

DEVELOP 10 TIMES FASTER



SUMMARY

Introduction	
Preliminary points	15
Tutorial overview	15
Legend of the symbols in this tutorial	16
How to access the online help	16
If you are familiar with WEBDEV 24	18
What is WEBDEV used for?	19

PART 1 - DISCOVERING WEBDEV

Lesson 1.1. Discover WEBDEV

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

Lesson 1.2. Web development and WEBDEV

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

Lesson 1.3. My first pages

Overview	40
Opening project	40
Creating a "Form" page	41
Creating the page	41
Creating controls	42
Improving the page	46
Modifying the style of "Login" and "Password" controls	48
Page test	50

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

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

This documentation is not contractually binding. Modifications may have been made to the software since this guide was published. Please check the [online help](#).

All product names or other trademarks mentioned in this publication are registered trademarks of their respective owners.
© PC SOFT 2020: This publication may not be reproduced in part or in full without the express consent of PC SOFT.

Saving the data in a data file	51
Creating the data file	51
Saving data	53
Page test	54
Viewing the data typed	54
Adding input checks	55
Mandatory input	55
Checking the password	56
Simple search in a database	57
Creating the page	57
Creating controls	58
Page test	61
Conclusion	61

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
Native Connector (Native Access)	69
Direct ODBC access	70
OLE DB access	70
ODBC access via OLE DB	70

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 data file	75
Creating a data file: creating the data file and 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 WEBDEV site in Session mode

Principle for displaying a WEBDEV site in Session mode	101
The WEBDEV session	102
The page contexts	102
Programming	102

Lesson 3.3. Pages for addition and modification

Overview	104
Creating a page in Session mode used to list the products	104
Creating a page that is using a template	104
Creating controls	106
Page test	112
Modifying products via a "Product form" page in Session mode	113
Creating the page	113
What should this page do?	113
Creating edit controls	114
Displaying data in the page	116
Creating buttons	117
Managing the product image	120
Displaying the form from the list of products	123
Adding a new product via the "Product form" page	126
Test of product addition	129
Viewing the records	130

PART 4 - INTERNET SITE WITH DATA

Lesson 4.1. Overview

Overview of the site created in this part	134
---	-----

Lesson 4.2. Principle for displaying an AWP site

Principle for displaying a WEBDEV AWP site	137
What is an AWP page?	137
Operating mode of AWP sites	137
How to share information (values) between AWP pages?	139

Lesson 4.3. Creating AWP pages	
Overview	141
Creating an AWP page used to list new products	141
Creating the page	141
Creating the list of new products	143
Viewing the details of a new product	148
Creating the page	148
What should this page do?	149
Creating controls	150
Displaying data in the page	154
Displaying the form from the list of products	154
Managing styles	156
Importing a CSS style sheet	156
Applying CSS styles	157
Linking the Internet site and the Intranet site	159
Creating a page used to display the general sales conditions	161
Creating the page	161
Creating the control	161
Conclusion	163

PART 5 - LET'S CONTINUE THE DEVELOPMENT

Lesson 5.1. Multicriteria search	
Overview	168
Creating the query used to find orders	169
Creating the query	169
Query test	173
Using parameters in the query	174
Test of query with parameters	178
Creating the page proposing the multicriteria search	179
Creating a page	179
Creating the controls used to configure the criteria and to display the result	179
Optimizing the page display	195
Lesson 5.2. Printing an invoice	
Overview	197
Principle for printing on Internet	197
Direct print	197
Generating documents in HTML, PDF or XML format	198
What is a report made of?	198
Creating the "Invoice" report	199
Creating the query	199
Creating the report based on a query	202
Modifying the "Invoice" report	208
Displaying the printed report from a button	212
Implementing the print	212
Print test	213

Lesson 5.3. Sending an email	
Overview	216
A popup page to send emails	217
Creating the popup page	217
Creating the controls used to type the email characteristics	217
Sending the email	219
Improving the page	220
Closing the popup page	220
Opening the popup page	222
Lesson 5.4. Identifying the user: the user groupware	
Overview	225
Integrating the user groupware	226
Configuring the user groupware	227
Including the customer section in the site	227
Site test	229
Creating users and groups	230
Defining rights	232
Site test	233
Disabling the management of user groupware	234
Lesson 5.5. Managing the multilingual feature	
What is a multilingual site?	236
Choosing the project languages	236
Localizing the project elements	239
Localizing an image	240
Localizing the menu	241
Localizing controls	241
Localizing a programming message	242
The translation tools	243
Direct input of translations	243
Translation with WDMMSG and WDTRAD	244
Other elements to translate: the framework messages	244
Programming the change of language	246
Adding a menu option	246
Programming	247
Project test	248
Lesson 5.6. SEO	
Overview	250
Reminder : AWP mode	250
The available methods	250
SEO wizard	251
Properties of pages	253
Update frequency	253
Importance in the site	253
Navigation map	254
Physical naming of pages	254

PART 6 - DEPLOYING A SITE

Lesson 6.1. Deploying a site

Overview	258
Required configuration	258
Installing and configuring the "WEBDEV Application Server - 10 Connections"	259
Installing the "WEBDEV Application Server - 10 Connections"	259
Configuration via the Hosting Control Center	261
Creating a deployment account.....	263
The different deployment modes.....	264
A detailed example: deployment via HTTP.....	265
Correspondence between the directories of the development computer and the deployment	270

Lesson 6.2. Managing a site

Overview	272
Local WEBDEV administrator	272
Remote WEBDEV administrator.....	274

PART 7 - SPECIFIC WEB FEATURES

Lesson 7.1. The styles

Overview	278
Discovering styles	278
A control: several elements	278
Discovering the styles via an example.....	279
Selecting the CSS style of a control.....	280
Editing the CSS style of a control.....	280
The WEBDEV styles	282
Implementing styles: a practical example	284
Overloading a CSS style for a control.....	285
Modifying a CSS style for all controls	285
Styles and rich text	286

Lesson 7.2. The cookies

What is a cookie?	288
What is a cookie made of?.....	288
Practical example.....	289
How to use the cookies?	289

Lesson 7.3. Secure transactions and payment

Securing the information and the pages via TLS/SSL	291
Overview	291
Implementing secure transactions via the TLS/SSL protocol.....	291
Transactions secured by TLS/SSL in a WEBDEV site	291

Secure payment.....	291
Overview	291
System for secure payment in a WEBDEV site.....	293

Lesson 7.4. Anchoring and zoning

Overview	295
Creating a page in zoning mode.....	295
Implementing the anchoring.....	298

PART 8 - RESPONSIVE WEB DESIGN

Lesson 8.1. Responsive Web Design

What is Responsive Web Design?	304
Benefits	304
The concepts in WEBDEV.....	304

Lesson 8.2. Responsive Web Design and WEBDEV

Overview	306
My first page in Responsive Web Design mode	306
Creating a page	306
Fluidity and Responsive Web Design	308
Resolution slices	309
How to adapt the controls to the different slices?	311

Lesson 8.3. Advanced concepts

Controls specific to Responsive Web Design	315
The anchors.....	316
Positioning anchors.....	317
Size anchors	318
Order of controls in Responsive Web Design mode.....	319

PART 9 - USEFUL FEATURES OF WEBDEV

Lesson 9.1. The internal components

Overview	328
Step by step.....	328
Step 1: Creating an internal component.....	328
Step 2: Using the internal component.....	332

Lesson 9.2. Automatic management of errors

Overview	336
Operating mode.....	336
Implementation.....	336
Types of affected errors	337
Automatic management of errors: a training example.....	337

Lesson 9.3. Import/Export

Importing elements.....	342
Exporting elements	343
Specific import operations	344
Importing a WINDEV project	344
Importing an HTML page.....	347

PART 10 - OPTIMIZING AND DEBUGGING A PROJECT

Lesson 10.1. Overview

Overview	352
Opening project.....	352

Lesson 10.2. Project audits

What is an audit?	354
Static audit.....	354
Procedure not run.....	356
Orphan element	357
Cleaning the project	358
Dynamic audit.....	358

Lesson 10.3. Performance profiler

Overview	362
Starting the performance profiler	362
Studying the result.....	363

Lesson 10.4. Debugging a project

Overview	369
Using the debugger.....	369

PART 11 - MANAGING A HFSQL CLIENT/SERVER DATABASE

Lesson 11.1. Introduction

Overview	378
Why switch a site to HFSQL Client/Server mode?	379

Lesson 11.2. Implementing a Client/Server database

Overview	381
Installing a local HFSQL server.....	381
Creating a site that is using a HFSQL Client/Server database	382
Adapting a site to use a HFSQL Client/Server database.....	382
Overview	382
Adapting the example	383
Features available in HFSQL Client/Server mode	386

Lesson 11.3. Managing a Client/Server database

Overview	388
Configuring the computers	388
The HFSQL Control Center	388
Creating a user account in the HFSQL Control Center	390
Saving the database.....	395
Conclusion.....	395

PART 12 - SOURCE CODE MANAGER

Lesson 12.1. SCM

Introduction.....	400
SCM (Source Code Manager).....	400
Principle of SCM	400
Creating the repository.....	401

Lesson 12.2. Integrating a project in SCM

Integrating a project in SCM	403
Adding the project into SCM	403
Opening a project from SCM.....	407
Configuring SCM.....	408

Lesson 12.3. Using a project in SCM	
Overview	410
Modifying a project parameter	410
Modifying a project page	411
Modifying the checked-out element	412
Checking the checked-out element back in	412
Synchronizing the project	414
Offline mode	415
SCM administrator	415
Disconnecting from SCM	416

PART 13 - THE WLANGUAGE BASICS

Appendix 1. Vocabulary	
Main terms used	420
Appendix 2. The variables	
What is a variable?	428
Declaring a variable	429
Assignment and use	429
The types of variables	430
The scope of variables	430
Global scope	430
Local scope	431
Summary scope diagram	431
Simple operations on the variables	432
Tips	433
Details of variable type: the String variables	433
The String type	433
Practical example	435
Details of another variable type: the arrays	438
Declaration	438
Filling an array and accessing the elements	439
Advanced arrays	439
Appendix 3. The conditional statements	
Overview	441
The IF and SWITCH statements	441
The IF statement	441
The SWITCH statement	442
Practical example: Using the IF and SWITCH statements	443
Project used	443
Creating the page	444
Creating the page controls for the conditional IF statement	444
Conditional IF statement	445
Creating the page controls for the conditional SWITCH statement	446
SWITCH statement	448

Appendix 4. The loops	
Overview	451
The FOR statement	451
The LOOP statement	451
The WHILE statement	452
Practical example: Using loops	453
Project used	453
Creating the page	453
Creating controls	453
Page test	457

Appendix 5. The procedures	
Overview	459
Types of procedures	459
Creating and calling a procedure	459
Global procedure	459
Local procedure	459
Internal procedure	460
Calling a procedure	460
Procedure parameters	460
What is a parameter?	460
How to use the parameters?	461
Mandatory or optional parameters?	462
Procedure result	462
Practical example: Using a procedure	462
Project used	463
Implementation	463
Creating and using a procedure	466
Conclusion	468

Appendix 6. Questions/Answers	
Questions/Answers	470
How to view the element to which the current event belongs?	470
How to print the source code?	470
How to perform a "find and/or replace"?	470
What is the meaning of "+" and "-" signs in the code editor?	471
Is it possible to identify the person who wrote a code line?	471
Is it possible to find out the code line number?	471
Is there a method to easily display the syntax or the help about a function?	472
What are the useful shortcuts in the code editor?	472

Conclusion

INTRODUCTION

Preliminary points

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

This tutorial does not cover all the features in WEBDEV.

Spend at least a few hours to follow this tutorial and learn WEBDEV: it is a good investment!
If you try to start developing a site straight away, you will lose time, much more than the few hours spent on this tutorial.

The tutorial has been designed to be followed according to one of the following methods:

- Method 1: follow all the detailed exercises in each lesson (recommended method).
- Method 2: if you are in a hurry and already have significant experience, you can simply read this tutorial without doing the exercises (all the actions are illustrated). However, in order to quickly assimilate the main concepts, we recommend that you follow the tutorial step by step.

As WEBDEV is constantly evolving, the images of windows and pages in the tutorial may differ from the pages in your product.

