyProcedure(Param1, OptionParam = "Default value")
```

## Calling a procedure

To call a procedure in a process, all you have to do is write its name in the code editor and specify parameters if necessary.

WEBDEV manages the calls to nested procedures.



TIP

In the code editor, when you are positioned on a procedure name, press the [F2] key if you want to view the code of this procedure.

Press [CTRL]+[F2] to go back to the name of the procedure in the previous code.

## Creating a procedure

A procedure can be created from the main editor of WEBDEV. All you have to do is use the "Project explorer" pane.

1. To create a local procedure: In the "Project explorer" pane, select the name of the page associated with the local procedure.
2. Click the arrow on the left to display the different elements.
3. Select "Local procedures" (or press the [F8] key).
4. In the popup menu of "Local procedures", select:
  - "New local procedure" to create a local server procedure.
  - "New local browser procedure" to create a local browser procedure.
5. In the window that is displayed, specify the name of the procedure and click "Add".
6. The procedure appears in the code editor. Enter the code of the procedure.

- ▶ To create a global procedure:
  1. Select the "Project explorer" pane.
  2. Select "Procedures".
  3. Create (if necessary) the set of procedures where the global procedure must be created. If this set exists, select it.
  4. Select "New global procedure" from the popup menu.
  5. In the window that is displayed, specify the name of the procedure and validate.
  6. The procedure appears in the code editor. Enter the code of the procedure.

**Reminder:** A procedure can also be created from the code selected in the code editor ("New .. Create a global procedure containing the selected code" from the popup menu).




Tip

#### When should the procedures be used?

- When a process is used several times in the same page, we recommend that you use a procedure local to the page that will contain this process.
- When a process is used several times in a set of pages, we recommend that you use a procedure global to the project that will contain this process.

## Practical example

Enough theory, let's get down to work!

- ▶ In the "WW\_Programming\_Concepts" project, open the "PAGE\_Code" page (double-click its name in the project explorer). This page presents several examples.
- ▶ Run the test of this page (  among the quick access buttons).
  - At the top, you can test the use of a procedure.
  - At the bottom, you can test the use of a function.
- ▶ Close the browser.
- ▶ Display the list of local procedures (accessible from the "Project explorer" pane). When you double-click the name of the procedure, the code of this procedure is displayed in the code editor.

## A practical exercise? It's your turn!

A small exercise will be used to implement the different features found in this section:

- ▶ In the "PAGE\_Code" page:
  - Create a procedure used to display all the even numbers from 10 to 1 (reverse order) in the EDT\_Code edit control.
  - This procedure must be called from the "It's your turn" button.
  - Run the page test to run the test of your code.

Correction: The "Correction" button contains the solution.


## Processing strings (server code and browser code)

The ability to handle character strings is one of the most important features of a programming language.

Several WLanguage features can be used to handle the character strings: WLanguage functions, extraction operators, concatenation operators, ...

The most common functions for handling the character strings will be presented in this tutorial. See the online help (keyword: "Character string") for more details.

### Practical example

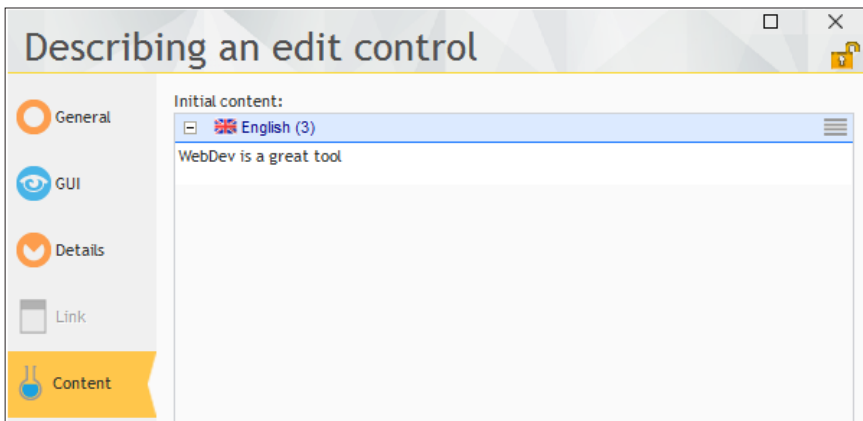
- ▶ In the "WW\_Programming\_Concepts" project, open the "PAGE\_String" page. This page presents the different operations that will be explained in this lesson.
- ▶ Run the page test (  among the quick access buttons).

### Main operations performed on a character string

#### Initialization

A text control (a Static control or an edit control for example) can be initialized by one of the following methods:

- in the "Content" tab of the description window of the control:



- by programming, by assigning the string to the name of the control:

```
EDT_Text = "WebDev is a great tool"
```

- by programming, by using a string variable:

```
str is string  
str = "WebDev is a great tool"  
EDT_Text = str
```



## Concatenation

A string can be built from several other strings. We talk of **string concatenation**. To concatenate two strings, all you have to do is use the "+" operator.

```
// Info is used to display the result on the screen
Info(EDT_Text1 + EDT_Text2)
```

## Extracting sections from a string

A section can be extracted from a string by:

- the [[ and ]] operators (caution: no space must be found between the [[ and ]] brackets).

```
Info(EDT_Text[[1 TO 6]]) // Displays "WebDev"
```

- **ExtractString**, that extracts a sub-string from a string:

```
Info(ExtractString(EDT_Text,1," ")) // Displays "WebDev"
```

- **Middle**, that extracts a section of string from a string:

```
Info(Middle(EDT_Text,4,2)) // Displays "De"
```

- **Left**, that returns the left section of a string:

```
Info(Left(EDT_Text,6)) // Displays "WebDev"
```

- **Right**, that returns the right section of a string:

```
Info(Right(EDT_Text,3)) // Displays "ool"
```

## Various operations

The size of a string is returned by **Length**:

```
Info(Length(EDT_Text)) // Display 22
```

A string can be changed into uppercase characters with **Upper** or to lowercase characters with **Lower**:

```
Info(Upper(EDT_Text))
Info(Lower(EDT_Text))
```

A string can be sought in another string by **Position**:

```
StringToFind is string = "WebDev"
Pos is int
Pos = Position(EDT_Text, StringToFind)
IF Pos = 0 THEN
    InfoBuild ("%1 not found in the text", StringToFind)
ELSE
    InfoBuild ("%1 found in the text", StringToFind)
END
```

You can also find the position of a character string inside another one while ignoring the case. To do so, **Position** must be associated with the **IgnoreCase** constant:

```
Pos = Position(EDT_Text, StringToFind, 1, IgnoreCase)
```

To find out the number of occurrences of a given character string inside another character string, use **StringCount**:

```
NbOccurrences is int
NbOccurrences = StringCount("anastasia", "a") // Returns 4
```

You also have the ability to find a string and to replace it in a single operation via **Replace**:

```
Replace(EDT_Text, "DeamDream", "WebDev")
```



Tip

To build a string from a string and from the result of a function, we recommend that you use **InfoBuild**. This function is used to build the text displayed according to parameters (%1, %2, ...):

```
InfoBuild("%1 not found in the text", ...
StringToFind)
```

You also have the ability to use the simple concatenation:

```
Info(StringToFind + " not found in the text")
```

Using **InfoBuild** presents the following benefits:

- the readability of the code: the string is easier to read.
- the translation: a string can be entirely translated. The words and the parameters can be reversed. The translator has a single obligation: include %1, %2, %3.

According to the same principle, WLanguage also proposes **StringBuild**, **ErrorBuild**, ...

## A practical exercise? It's your turn!

A small exercise will be used to implement the different features found in this section:

- In the "PAGE\_String" page, an exercise is proposed in the "It's your turn" area:
  - In the "Your code" button, write the code used to display the 5th word of the text in uppercase characters as well as the size of this word.
  - Run the page test to run the test of your code.

Correction: in the "Corrected" section, the "Your code" button contains the solution.

## Processing numeric values (server code and browser code)

The calculations on numeric values can be performed from numeric edit controls or by using the typed variables (integer, real, numeric, currency, ...).

### Practical example

- ▶ In the "WW\_Programming\_Concepts" project, open the "PAGE\_Numeric" page.
- ▶ Run the page test. This page presents an overview of the operations on "numeric" values.

### Main operations performed on the numeric values

#### Initialization

A numeric edit control can be initialized by one of the following methods:

- the "Content" tab in the description window of the control.
- by programming, by assigning the numeric value to the name of the control:

```
EDT_InitialValue = 3.14
```

- by programming, by using a numeric variable:

```
int1 is int  
int1 = 1234  
EDT_InitialValue = int1
```

The "+" operator is used to concatenate a string and a numeric value:

```
Info("The EDT_InitialValue control contains the value: " + ...  
    EDT_InitialValue)
```



Tip

If you run the following code:

```
Info("A calculation: " + 1 + 2)
```

The dialog box will display "A calculation: 12".

To display the result of the calculation, you must use the following code:

```
Info("A calculation: " + (1 + 2))
```

The dialog box will display "A calculation: 3".

## Various operations

Some examples of numeric operations performed in WLanguage:

- The integer part of a number is returned by **IntegerPart** and the decimal part of a number is returned by **DecimalPart**:

```
InfoBuild("Integer part of %1: %2" + CR + ...
  "Decimal part of %1: %3", EDT_SignedReal, ...
  IntegerPart(EDT_SignedReal), ...
  DecimalPart(EDT_SignedReal))
```

- The absolute value of a number is returned by **Abs**:

```
InfoBuild("Absolute value of %1: %2", ...
  EDT_SignedReal, Abs(EDT_SignedReal))
```

- The rounded value of a number is returned by **Round**:

```
InfoBuild("Rounded value of %1: %2", ...
  EDT_SignedReal, Round(EDT_SignedReal, 1))
```

- **Root** is used to calculate the Nth root of a number:

```
InfoBuild("Square root of %1: %2", ...
  EDT_Reference_Root, Root(EDT_Reference_Root, 2))
```

- **Power** is used to raise a number to the power of N:

```
InfoBuild("Power 2 of %1: %2", ...
  EDT_Reference_Power, ...
  Power(EDT_Reference_Power, 2))
```

### Note: the type of the numeric edit controls

When a control is defined as a numeric control, its type may be undefined (integer, real, double real, ...). Its type is automatically defined according to the mask selected for the control.

To force the type of a control, all you have to do is use a typed variable. For example:

```
Val1 is int
Val1 = 123456789
EDT_Num1 = Val1      // Assign the control
Val1 = EDT_Num1      // Retrieve the control
```

## A practical exercise? It's your turn!

A small exercise will be used to implement the different features found in this section:

- ▶ In the "PAGE\_Numeric" page, in the "It's your turn" area, an exercise is proposed:
  - In the "Calculate" button, write the code used to display (in the "Result" control) the result of the subtraction performed between the "Amount 1" and "Amount 2" controls, rounded to 1 decimal.
  - Run the page test to run the test of your code.

Correction: In the "Corrected" section, the "Calculate" button contains the solution.

## Processing currencies (server code only)

### Practical example

- ▶ In the "WW\_Programming\_Concepts" project, open the "PAGE\_RealCurrency" page.
- ▶ Run the page test. This page is used to run the test of the different code sections presented in this paragraph.

### Details

The reals are used to handle the decimal numbers by programming.

The currencies can also be used to manage these decimal numbers but with a better precision.

When using reals, the rounding is performed by the system and these rounding operations may induce calculation errors.

For example:

```
// The calculation is false with a Double Real variable
x is real
x=18.6 - 8.6 - 10
Error ("18.6-8.6-10="+x)
```

- ▶ To run the test of this code, click the "Operations with reals" button in the test page.

With currency variables, the calculation is performed by WLanguage.

WLanguage guarantees 23 significant digits which allows for more precise calculations.

The calculation is correct with the following code:

```
x is currency
x=18.6 - 8.6 - 10
Info ("18.6-8.6-10="+x)
```

- ▶ To run the test of this code, click the "Operations with currencies" button in the test page.

To perform divisions on currencies, we recommend that you use intermediate variables of "Currency" type.