Tutorial overview







This tutorial has been designed to progressively teach you how to use WEBDEV. By following this tutorial:

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

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

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

Legend of the symbols in this tutorial

-  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 can be opened from the WEBDEV home page (Ctrl + <).
-  This symbol introduces a "Tip": reading the associated text is strongly recommended.
-  This symbol introduces a "Warning": reading the associated text is essential.
-  This symbol introduces a "Note": reading the associated text is recommended.
-  This symbol introduces a feature specific to Internet: reading the associated text is strongly recommended.




Remark

The tutorial may have evolved since this document was printed. Feel free to consult the online version (<http://doc.windev.com>).

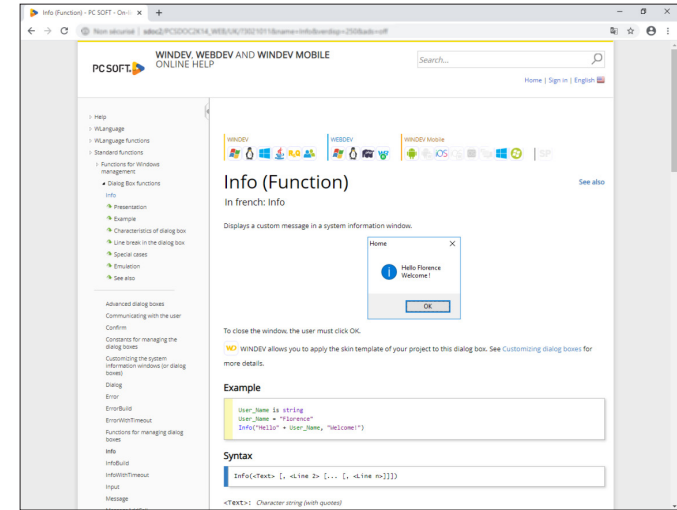
How to access the online help

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

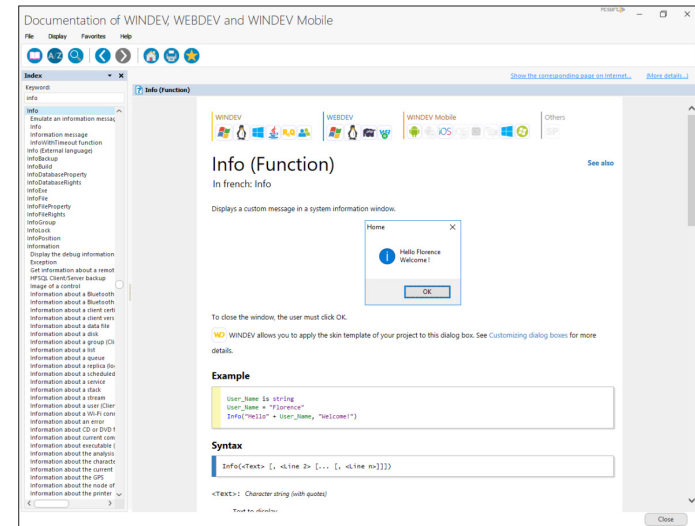
The online help is available at any time in WEBDEV:

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

- ▶ The help can be displayed:
 - in an Internet browser, if you have access to Internet (default mode):



- in a specific "help browser":



**Remark**

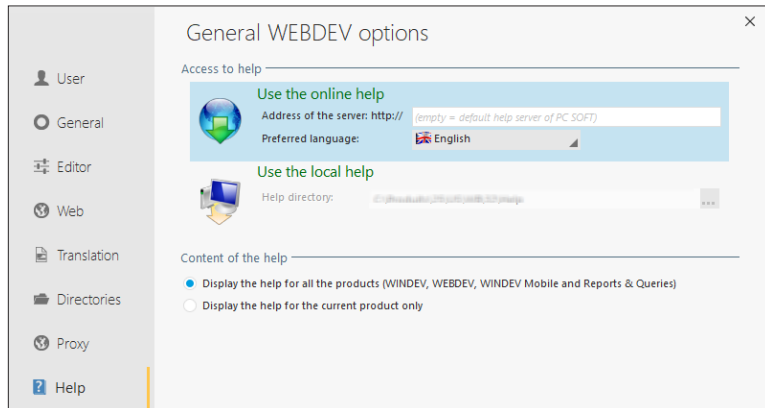
We advise you to check the online help on Internet rather than the local online help. Indeed, the online help on Internet is updated on a regular basis.

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, ...

Remark: If you have no access to Internet, you have the ability to start the local help from the product:

1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV".
2. Display the "Help" tab.



3. Select:

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

If you are familiar with WEBDEV 24

If you are already familiar with WEBDEV 24, this tutorial will still be very beneficial: it's a good opportunity to "review" the WEBDEV features!

What is WEBDEV used for?

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

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

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

Unlike other programming languages, there is no need to find and add modules to be able to design, check and 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!

Discovering
WEBDEV



LESSON 1.1. DISCOVER WEBDEV

This lesson will teach you the following concepts

- Starting WEBDEV.

 Estimated time: 5 mn

Overview

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

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

WEBDEV allows you to create:

- static Internet/Intranet sites. These sites manage data that does not vary (company showroom site, site available on a DVD-ROM, etc.).
- dynamic Internet/Intranet sites, that manage data. The WEBDEV sites access all 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 25 (if not already done).
- ▶ **When WEBDEV 25 is launched for the first time**, a welcome wizard will start:
 - If you have 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.
- ▶ **If you have already launched WEBDEV 25**, identify yourself if necessary. The development environment starts. The WEBDEV home page is displayed. This home page is used to:
 - display the news,
 - 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. To do so, from the WEBDEV home page:
 - Click "Tutorial".
 - Double-click "Full WEBDEV Site (Answer)".
 - The corresponding project is opened in the editor. The project dashboard is displayed. The project dashboard allows you to check the progress of a project via several elements (called widgets).

Development environment

The editor

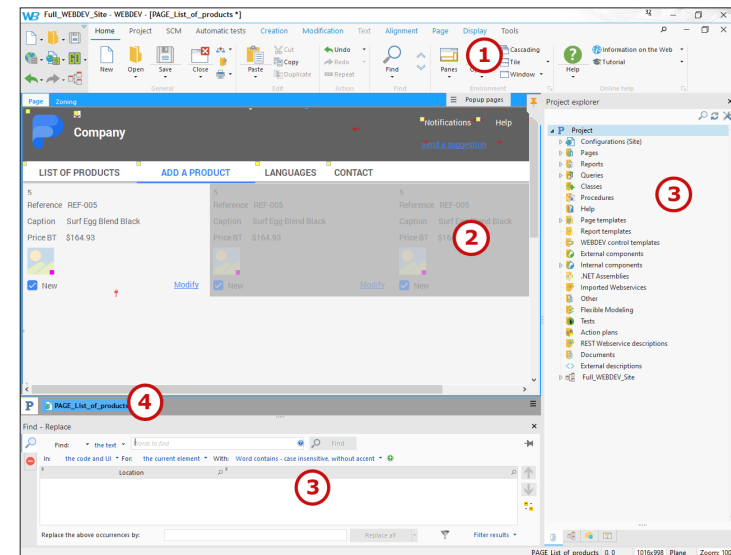
The development environment of WEBDEV includes a specific interface and several editors allowing you to create the different elements of your applications.

For example, the page editor is used to create pages, the report editor is used to create reports, ...

▶ To discover WEBDEV, we will open the page used to list the products:

1. Press Ctrl + E.
2. In the window that is displayed, type the name of the page to open: PAGE_List_of_products.
3. Validate. The page is displayed in the editor.

All editors are using the same environment:



1. **Menu of editors**, displayed in ribbon format (we'll see how to use it in the next paragraph).
2. **Current editor** (here, the page editor). This space allows you to see 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 see all 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 ribbon format. This ribbon includes panes in which the different options of editors are grouped.

We are going to take a closer look at the main ribbon elements, 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 (1)

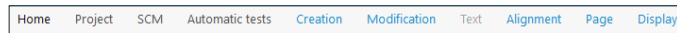


The button area groups the **quick access buttons**. These buttons are used to perform the most usual operations, common to all editors: save, open, create, ...

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

The arrow at the top right of the button area allows you to find the toolbars and drop-down menus of the old editor interface.

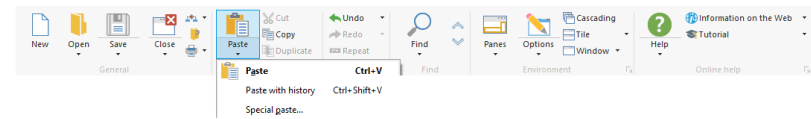
The pane area (2)



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

- the current pane: The name of the pane appears in bold and is underlined by a blue line.
- the popup panes, specific to the current element: The pane name is displayed in blue.
- the available panes: The pane name is displayed in black.

The option area (3)



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

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

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

In this tutorial, to identify a menu option, we will be talking about panes and groups. For example: "To display the help, on the "Home" pane, in the "Online help" group, click "Help"."

The environment colors

The environment is using a light theme by default.

Several other themes are also available:

- Light theme, grey ribbon. In this mode, the menu bar is not colored anymore: it is grayed.
- Gray theme. In this mode, the environment and the interface windows are displayed on a light gray background.
- Dark theme. In this mode, the environment and the interface windows are displayed on a black or dark gray background.

► To modify the theme used by the environment:

1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV".
2. In the "Editor" tab, in the "Themes" area, select the theme to use.
3. Validate. The theme will be taken into account during the next WEBDEV start.

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

► To continue this lesson, we are going to close the current project:

1. On the "Home" pane, in the "General" group, expand "Close" and select "Close the project".
2. Save the modified elements if necessary.
3. The WEBDEV home page is displayed.

WLanguage


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

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

LESSON 1.2. WEB DEVELOPMENT AND WEBDEV

This lesson will teach you the following concepts...

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

 Estimated time: 30 mn

Principle of Browser/Server

How does it work?

An Internet or intranet site operates as follows:

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

Therefore, there are two types of events:

- Events run at browser level, on the Web user's computer.
- Events 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.

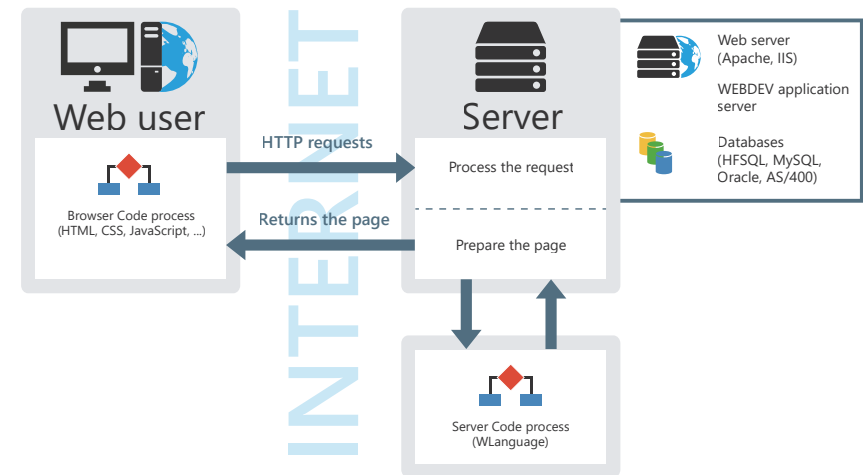
The server code is run in WLanguage.

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 server and browser events, we have prepared a simple example:

1. Start WEBDEV 25 (if not already done). Display the WEBDEV home page if necessary: press Ctrl + <.
2. Open the "WEBDEV concepts" project. To do so, in the home page, click "Tutorial" and select the project named "WEBDEV concepts (Exercise)".



Remark

If the UAC is enabled in Windows, a specific window may be displayed. This window indicates that the WD250Admin.exe program will be run on the current computer. Grant the authorization. This program corresponds to the local WEBDEV administrator that is used to run the test of sites developed with WEBDEV. Its features will be presented later in this tutorial.

- Open the "PAGE_Registration" page in the editor: double-click its name in the "Project explorer" pane.