```
Currency1, Currency2 are currencies
// Assigns a number to the two currency variables
Currency1 = "12 345 678 901 234 567,123456"
Currency2 = "12 345 678 901 234 567,123456"
// Divide Currency1 by Currency2
Currency1 = Currency1/Currency2
// Displays the result of the division
Info("12 345 678 901 234 567,123456"," divided by ",...
    "12 345 678 901 234 567,123456 " , " = " + Currency1)
```

To run the test of this code, click the "Division of currencies" button in the test page.



Note

To perform advanced calculations, WLanguage proposes the **Numeric** type. By default, the Numeric type corresponds to 32 digits for the integer part and to 6 digits for the decimal part (like the currency type). The numeric type allows you to configure the number of digits that will be used for the integer part and the number of digits that will be used for the decimal part.

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

## Mixing strings and numeric values

The numeric values and the strings can be mixed together. WEBDEV is very flexible for assigning variables. For example, a string of digits can be assigned in a numeric variable (and conversely). For example:

```
i is int
c is string
i = 123
c = i // the variable c contains the string "123"
c = "456"
i = c // the variable i contains the value 456
```

To transform a number into a character string while respecting a specific format, all you have to do is use **NumToString**. For example:

```
NumToString(1234.567, "012,3f") // returns "00001234,567"
```

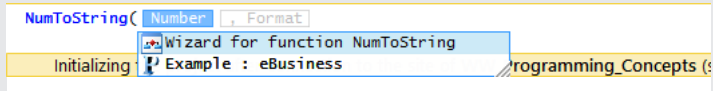
- In the "WW\_Programming\_Concepts" project, open the "PAGE\_NumericToString" page.
- Run the page test. This page is used to test the different parameters of **NumToString**.

Whenever you must call **NumToString** without exactly knowing which parameters to use, use the code wizard proposed by WEBDEV: this wizard will give you the syntax to use according to the requested result.



Notes

In the code editor, various information can be displayed when typing the name of a function followed by the opening bracket:



- the name of the wizard corresponding to the function. A wizard starts if this option is selected. Via simple questions, this wizard automatically writes the requested syntax.
- the name of an example that is using this function. The example is automatically opened in the editor if this option is selected.

The "reverse" function of **NumToString** is **Val**. This function is used to convert a string into a numeric value.

## Processing dates and times (server code and browser code)

To easily manage the dates and times in your applications, WEBDEV includes:

- a **Date, Time or Duration edit control**. With this control, no more hassle to enter a valid date or a valid time.
- a **Date, Time or Duration static control**. With this control, no more hassle to display a date or a time in the proper format.
- **Date, Time, DateTime and Duration variables**. These variables allow you to easily handle the dates and times by programming and to perform multiple calculations.

## The dates (server code and browser code)

### Practical example

- In the "WW\_Programming\_Concepts" project, open the "PAGE\_Date" page. This page illustrates the explanations given in the rest of this lesson.

### Input mask and returned value

The dates (like the times) are strings with a preset format.

In a "Date" (or "Time") edit control, you must distinguish between:

- The **edit mask**: it is the format used to enter the date and the time.
- The **returned value** (or stored value): it is the value returned by the control to the program.

This information is entered in the description window of the edit control.

- In the "PAGE\_Date" page, display the description window of the "Enter a date" control ("Description" from the popup menu of the control). Display the "General" tab. The following information is displayed:

The screenshot shows the 'General' tab of a control's description window. On the left is a sidebar with three icons: a pencil for 'Note, Help', a yellow arrow for 'Advanced', and a globe for 'Style'. The main panel contains the following settings:

- Type:** A dropdown menu showing 'Date' with a calendar icon.
- Input mask:** A text box containing 'Date defined by the project' with a small 'P' icon and a help icon to its right. Below it is the text 'Select or enter a mask'.
- Returned value:** A dropdown menu showing 'YYYYMMDD'.
- Display a calendar button:** An unchecked checkbox.

For example, for a "Date" control:

- the input mask will be "DD/MM/YYYY". The date entered by the user will have the following format: "01/11/2013".
- the returned value will be "YYYYMMDD" (for example: the value entered in "23/04/2013" format will return "20130423" to the program).



Note

You have the ability to choose a "System date" input mask. In this case, the format displayed will be the one defined in the "Regional settings of the control panel of Windows" of the computer currently run.

**Caution:** The parameters operating system are the ones of the server. If your site is in English but hosted on a French server, the dates will be proposed in "DD/MM/YYYY" format.

Let's take a look at the following examples to understand the relationships between the input mask and the returned value:

- If the input mask of a date control is "DD/MM/YY" and the returned value is "YYYYMMDD":