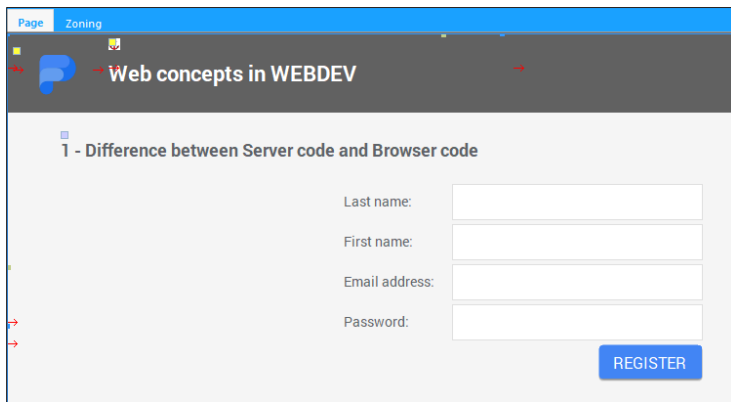


Remark

The "Project explorer" pane allows you to display in the environment a list of all the elements found in the project. These elements are grouped by theme: Pages, Procedures, ...

To display the "Project explorer" pane:

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 control. The "REGISTER" Button control must do two things:

1. Check that all controls have been filled.
2. Save the values of controls in the database.

- Let's see the WLanguage code associated with the Button control:

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

```

1  Initializing of BTN_Register (server)
2  Click (onClick) on BTN_Register (browser)
3  // Check whether the "EDT_LastName" control is equal to an empty string (excluding spaces and punctuation)
4  IF EDT_LastName == "" THEN
5  // The control is empty, display an error message to the user
6  Error("Enter your name")
7  // Return in edit into the "EDT_LastName" control (without running the rest of code)
8  ReturnToCapture(EDT_LastName)
9  END
10 // Check whether the "EDT_FirstName" control is equal to an empty string (excluding spaces and punctuation)
11 IF EDT_FirstName == "" THEN
12 // The control is empty, display an error message to the user
13 Error("Enter your first name")
14 // Return in edit into the "EDT_FirstName" control (without running the rest of code)
15 ReturnToCapture(EDT_FirstName)
16 END
17 // Check whether the "EDT_Email_address" control is equal to an empty string (excluding spaces and punctuation)
18 IF EDT_Email_address == "" THEN
19 // The control is empty, display an error message to the user
20 Error("Enter your email address")
21 // Return in edit into the "EDT_Email_address" control (without running the rest of code)
22 ReturnToCapture(EDT_Email_address)
23 END
24 // Check whether the "EDT_Password" control is equal to an empty string (excluding spaces and punctuation)
25 IF EDT_Password == "" THEN
26 // The control is empty, display an error message to the user
27 Error("Enter your password")
28 // Return in edit into the "EDT_Password" control (without running the rest of code)
29 ReturnToCapture(EDT_Password)
30 END
31 // The browser code is ended, the page will send the values of controls to the server and will
32 // ask it to run the server code of the button
33 Click on BTN_Register (Server) Ajax
34 // Select the customer structure
35 HReset(Customer)
36 // Retrieve the values of controls in the customer structure directly
37 ScreenToFile()
38 // Add the customer into the database
    
```



Remark

The code editor allows you to write the WLanguage 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.

Remark: Events are displayed in the order in which they will be run.

For example, the events associated with the Button control are:

- Initialization.
- Browser click.
- Server click.

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

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

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

```
// Check whether the "EDT_LastName" control is equal to empty string
// (excluding spaces and punctuation)
IF EDT_LastName ~= "" THEN
  // The control is empty, display an error message to the user
  Error("Type your last 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 typed to the server. Then, the server runs the server code. In the server code, you have the ability to process the information received.

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

```
// Reset the customer structure
HReset(Customer)

// Get the values of controls in
// the customer structure
ScreenToFile()

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

This operation cannot be performed in browser code because the database is common to all 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 cross in the top right corner).

Intranet/Extranet/Internet

Principle

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 by managing logins, passwords, rights, etc.

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

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

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

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

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

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

Drawback: The search engines cannot index this site.

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

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

Practical example

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

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

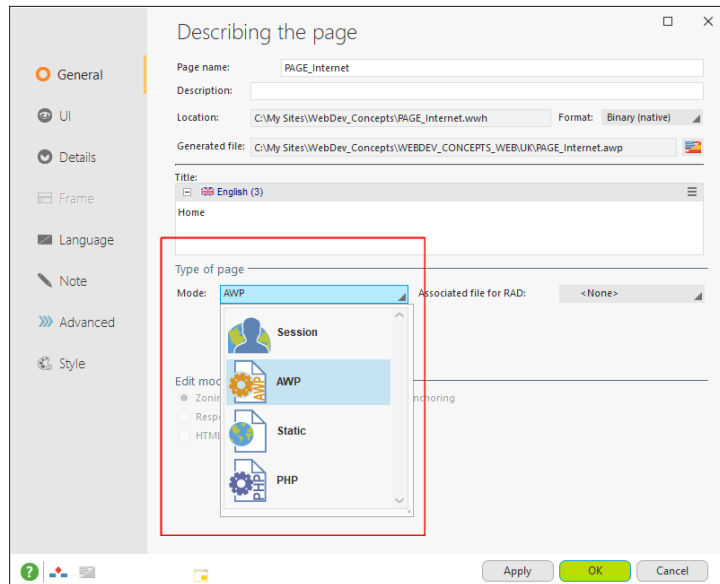


Tip

To quickly find a page in the current project, press Ctrl + E. A window is displayed, allowing you to perform a search on all 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.

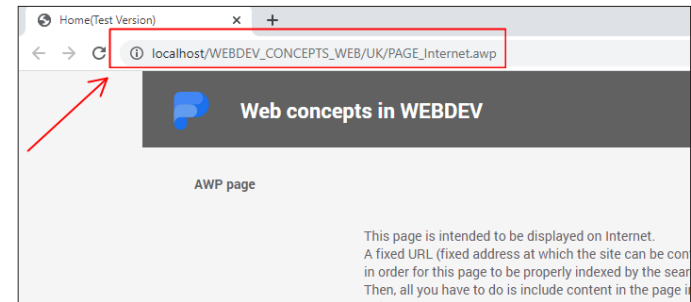
2. Let's check the page type:

- 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 a page in AWP mode.



- Validate the page description window.
3. Run the test of this page: click the icon among the quick access buttons.
 4. The page is displayed in your default browser.

5. Study the address displayed in the browser: this page has a fixed address. If you copy this address and if you paste it in a new browser, the same page will be displayed!



Remark: if you use the IIS Express Web server, "localhost" can be followed by an identifier.

6. Close the browser that displays the test page.

► 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" pane.

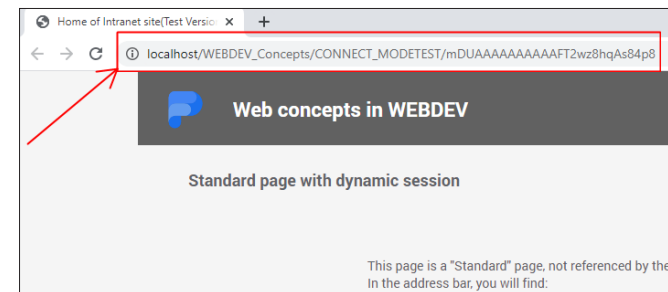
2. Let's check the page type:

- 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 a page in session mode, a mode used for Intranet/ Extranet sites.

3. Run the test of this page: click the icon among the quick access buttons.

4. The page is displayed in the browser.

5. Study the address displayed in the browser. The URL has the following format: **http://localhost/WEBDEV_Concepts/CONNECT_TESTMODE/<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.



Remark: if you use the IIS Express Web server, "localhost" can be followed by an identifier.

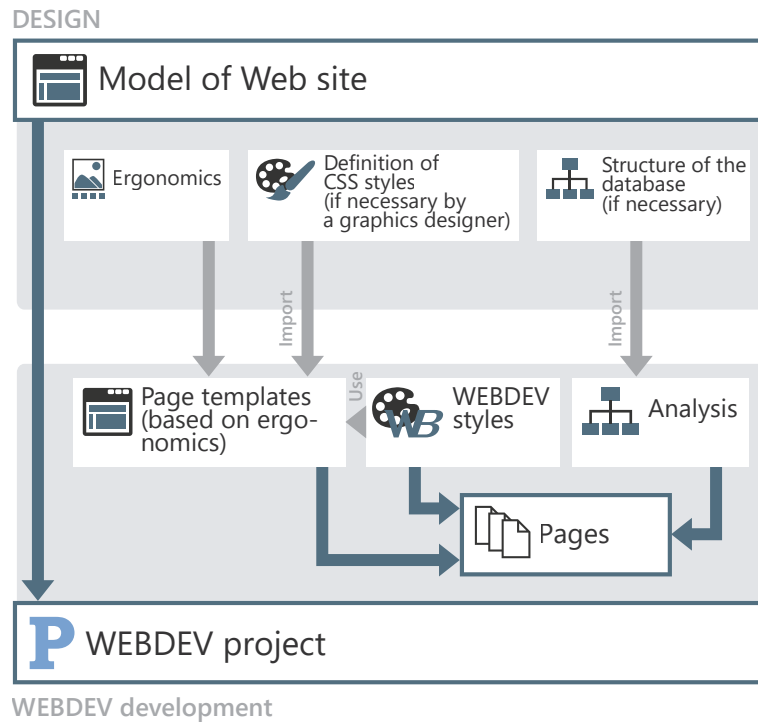
6. Close the browser.

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



LESSON 1.3. MY FIRST PAGES

This lesson will teach you the following concepts...

- Creating a form page.
- Saving data.
- Checking inputs.
- Simple search in a database.



Estimated time: 30 mn

Overview

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



Remark

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 project

- ▶ Start WEBDEV 25 (if necessary).
- ▶ Open the "My_First_Pages" project.
 1. Display the WEBDEV home page (Ctrl + <).
 2. Click "Tutorial", then "My first pages (Exercise)".



Important!

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




Answer

A corrected project is available. This project contains the different pages created in this lesson. To open the corrected project, in the WEBDEV home page, click "Tutorial" and select "My first pages (Answer)".

Creating a "Form" page

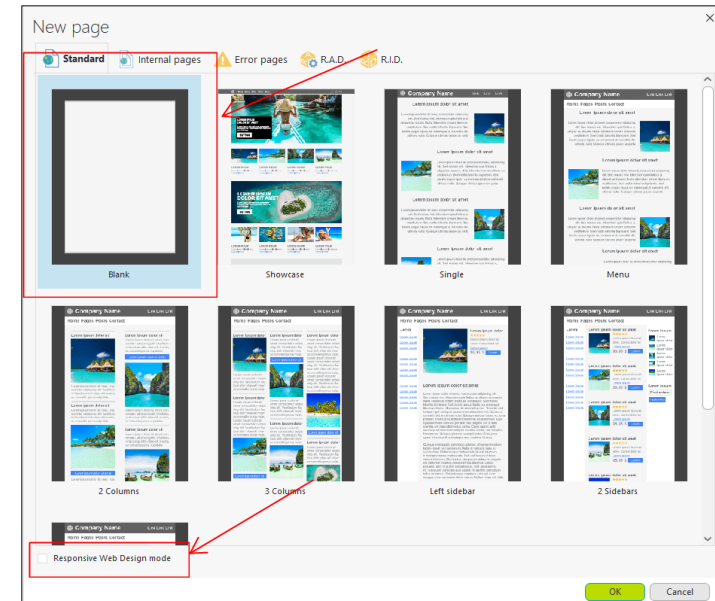
We are going to create a registration form. This form will contain several controls used to identify the person.

Creating the page

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



2. The window for creating a new element appears. This window is used to create all elements that can be associated with a project.
3. Click "Page" then "Page". The page creation wizard starts.
4. Select "Blank".

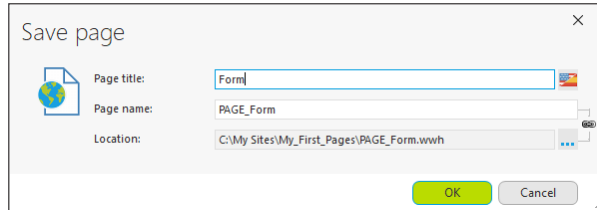


5. Make sure that "Responsive Web Design mode" is unchecked. In this part, we will focus on the creation of simple pages. The pages in Responsive Web Design mode will be presented in Part 8 of this tutorial.

6. Validate ("OK" button). The page is automatically created in the editor. The window for saving the page is displayed. This window displays:

- The page title. This title will be displayed in the title bar of browser.
- The page name. This name corresponds to the logical page name. This name will be used to handle the page by programming.
- The location. This location corresponds to directory in which the file corresponding to the page will be saved.

7. Type the page title: "Form".



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

9. Validate ("OK" button).



Remark

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

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

Creating controls


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

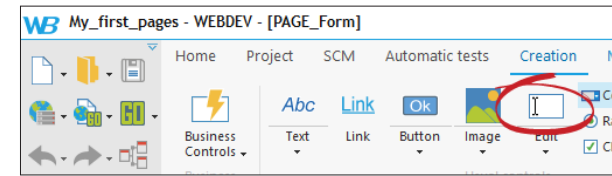
The form includes the following controls:

- Last name.
- First name.
- Address.
- Email.
- Login.
- Password.

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

► To create an edit control:

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



2. The control currently created follows the mouse movement.

3. Click inside the blank page to create the control. The control is created and handles are displayed to indicate that the control is selected.



Remark

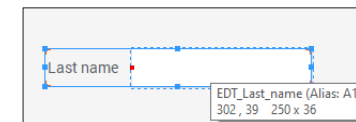
You must create a control? Display the "Creation" pane of WEBDEV: all available controls are accessible in this pane.

► To modify the control caption:

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



2. Type "Last name" and press Enter to validate. The caption is immediately modified. The control name is also modified: it appears in the tooltip when hovering over the control: EDT_Last_name.





Remark

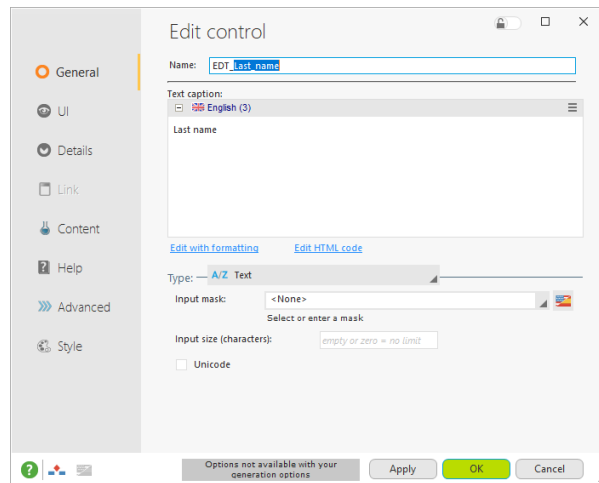
Study the control name proposed by WEBDEV: this name starts with the letters "EDT_". This prefix is automatically added because the project is using a code style.

The code style 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 saving the page).
- an edit control starts with "EDT_".
- a button starts with "BTN_", etc.

If you do not want to use this code style, you can simply disable it: in the ribbon, on the "Project" pane, in the "Other actions" group, expand "Code style" and uncheck "Use code style".

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

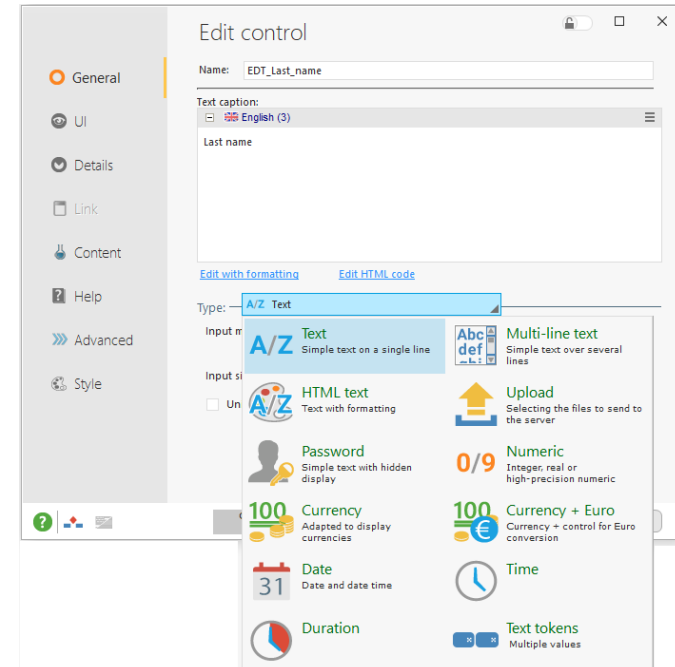


- We are going to modify the maximum input size: all you have to do is type the new size (30 in our case) in the "Input size (characters)" control. Several other characteristics can be modified. Click "OK" to save the changes.

- Now, it's your turn. Use the same method to create the following edit controls, below the "Last name" control:

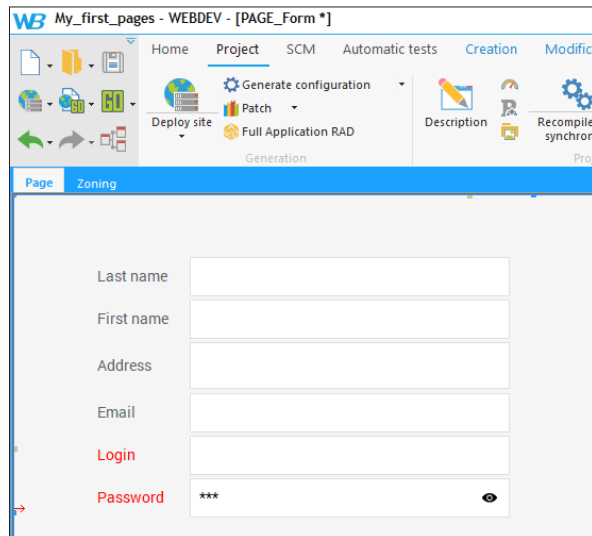
Caption	Type	Size
First name	Text	30
Address	Multi-line text	
Email	Text	255
Login	Text	10
Password	Password	10

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



The form base is created. We are now going to improve it.

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.

Improving the page

We are going to make some changes to the interface of this page.

First of all, let's take a look at the "Address" control. This control must display several lines on the screen. We are going to enlarge it.

Then, we are going to align the controls in the page.

Finally, we will modify the style of "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 input area and enlarge the area.
- ▶ This allows you to increase the width of different page controls.

- ▶ To align the controls in the page:

1. Select the "Last name" control then all edit controls found in the window (hold the Ctrl key down while clicking the different controls).

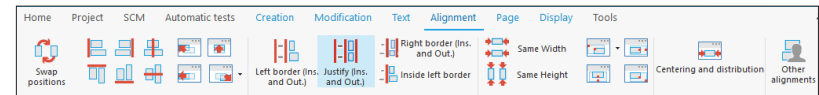


Remark

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

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

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



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




Remark

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

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

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

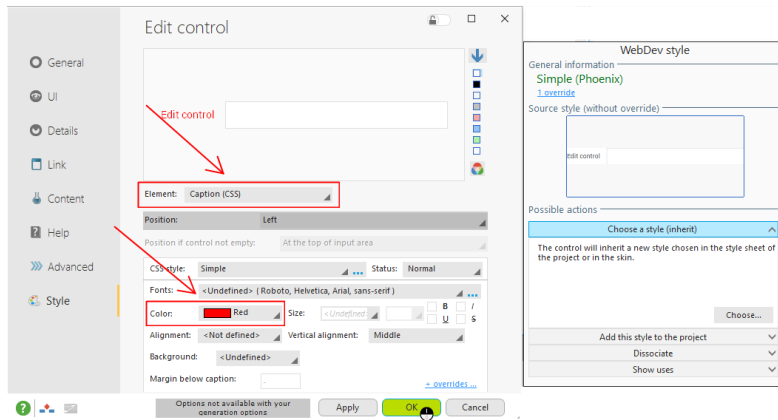
4. Save the page: click  among the quick access buttons (or press Ctrl + S).

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

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

► To modify the default style:

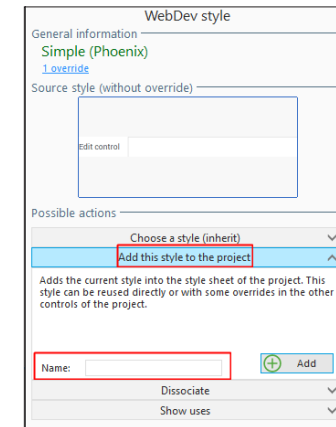
1. Select the "Login" control.
2. Open the control description window (double-click the control for example).
3. In the "Style" tab, select the "Caption (CSS)" element.
4. Expand the "Color" combo box and select the red color.



Remark

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

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



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

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

Remark: Choosing an existing style will be performed in lesson 5.

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

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

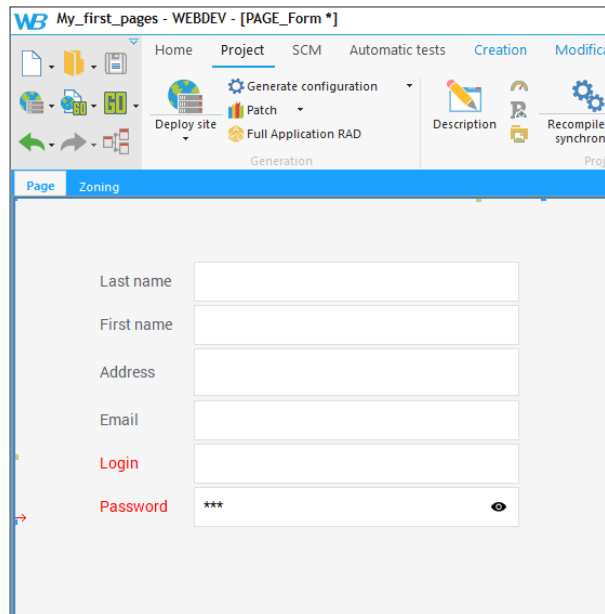
1. Select the "Password" control.
2. Press F4.




Remark

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

3. You get the following interface:



Page test

- ▶ Run the test of this page (click the  icon among the quick access buttons). You have the ability to type 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 typed. This data can be saved in a text file, in an XML file or even in a database.

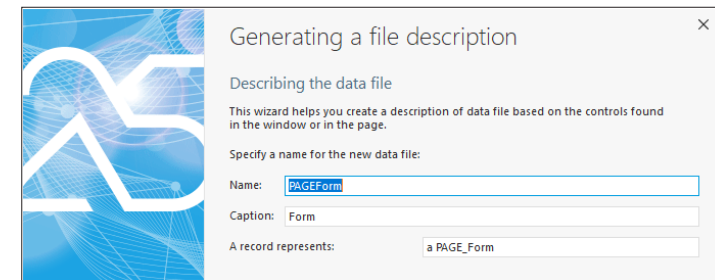
We have chosen the last option.

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

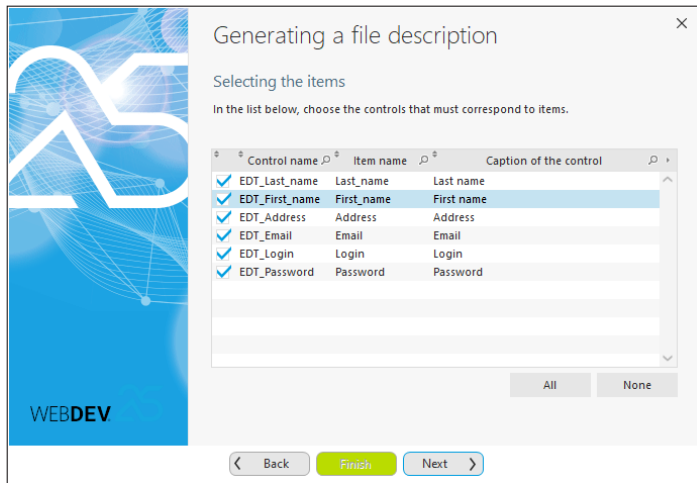
Creating the data file

- ▶ 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 data files that will be used by the project.
 3. The data file generation wizard starts.



4. The name of the data file is automatically filled with the page name. 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 ("Next" button at the bottom of the wizard).
6. Choose the controls to retrieve. In our case, all page controls correspond to an item.



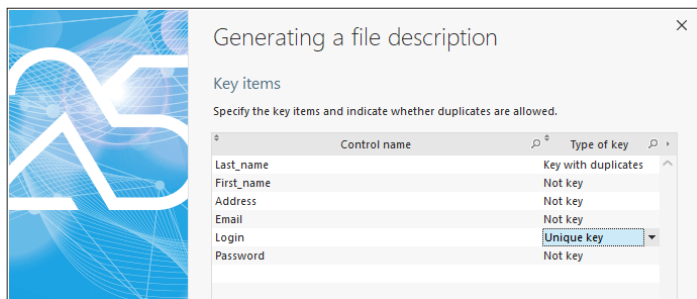
7. Go to the next step.
8. This step is used to define the keys (also called indexes).