```
DateD = "20131225" // Displays the date in "25/12/13" format
DateD = "131225"  // Displays the date in an incorrect format
```

The date displayed will be incorrect.

- If the input mask of a date control is "MM/DD/YYYY" and if the returned value is "YYMMDD":

```
DateD = "131225" // Displays the date in "12/25/2013" format
DateD = "20131225" // Displays the date in an incorrect format
```

The date displayed is incorrect.

**Therefore, pay great attention to the format of the returned value in a date or time control.**

The WLanguage functions that handle:


- the dates use the "YYYYMMDD" format.
- the times use the "HHMMSSCC" format (some of them use the "HHMMSSCCC" format to manage the milliseconds).



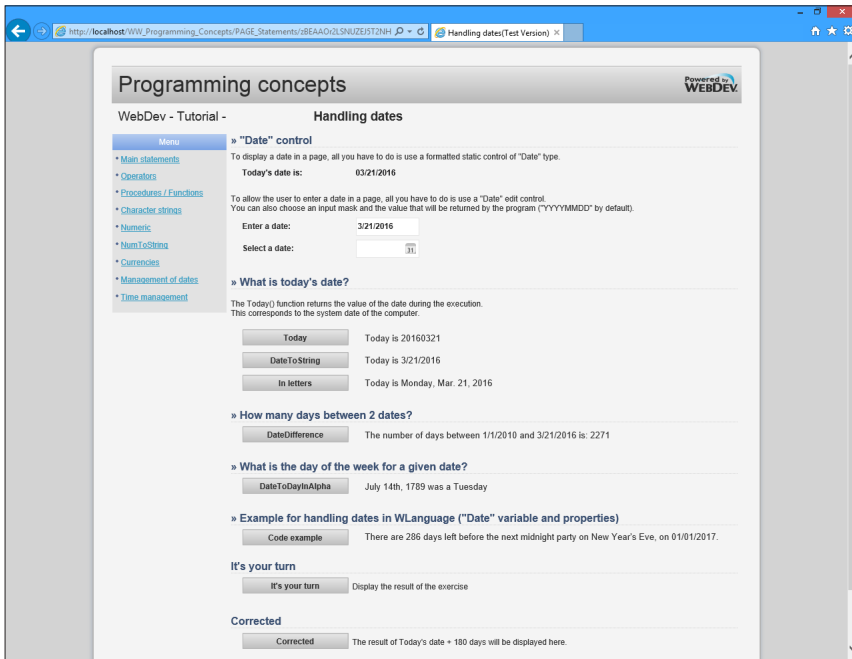
Notes

**Reminder:** By default, the format (input mask) of the control corresponds to the date mask defined in the "Languages" tab of the description window of the project (on the "Project" pane, in the "Project" group, click "Description"). Therefore, the same mask is automatically used in all the date controls of the project.

This feature is also very useful in the multilingual sites.

- Let's now see how the dates can be managed by programming.
  1. Close (if necessary) the description window of the control.
  2. Run the test of the "PAGE\_Date" page (  among the quick access buttons).





## What is today's date?

To find out today's date, all you have to do is use **Today** (or **DateSys**). **Today** returns the system date of your computer as a character string in "YYYYMMDD" format. Example:

```
Info("Today is " + Today())
```

- To test this feature in our example, click the "Today" button.  
You will notice that the date is displayed in "YYYYMMDD" format.

To display the date in a more explicit format, all you have to do is use **DateToString**:

```
Info("Today is " + DateToString(Today(), ...  
maskSystemDate))
```

**DateToString** transforms a string in "YYYYMMDD" format into a string in the selected format. The **maskSystemDate** constant allows you to use the date format defined in the "Languages" tab of the project characteristics.

Reminder: To display the project characteristics, on the "Project" pane, in the "Project" group, click "Description".

- To test this feature in our example, click the "DateToString" button.

## What is today's date, with the day and the month in letters?

To display the date in letters, you must use **DateToString** with a specific date format:

```
Info("Today is " + DateToString(Today(), "Dddd, Mmm DD, YYYY"))
```

In this code:

- the "Dddd" string is used to get the day in letters. For example: Tuesday.
- the "DD" string is used to get the day in number. For example: 25.
- the "Mmm" string is used to get the month in letters. For example: September.
- the "YYYY" string is used to get the year. For example : 2013.

► To test this feature in our example, click the "In letters" button.

## How many days between 2 dates?

You want to find out how many days have passed between two dates? It's quite simple: all you have to do is use **DateDifference**:

```
NumberDay is int
NumberDay = DateDifference("20100101", Today())
InfoBuild("Numbers of days between %1 and %2: %3", ...
    DateToString("20100101", maskSystemDate), ...
    DateToString(Today(), maskSystemDate), NumberDay)
```

► To test this feature in our example, click the "DateDifference" button.



Note

You also have the ability to get the duration between today's date and a specific date in plain text (two days ago, in 3 weeks, ...). All you have to do is use:

- the "Relative duration" display mask,
- **DateToString** associated with the **maskDateRelativeDuration** constant. For example:

```
// The date is 01/22/2013
Res = DateToString("20130101", ...
    maskDateRelativeDuration) // Returns: 3 weeks ago
```

## What is the day of the week corresponding to a given date?

To find out the day corresponding to a given date, all you have to do is use **DateToDayInAlpha**:

```
D is string
D = DateToDayInAlpha("17890714")
Info("July 14th, 1789 was a " + D)
```

► To test this feature in our example, click the "DateToStringInAlpha" button.

## Practical exercise

Now that you are a bit more familiar with the dates: what is the purpose of the following code?

```
DateToString (Today () , "DDDD DD MMM YYYY")
```

Answer: Displays the literal date (the day and the date in everyday's language).

## Handling Date variables

Specific types of variables are available for the Date, DateTime and Duration values.

These variables can be handled like "strings". Each one contains a value according to the following table:

Type	Default format
Date	YYYYMMDD
DateTime	YYYYMMDDHHMMSSCC
Duration	YYYYMMDDHHMMSSCC

The "Date" type supports the dates from 01/01/0001 to 12/31/9999 (that should keep us going for a while!).

More seriously, to handle the value of these variables, you have the ability to use the following syntaxes:

```
MyDate is Date = "20121021"
Info(DateToString(MyDate)) //Displays "21/10/2012"
MyDate..Year = MyDate..Year + 1
MyDate..Month = MyDate..Month + 1
MyDate..Day = MyDate..Day + 1
Info(DateToString(MyDate)) //Displays "22/11/2013"
```

In this code, **Year**, **Month** and **Day** are WLanguage properties.

- In the "PAGE\_Date" page, the "Code sample" button is used to calculate the number of days until the next January 1st by using a Date variable.

The following code is used:

```
MyDate is Date
// MyDate is automatically initialized to today's date

// Calculate the date of the next January 1st
NewYear is Date
NewYear..Year++
NewYear..Day = 1
NewYear..Month = 1
MyDuration is Duration = NewYear - MyDate