Remark

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 "Last_name" item will be a key with duplicates and the "Login" item will be a unique key. Therefore, the data 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 description appears in the data model editor.
12. Save the analysis: click among the quick access buttons (or press Ctrl + S).
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 data

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



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

- ▶ We need a Button control to save the data from our form. This control will be used to validate the information entered in the page and to save the data in the Registration data file.
- ▶ To create a Button control:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. The control currently created follows the mouse movement.
 3. Click inside the page to create the control (below the edit controls for example). The control is created.
 4. Press the space bar: the caption becomes editable.
 5. Type "Save".
 6. Press Enter to validate.
- ▶ To write the WLanguage code associated with the Button control:
 1. Display the popup menu of the control and select "Code" (or press F2).
 2. The events associated with the control are displayed. Yellow headers correspond to the events performed on the server, and green headers correspond to the events performed on the browser.
 3. We want to save data in the database. The access to the database can be performed in Server code only: we must write the WLanguage code in the "Server click" event (yellow).
 4. Write the following WLanguage code:

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


5. Let's study this code:

- **PageToFile** is used to retrieve the content of the edit controls 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 (cross in the top right corner).

Page test

► Run the test of this page:

1. Click the  icon among the quick access buttons.
2. The page is displayed in a browser.
3. Type the following data:
 - Last name: Doe
 - First name: John
 - Address: Flower Rd, 75000 Paris
 - Email: john.doe@atandt.com
 - Login: john
 - Password: john
4. Click on "SAVE". The browser displays a message to confirm that the record has been added!
5. Close the browser.

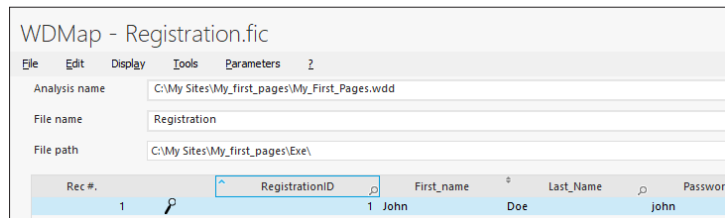
Viewing the data typed

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

This tool is named WDMaP. We are going to use it in order to check whether the information typed was saved.

► To start WDMaP:

1. In the ribbon, on the "Tools" pane, in the "Database" group, click "WDMaP".
2. Select the "Registration" data file. The data file content is displayed.




3. We can find the data typed.
4. Close WDMaP ("Close" button).

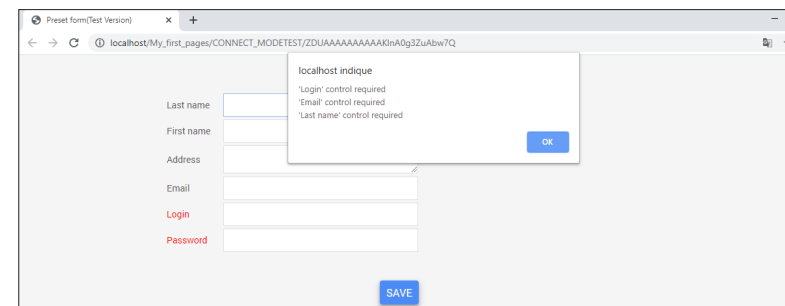
Adding input checks

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

- force the Web user to type the last 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 "Last name" control:
 1. Double-click the "Last name" control: the description window of control is displayed.
 2. Display the "Details" pane.
 3. Check "Mandatory input". If this option is checked, WEBDEV will automatically check whether this edit control was filled.
 4. Validate the description window.
- To apply this modification to the "Email" control:
 1. Select the "Email" control (mouse click).
 2. Press F4: the last action performed on a control is rerun on the selected control.
- Similarly, apply this modification to the "Login" control.
- Run the test of this page:
 1. Click  among the quick access buttons.
 2. In the window that appears, click "SAVE".
 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 will write the WLanguage code required for the check.


- ▶ Create a new edit control. The caption of this control is "Check" and its type is "Password".
- ▶ The WLanguage code for checking the password must be written in the events associated with the "SAVE" Button control.
 1. Click the "SAVE" Button control and press F2.
 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. We are going to type the following code in the browser "Click" event:

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

4. Let's study this code:

- The IF statement is used to perform an action on a condition. In our case, the action is performed if the "Password" and "Check" controls are different ('<>' operator).
- In case of difference, the edit controls are "cleared": an empty string is assigned to them.
- **ReturnToCapture** is used to position the cursor in the specified control (the "Password" control here) without running the code that follows. In our case, if the passwords entered are different, the "Password" control takes the focus and the code being run 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 the  icon among the quick access buttons.
2. Enter the following information in the page that appears in a browser:
 - Last name: "Doe"
 - First name: "Peter"
 - Email: "Peter.Doe@gmail.com"
 - Login: "Peter"
 - Password: "Peter"
 - Check: "Louis"
3. Click "SAVE".
4. An error message is automatically displayed to indicate that the passwords are different.
5. Validate this message.
6. Close the browser.

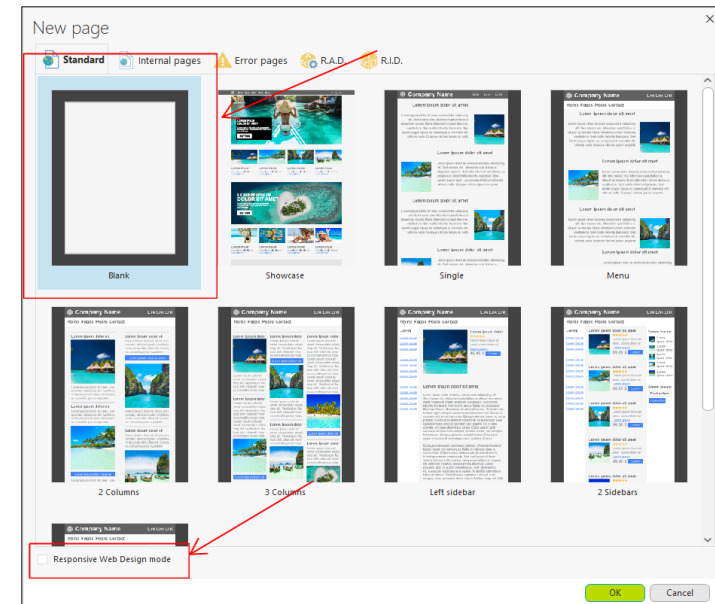
Simple search in a database

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

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

Creating the page

- ▶ To create the page:
 1. Click  among the quick access buttons.
 2. The window for creating a new element is displayed.
 3. Click "Page" then "Page". The page creation wizard starts.
 4. Select "Blank".



5. Make sure that "Responsive Web Design mode" is unchecked.
6. Validate.
7. The page is created and the save window is automatically displayed.
8. Type the page title: "Login".
9. Validate.


Creating controls

The login page will contain the following controls:

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

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

► To create an edit control:


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



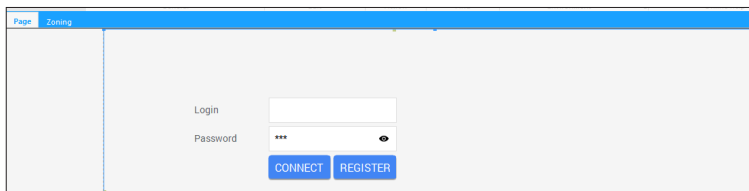
Remark

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

► To create a Button control:

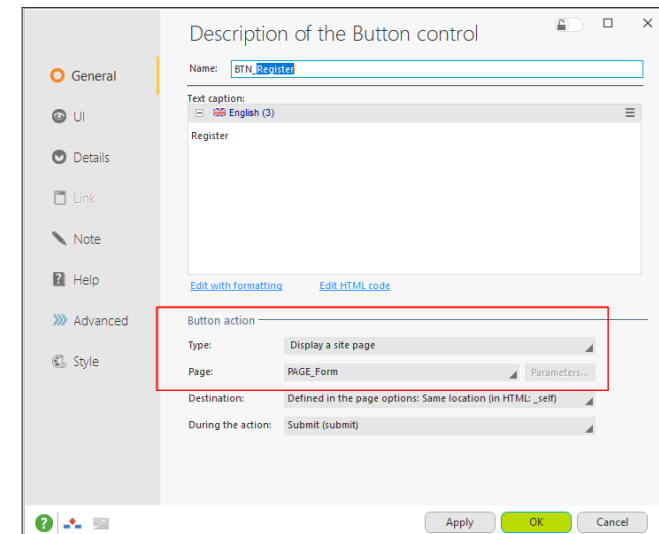
1. On the "Creation" pane, in the "Usual controls" group, click .
2. Click inside the page to create the Button control (below the edit controls for example). The control is created.
3. Press the space bar: the caption becomes editable.
4. Type the caption ("Connect" and "Register") and press Enter to validate. The caption is immediately modified.

► You get the following page:



► We will now see in detail how Button controls work. First of all, let's look at the "REGISTER" Button control.

1. Select the "REGISTER" Button control.
2. Open the control description window (double-click the control).
3. In the "General" tab, in the "Button action" area, we are going to specify the action to perform. In our case, we are going to open the registration page:
 - In the "Type" area, select "Display a site page".
 - In the "Page" area, select "PAGE_Form".



4. Validate the description window.



Remark

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

► For the "CONNECT" Button control, we must write the WLanguage code to perform the search:

1. Select the "CONNECT" Button control and display the associated events (F2).
2. Write the following WLanguage code in the "Click (server)" event:

```
HReadSeekFirst(Registration, Login, EDT_Login)
IF NOT HFound(Registration) THEN
    Error("Unknown login.")
ELSE
    IF EDT_Password = Registration.Password THEN
        Info("OK, you are connected.")
    ELSE
        Error("Incorrect password.")
    END
END
END
```

3. Let's take a look at this WLanguage code:

- This code is run in server code because a search is performed in the database that is located 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 data 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") as well as the action to perform if the login is found.
- If the login is found, the password typed in the EDT_Password control is compared to the one found in the data file (for the specified login). If the two passwords are the same, then the user is connected, otherwise an error message is displayed.



Remark


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

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

4. Save (Ctrl + S) and close the code editor (cross in the top right corner).

Page test

► Run the test of this page:

1. Click the  icon among the quick access buttons.
2. Enter the following data in the page that appears in a browser:
 - 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 WEBDEV features:

- the creation of controls: edit controls and Button controls,
- the possibilities for aligning and organizing 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 WLanguage server and browser events,
- ...

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

PART 2

My first
database



LESSON 2.1. OVERVIEW

This lesson will teach you the following concepts...

- Overview of the project developed in this part.

 Estimated time: 5 mn

Overview of the project developed in this part

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

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

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

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

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

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

LESSON 2.2. WEBDEV AND THE DATABASES

This lesson will teach you the following concepts...

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



Estimated time: 5 mn

Overview

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

In WEBDEV, when creating a project that is using data, 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.



Remark

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

HFSQL

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

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

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

HFSQL can be freely distributed with your WEBDEV sites.

HFSQL proposes all database features, especially:

- the log process,
- the transactions,
- the replication,
- the triggers.

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

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

The different modes for accessing the databases

Native Connector (Native Access)

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

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

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

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

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

Direct ODBC access

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

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

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

OLE DB access

An OLE DB access is an access that uses 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. If you want to use this type of access, check whether an OLE DB driver exists.

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

ODBC access via OLE DB

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

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

LESSON 2.3. PROJECT AND ANALYSIS

This lesson will teach you the following concepts...

- Creating a project.
- Creating an analysis.



Estimated time: 40 mn

Overview

To create a site with database, you must:

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

Creating the project

► To create the project:

1. Start WEBDEV (if not already done).
2. Display the WEBDEV home page if necessary (Ctrl + <).
3. In the home page, click the "Create a project" button and select "Site". The project creation wizard starts. The different steps of the wizard help you create your project. The information specified in this wizard can be modified later.

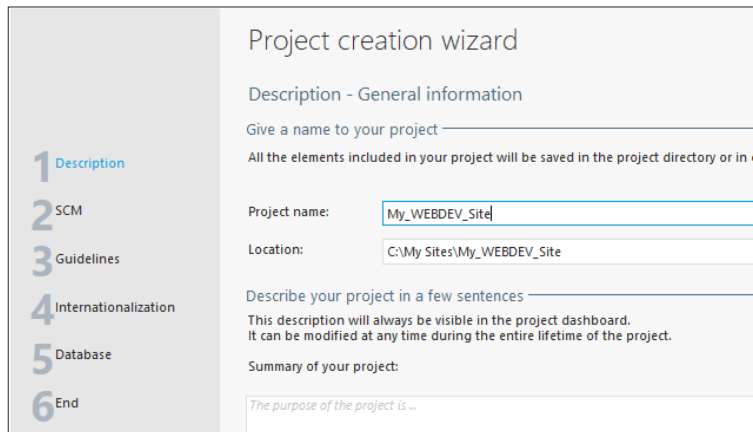


Remark

Tip: To create a project, you can also:

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

4. The wizard proposes to create a blank project or a project based on an example. Select "Create a blank project" and go to the next step ("Next").
5. Specify the type of site to create. Select "WEBDEV site" and go to the next step.
6. The wizard proposes two generation modes. Select "Session mode" and go the next step.
7. The wizard allows you to enter the name of the project, its location and description. In our case, this project will be named "My_WEBDEV_Site".



8. 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.



Remark

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

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

9. The different wizard sections are indicated in the left margin. You can easily go from a section to another one by clicking on it. Since the other steps of the "Description" are not strictly necessary, you can click "Guidelines" directly.

10. This section is used to define the different guidelines associated with the project. For the code style, don't modify the suggested options. Go to the next step.

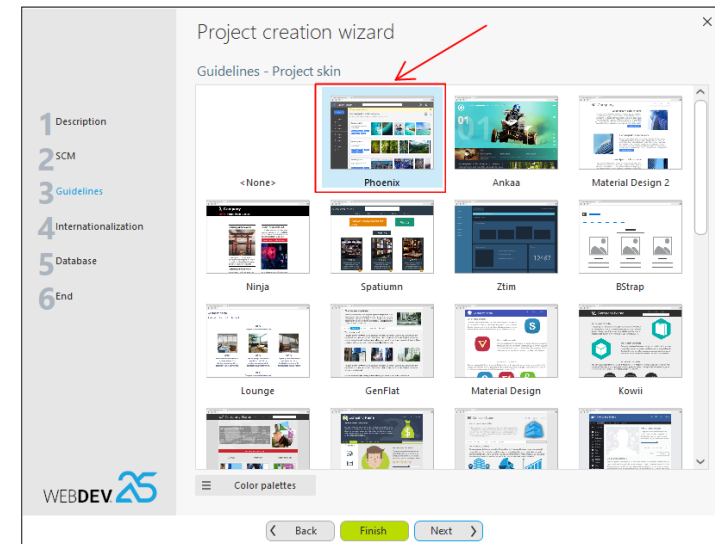
11. This step is used to define the style book (also called "Skin"). The style book is used to define the site style.



Remark

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 "Phoenix".



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

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



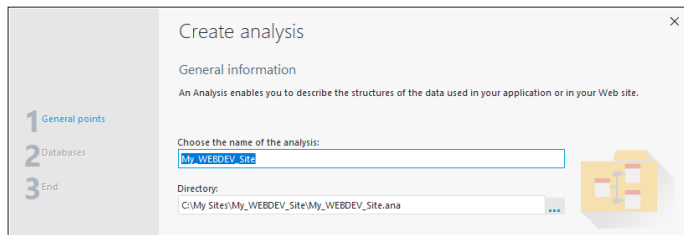
Remark

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 operations performed. To open this corrected project, in the WEBDEV home page (Ctrl + <), click "Tutorial" and select "My WEBDEV site (Answer)".

Creating the analysis

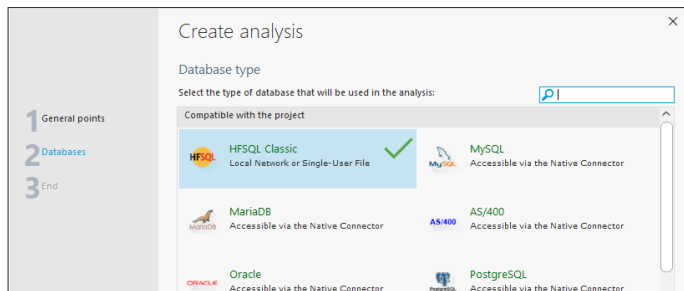
► The steps of the analysis creation wizard are as follows:

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



Go to the next step of the wizard.

2. You can then choose the type of database used by the project. Select "HFSQL Classic" (WEBDEV's default database).



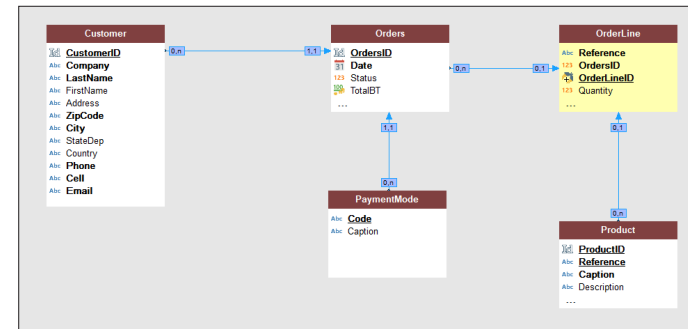
Go to the next step of the wizard.

3. Validate. The data file creation wizard starts automatically.

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.

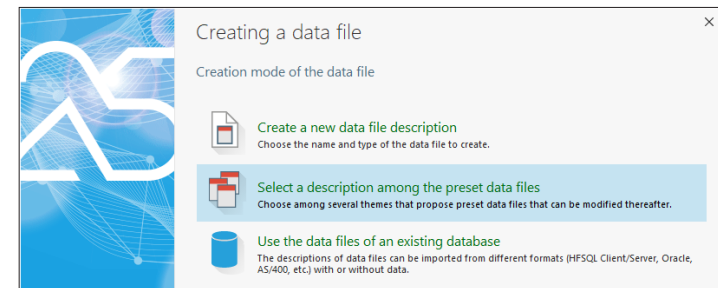


To create the data files of this analysis, we will be using the different methods proposed by WEBDEV.

Creating a data file: using a preset data file

► The steps of the data file creation wizard are:

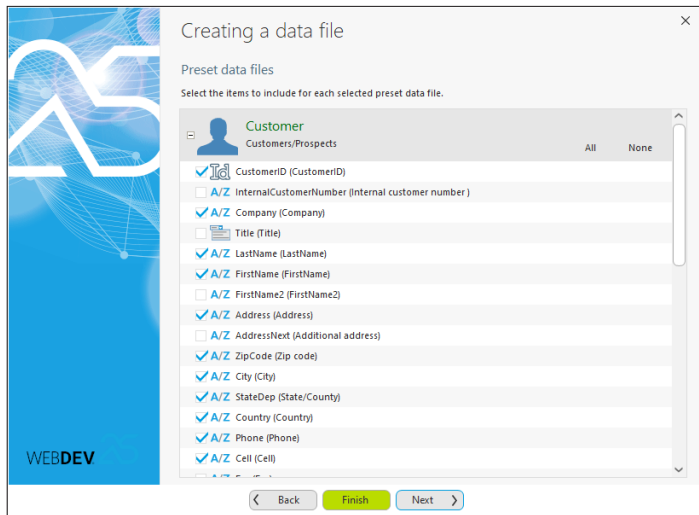
1. In the wizard, select "Select a description among the preset data files".



Go to the next step of the wizard.

2. The list of preset data files is displayed. We are going to create the "Customer" data file. In the list of data files, select "Customer". Go to the next step.

3. The wizard proposes the list of items that will be included in Customer data file. This list is very impressive because it allows you to manage several types of Customer data files.

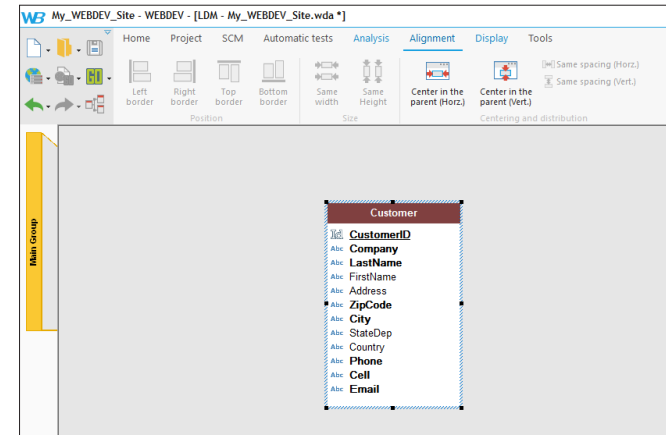


4. Click "None" to deselect all items. Then, check the following items:

- CustomerID,
- Organization,
- Name,
- FirstName,
- Address,
- ZipCode,
- City,
- StateDep,
- Countries,
- PhoneNumber,
- Mobile,
- Email.

5. Go to the next step of the wizard.

6. Validate the wizard. The "Customer" data 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 data file and items

► To create a data file from the window for creating a new element:

1. Click "Data" then "Data file".

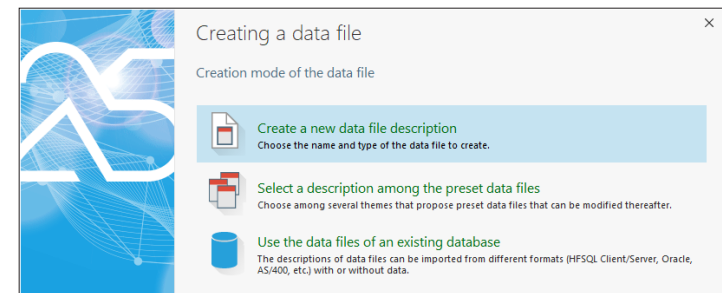


Remark

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

2. The data file creation wizard starts.

3. In the wizard, select "Create a new data file description".



Go to the next step of the wizard.

4. We are going to create the "Orders" data file. Type its name ("Orders") in the wizard. This name will be used:

- to handle the data file by programming. The variable associated with the data file will be "Orders".
- to build the name of associated physical data file ("Orders.fic" file).

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



Remark

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

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

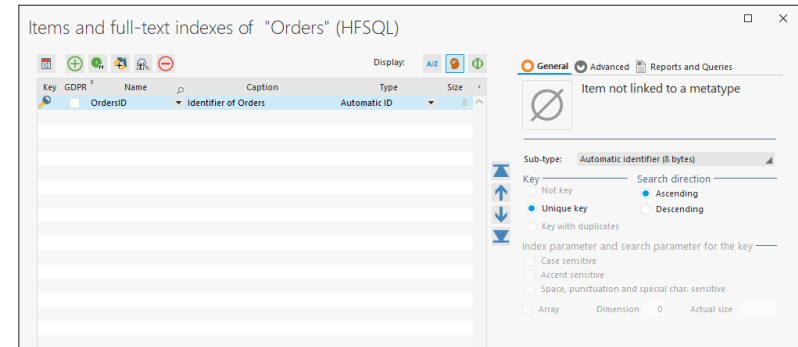


Remark

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

6. Go to the next step and select the type of database associated with the data file. We are going to work on HFSQL Classic data files. Go to the next step.

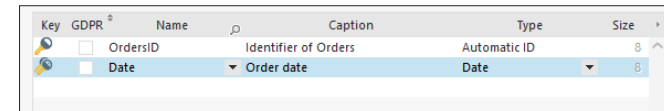
7. Click "Done" to validate. The data file is automatically created in the analysis. The description window of items is opened.



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

► First, we are going to create the "Date" item. This item will contain the order date.

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



3. This item will be a key item (index) in our data file: the keys are used to accelerate the accesses to data and the sorts.



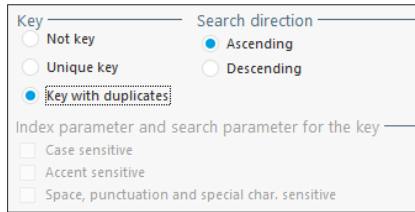
Remark

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

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

- 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).

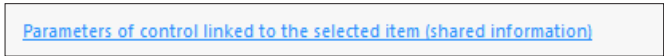
4. The key definition is performed as follows: re-select the line of "Date" item to enable the description controls found on the right of screen. Then, simply specify the type of key used. In our case, the date is a key with duplicates.