Info(StringBuild(...
"There are %1 days left before the next New Year's Eve",...
" on %2.", MyDuration..InDays, DateToString(NewYear))
```

## A practical exercise? It's your turn!

A small exercise will be used to implement the different operations that can be performed on the dates:

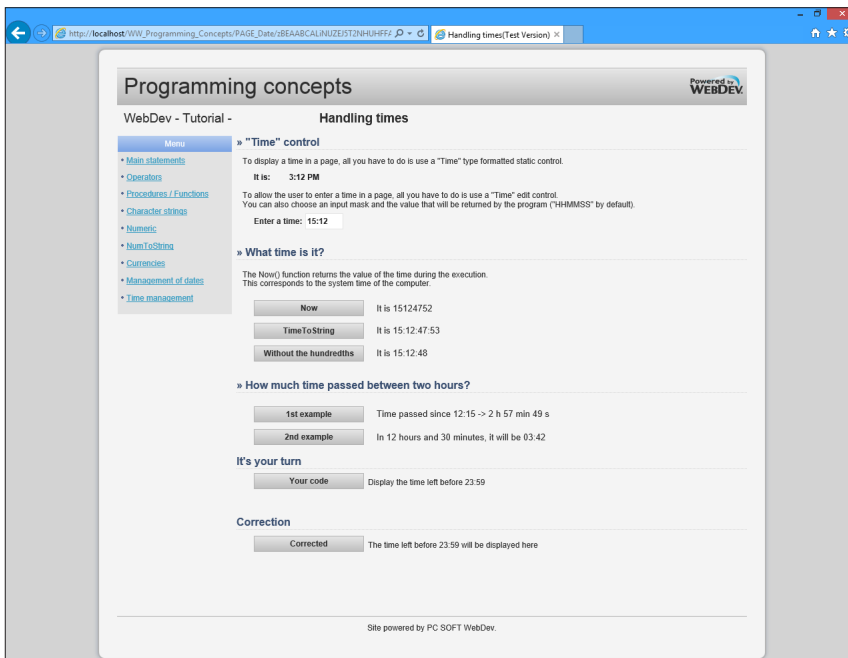
- ▶ In the "PAGE\_Date" page, in the "It's your turn" area, an exercise is proposed:
  - In the "It's your turn" button, write the code used to calculate the date in 180 days and display this date in letters.
  - Run the page test to run the test of your code.

Correction: The "Correction" button contains the solution.

## The times (server code and browser code)

### Practical example

- ▶ In the "WW\_Programming\_Concepts" project, open the "PAGE\_Time" page. This page illustrates the explanations given in the rest of this lesson.
- ▶ Run the test of this page.



## What time is it?

To find out the time, all you have to do is use **Now** (or **TimeSys**). **Now** returns the current time of your system as a character string in "HHMMSSCC" format:

```
Info("It is " + Now())
```

► To test this feature in our example, click the "Now" button.

► You want to display the time in "HH:MM:SS:CC" format? The code becomes:

```
Info("It is " + TimeToString(Now()))
```

**TimeToString** transforms a time in "HHMMSSCC" format into a string in "HH:MM:SS:CC" format.

► To test this feature in our example, click the "TimeToString" button.

If you do not want to display the hundredths of a second, the code is:

```
Info("It is " + TimeToString(Now(), "HH:MM:SS"))
```

All you have to do is specify the display format of the time with **TimeToString**.

► To test this feature in our example, click the "Without the hundredths" button.

## How much time has passed between two given times?

It is 17:25. How much time has passed since 12:15? The code is as follows:

```
// Displays the time passed since 12:15
Diff is int = TimeDifference("1215", Now())
Result_Time is Time = IntegerToTime(Abs(Diff))

IF Diff < 0 THEN
  InfoBuild("Time passed before 12:15 -> %1 h %2 min %3 s ", ...
  Result_Time..Hour, Result_Time..Minute, ...
  Result_Time..Second)
ELSE
  InfoBuild("Time passed since 12:15 -> %1 h %2 min %3 s ", ...
  Result_Time..Hour, Result_Time..Minute, ...
  Result_Time..Second)
END
```

**IntegerToTime** transforms an integer (that corresponds to the number of hundredths of a second passed since midnight (or 00:00)) into a time in "HHMMSSCC" format.

**TimeToInteger** performs the reverse operation.



Caution !

- No ":" must be found in the time passed in parameter to **TimeToInteger** otherwise the result would be incorrect.
- To calculate durations exceeding 24 hours, use the Date type, Time type, ...

► To test this feature in our example, click the "1st example" button.

The same calculation can be performed by using the Time and Duration variables.

Specific types of variables are available for the Date, Time, DateTime and Duration values. These variables can be handled like "strings". Each one contains a value according to the following table:

Type	Default format
Time	HHMMSSCC
Duration	YYYYMMDDHHMMSSCC

The code can become:

```
In12H is Time
In12H..Hour+= 12
In12H..Minute+= 30
InfoBuild("In 12 hours and 30 minutes, it will be %1", ...
    TimeToString(In12H, "HH:MM"))
```

- To test this feature in our example, click the "2nd example" button.
- Close the browser to go back to the editor.

### A practical exercise? It's your turn!

A small exercise will be used to implement the different operations that can be performed on the dates:

- In the "PAGE\_Time" page, in the "It's your turn" area, an exercise is proposed:
  - In the "Your code" button, write the code used to display the time left before 23:59 in HH:MM format.
  - Run the page test to check your code.

Correction: The "Correction" button contains the solution.

## Calculations with dates and times

Several WLanguage functions can be used to manage the dates and times and to perform different calculations.

See the online help (keyword: "Date, WLanguage functions") for more details.

## APPENDICE 3. CODE EDITOR: QUESTIONS/ANSWERS

**This lesson will teach you the following concepts...**

---

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



Estimated time: 10 min

## Questions/Answers

### Question How to view the element to which the current process belongs ?

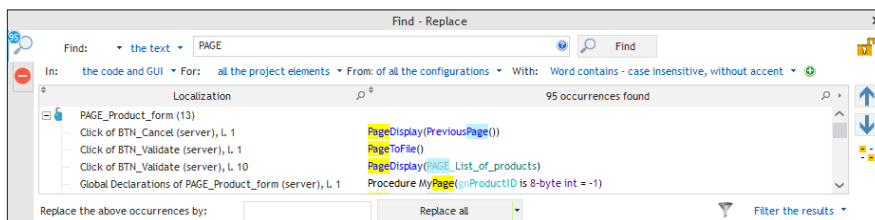
To view the element corresponding to the current process, on the "Code" pane, in the "Navigation" group, click "Go to the object" (CTRL + ALT + F2). The page containing the requested element is displayed.

### Question How to print the source code?

To print the current source code, click  in the "Home" pane or press [CTRL] + [P].

### Question How to 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:



```

PROCEDURE AddBasket()

    aaCharacteristics is associative array of Variants
    sRefAddedProduct is string
    nNbAdditions is int
    nVariationID is int

    // Read the product record
    HReadSeekFirst(product, productid, EDT_ProductID)
    IF HFound(product) THEN

        IF HFound(product) THEN

            // Number of products to add
            nNbAdditions = COMBO_QuantityToAdd..DisplayedValue

            // Identifier of the variation
            nVariationID = GetVariationID()

            // Builds the array of characteristics
            IF arrBuildArrayOfCharacteristics(aaCharacteristics) = True THEN

```

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". The corresponding shortcut is [CTRL] + [SHIFT] + [G].

**Question** **Is there a method to easily display the syntax or the help about a function?**

When typing a function, the syntax of the function is displayed:

- in a tooltip found below the line currently edited. An information tooltip is displayed for each parameter (including for the result of the function).  
If several syntaxes are available, press [CTRL] + [ALT] + Right Arrow or [CTRL] + [ALT] + Left Arrow to go from a syntax to another one.
- in the status bar of the 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 currently typed.

Examples of assisted input for **HReadFirst**: the <Wizard> option is used to start a code wizard. This wizard asks you questions regarding the use of the function and it automatically generates the corresponding code.

A help page is associated with all the WLanguage functions and properties. 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?**

- [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).
- [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 move from a process to another one, press [F2] repeatedly. To go back to the initial process, press [CTRL]+[F2] the same number of times.
- [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 time to discuss all the available topics (there are hundreds, even thousands!). WEBDEV proposes several other features not presented in this tutorial:

- 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 21**!