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".

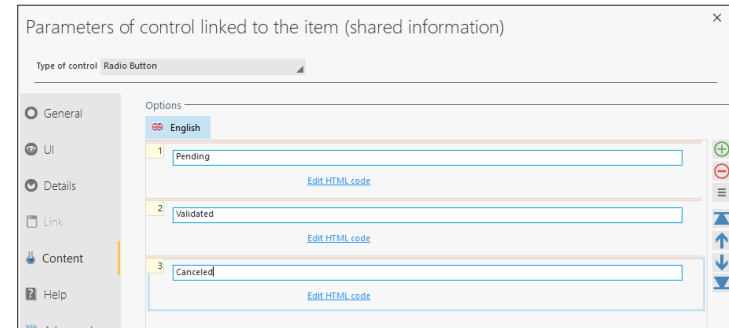
► We are now going to create the "Status" item that is used to find out the order status.

1. Position the cursor on a new row in the array of items. Enter:
 - the name: Status
 - the caption: Order status
 - the type: Radio Button, List Box, Combo Box. In the window that appears, you can 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 link to display the parameters of the control linked to the selected item.

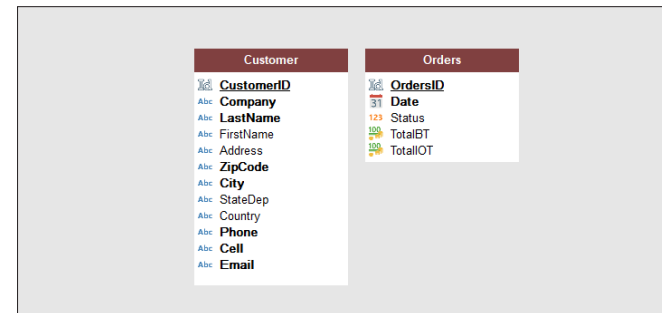


The information typed in the new window that is displayed will be automatically used when creating pages linked to the data file. You will find here the control type and the caption. We are going to type the different options corresponding to the order status in the "Content" tab:

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



3. Validate the parameter window ("OK").
4. Similarly:
 - Position the cursor on a new row and create the item "Total BT". This item is a "Currency" item.
 - Position the cursor on a new row and create the item "Total IOT". This item is a "Currency" item.
5. That's it, the description of "Orders" data file is complete. Validate the description window of items.
6. The "Orders" data file appears in the data model editor.



Importing a CSV file

A different method will be used to create the "PaymentMode" data file, which contains the payment characteristics: importing a CSV file.



Remark

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 semicolon or a tab in most cases).

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

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

► To import a CSV file into the analysis:

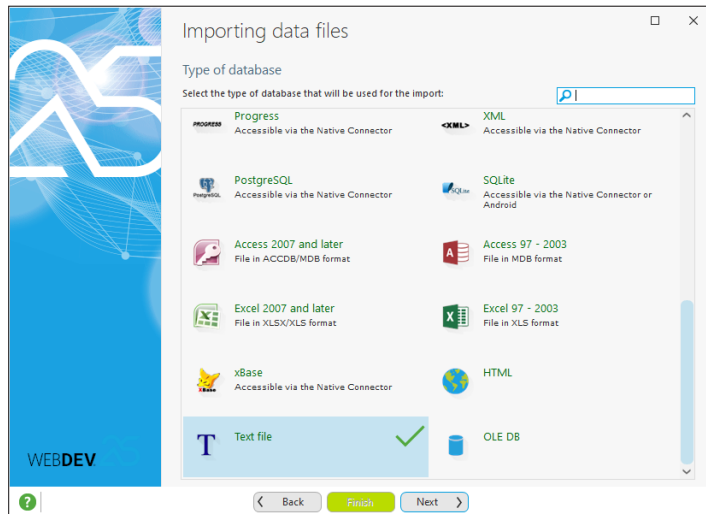
1. In the ribbon, in the "Analysis" pane, in the "Creation" group, expand "Import" and select "Import the descriptions of files/tables".



Tip

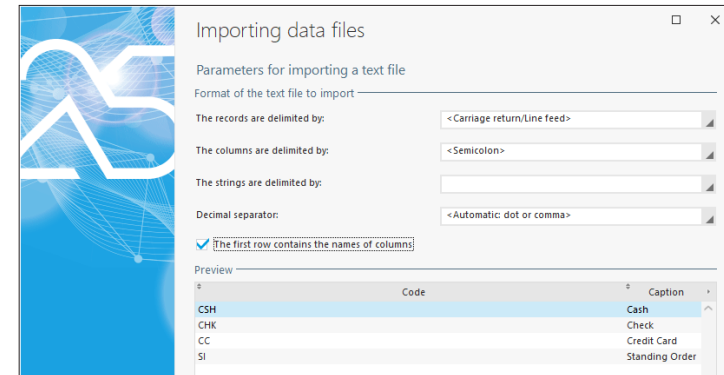
To import a CSV file (or any other file) into the analysis, you also have the ability to Drag and Drop the CSV file (from the Windows explorer) to the data model editor. This will be presented in the next paragraph.

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

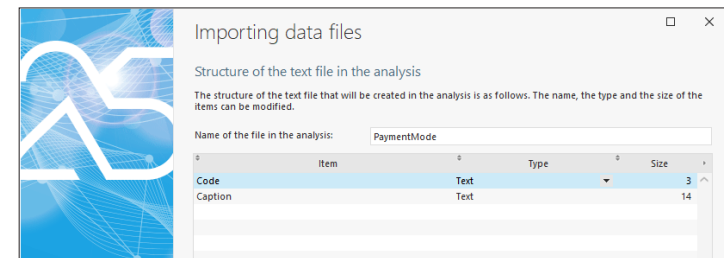


Go to the next step of the wizard.

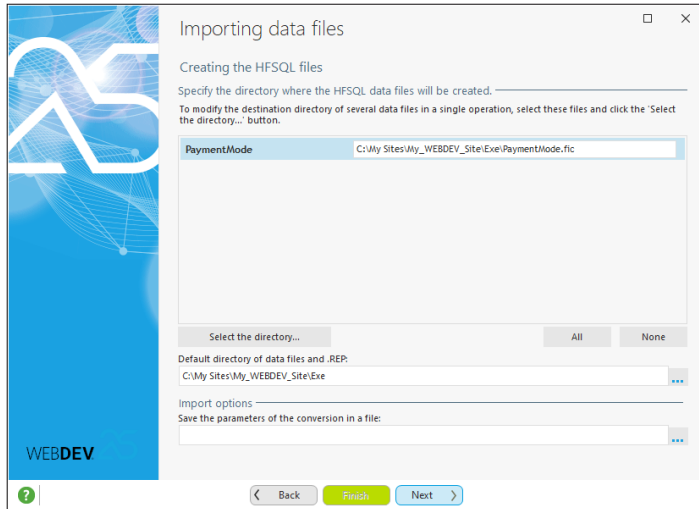
5. Specify the path of file to import: "\Tutorial\WB\Exercises\My_WEBDEV_Site\PaymentMode.csv" found in the WEBDEV setup directory.
6. Go to the next step of the wizard.
7. Specify the following import parameters:
 - Records delimited by: "<Carriage Return/Line Feed>"
 - Columns delimited by: "<Semicolon>"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



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



11. The content of CSV file will be automatically converted to HFSQL format. The wizard proposes to create the HFSQL data file in the project directory.

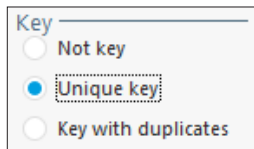


Keep the proposed options and proceed to the next step.

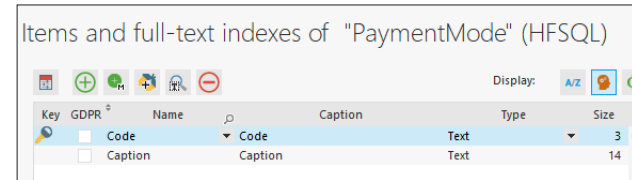
12. Validate the wizard. WEBDEV creates the data file.

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

1. Select the "PaymentMode" data file then select "Description of data file" in the popup menu.
2. Change the data file caption in the window that opens up: delete "(Imported)".
3. Click the icon to display the description of the data file items.
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:



6. Validate the item description window and the data file description window.

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" data file.

► To import HFSQL data files:

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

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



Caution!

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

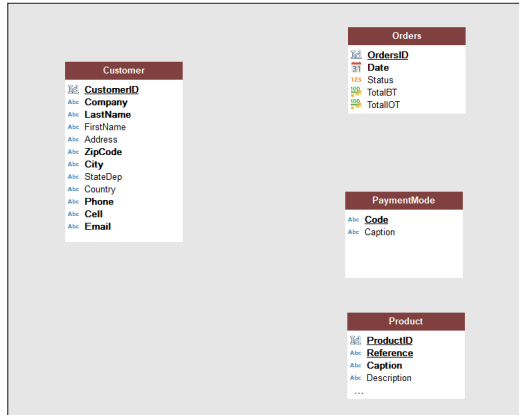
To handle the data in the data file has just been imported:

1. Open the Windows file explorer.
2. Select the following WEBDEV subdirectory: "\Tutorial\WB\Exercises\My_WEBDEV_Site".
3. Copy "Product.fic", "Product.mmo" and "Product.ndx" to the EXE subdirectory of your project directory.

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

Creating the links

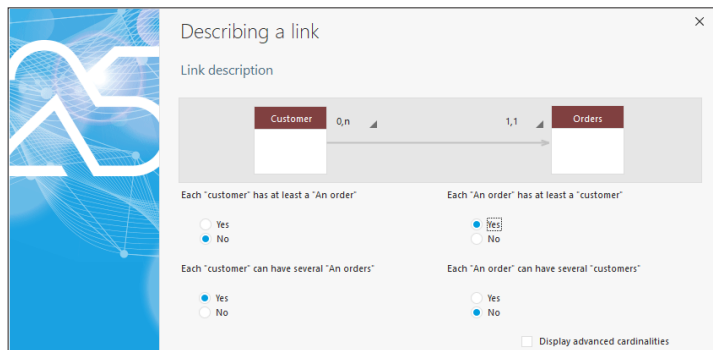
We have created all the descriptions of the data files necessary for the product management application.



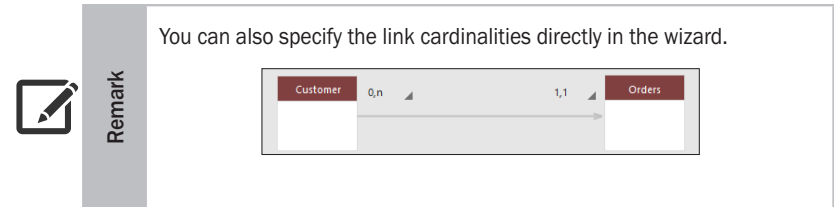
We are now going to create the links between data files. A link is used to define the integrity constraints (cardinalities) between two data files.

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

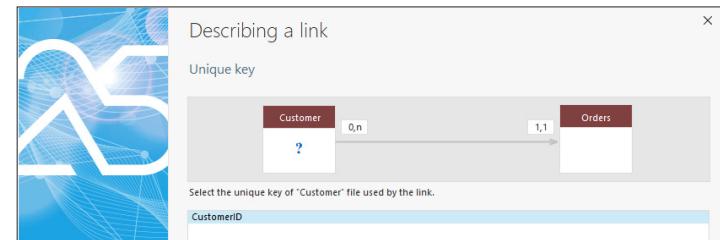
1. On the "Analysis" pane (found in the ribbon), in the "Creation" group, click "New link". The mouse cursor turns into a pen.
2. Click the "Customer" data file, then the "Orders" data file.
3. The wizard for link creation starts.
4. Answer the questions asked by the wizard:



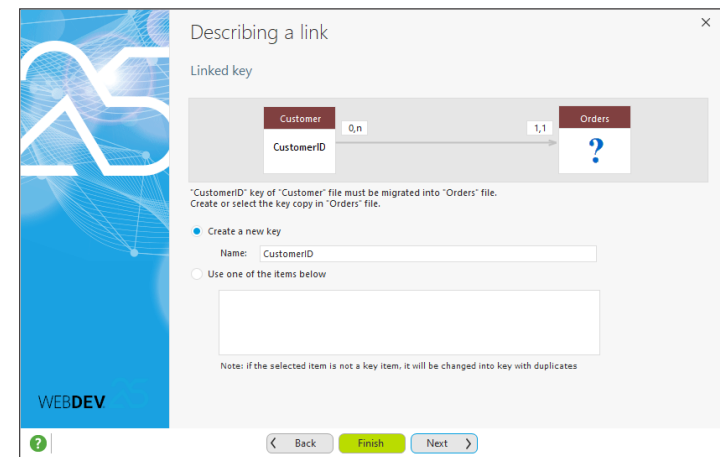
- Each Customer has at least one Order: No
- Each Customer can have several Orders: Yes
- Each Order has at least one Customer: Yes
- Each Order can have several Customers: No



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



6. Display the next step of the wizard. The wizard offers to create the "CustomerID" key in the Orders data file to store the customer corresponding to the order.

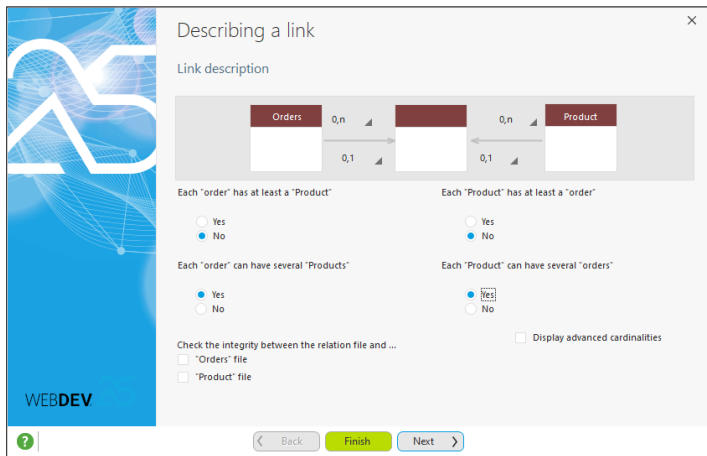


7. Accept this option by going to the next step.

8. This step defines 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 step of the wizard.
10. Click "Finish". The link is automatically created in the data model editor.

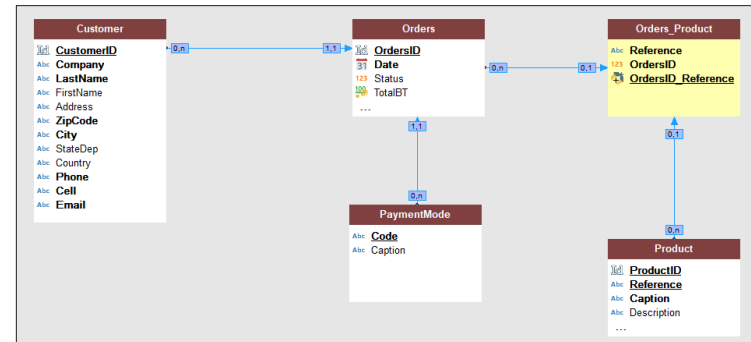
- ▶ Likewise, create a link between the "PaymentMode" and "Orders" data files. These two data files are linked as follows:
 - An order must have a payment mode.
 - A payment mode can be used in several orders.
 In the wizard:
 - The cardinalities are as follows: PaymentMode (0,n), Orders (1,1).
 - The link key corresponds to the "Code" item.

- ▶ We are now going to create a link between the "Orders" and "Product" data files. This link will be used to create a link file, the data file of order lines.
 1. Likewise, create the link between the "Order" and "Product" data files.
 2. Answer the questions asked by the wizard:
 - Each Order has at least one Product: No
 - Each Order can have several Products: Yes
 - Each Product has at least one Order: No
 - Each Product can have several Orders: Yes



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

4. The wizard proposes to use the unique key of the Orders data file: "OrdersID". Go to the next step.
5. Validate the key creation 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 data file. Select "Reference". Go to the next step.
8. Validate the key creation by going to the next step.
9. Keep the default options regarding the integrity rules and go to the next step.
10. Click "Finish". 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. This file will contain the order lines.

We are going to:

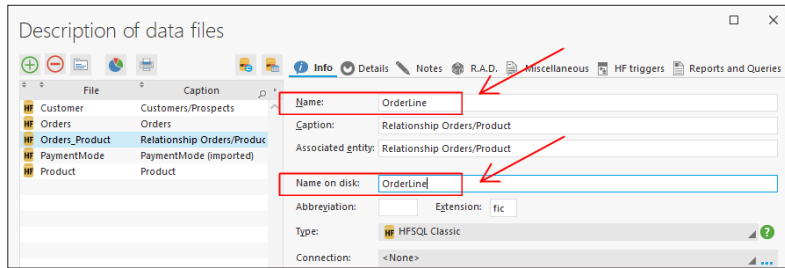
- Modify the name of the relation file.
- Modify the name of its items.
- Add items to find out the quantity of products ordered and the total of order line.

▶ First, we are going to rename the relation file. We already did something similar when we changed the caption of the imported data file. However, in this case, not only the caption will be modified: we will also rename the physical file linked to the data file description.

1. Select the "Orders_Product" data file. Select "Description of data file" from the popup menu.

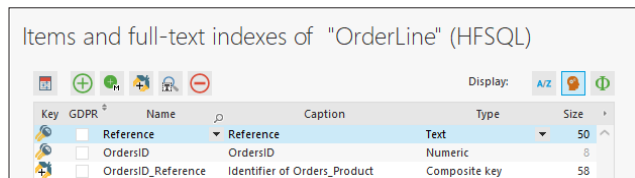
2. In the window that is displayed, modify:

- the name of data file: "OrderLine".
- the name on disk: "OrderLine".



► We are now going to modify the items of this relation file.

1. Click the icon to display the description of file items.



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



Remark

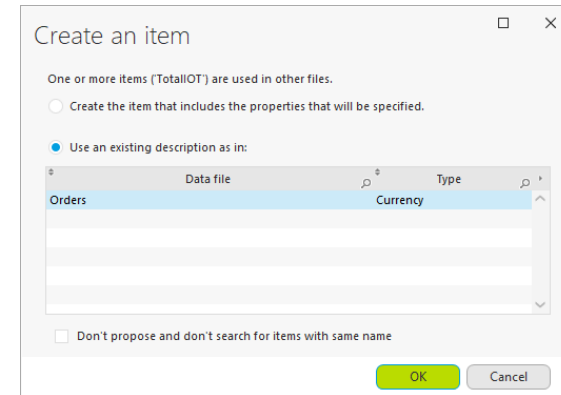
A composite key is a set of items that constitute an index. This type of key is used to browse the data file according to complex criteria or to perform specific searches on several items at the same time.

3. To rename this item:

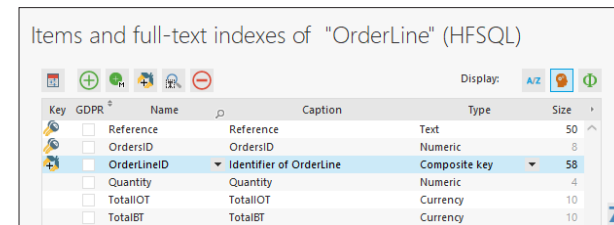
- Click the "Name" column.
- Replace "OrdersID_Reference" by "OrderLineID".
- Click the "Caption" column.
- Replace the current caption by "Identifier of OrderLine".

► We are now going to add 3 new items into this relation file: Quantity, TotalIOT and TotalBT.

1. Position the cursor on a new row and create the item "Quantity". This item is a "Numeric" item.
2. Position the cursor on a new table row and create the item "TotalIOT". 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:

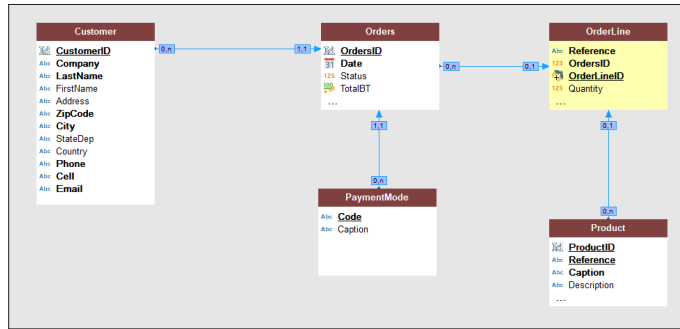


4. Keep the options selected by default and validate ("OK").
5. Position the cursor on a new table row and create the item "TotalBT". This item is a "Currency" item. Once again, use the existing description.
6. The description of the "OrderLine" data file items is as follows.



7. Validate the item description and the data file description (click "OK").

► The analysis is as follows:



Generating the analysis

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

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 ribbon of the data model editor, on the "Analysis" pane, in the "Analysis" group, click "Generation".
2. The analysis generation is automatically started.

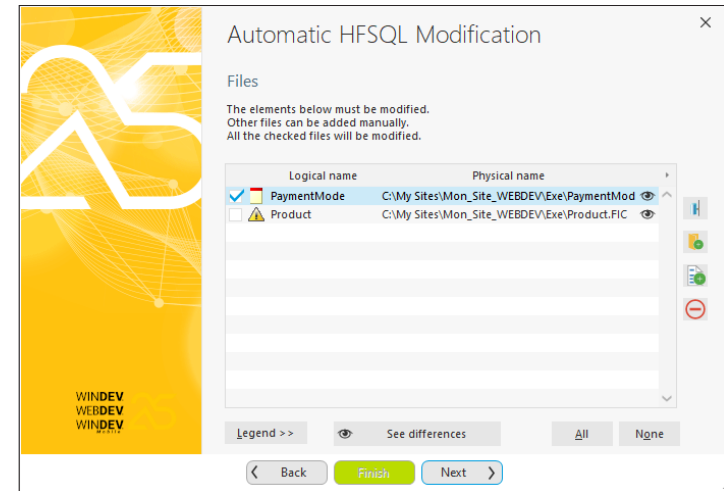
The descriptions of data files found in the analysis have been modified.

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

► Click the "Start" button.

► The wizard for automatic modification starts.

1. Validate the different steps of the wizard until you reach the the list of the data files to take into account:



- The PaymentMode data file must be updated. Keep this file selected.
- The Product data file was copied into the project directory. WEBDEV proposes to associate it with this analysis. Check the data file.

2. Go to the next step.

3. The wizard proposes to save the existing data files, don't change anything and go to the next step.

4. The wizard proposes to type the passwords for protecting the data files modified by the automatic modification. Keep the default options and go to the next step.

5. The list of data files to modify is displayed. End the wizard.

6. The data files are updated.

► Close the data model editor.

The main steps for creating an analysis have been presented.

When your analysis is described and generated, you can:

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

PART 3

Intranet site
with data



LESSON 3.1. OVERVIEW

This lesson will teach you the following concepts...

- Overview of the site created in this part.

 Estimated time: 5 mn

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 in Session mode.**



Remark

SEO

In a WEBDEV site in Session mode, only the site home page can be referenced. The search engine optimization of a WEBDEV site is presented in "Lesson 5.6 SEO", page 249.

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

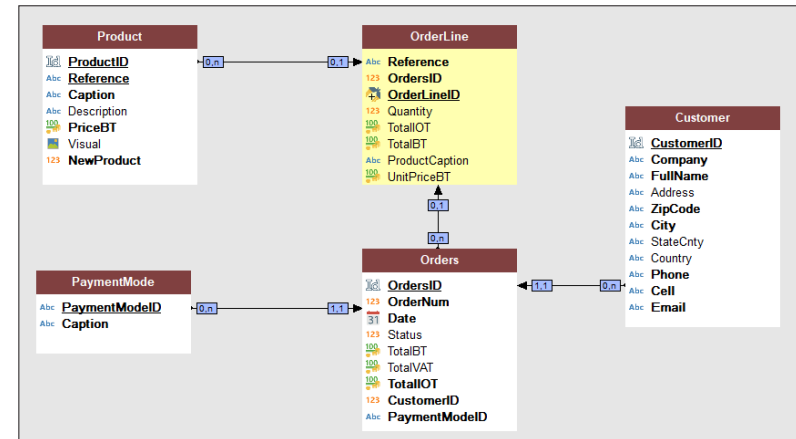
Part 5 will allow you to continue the site development, by proposing the following features:

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

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

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

LESSON 3.2. PRINCIPLE FOR DISPLAYING A WEBDEV SITE IN SESSION MODE

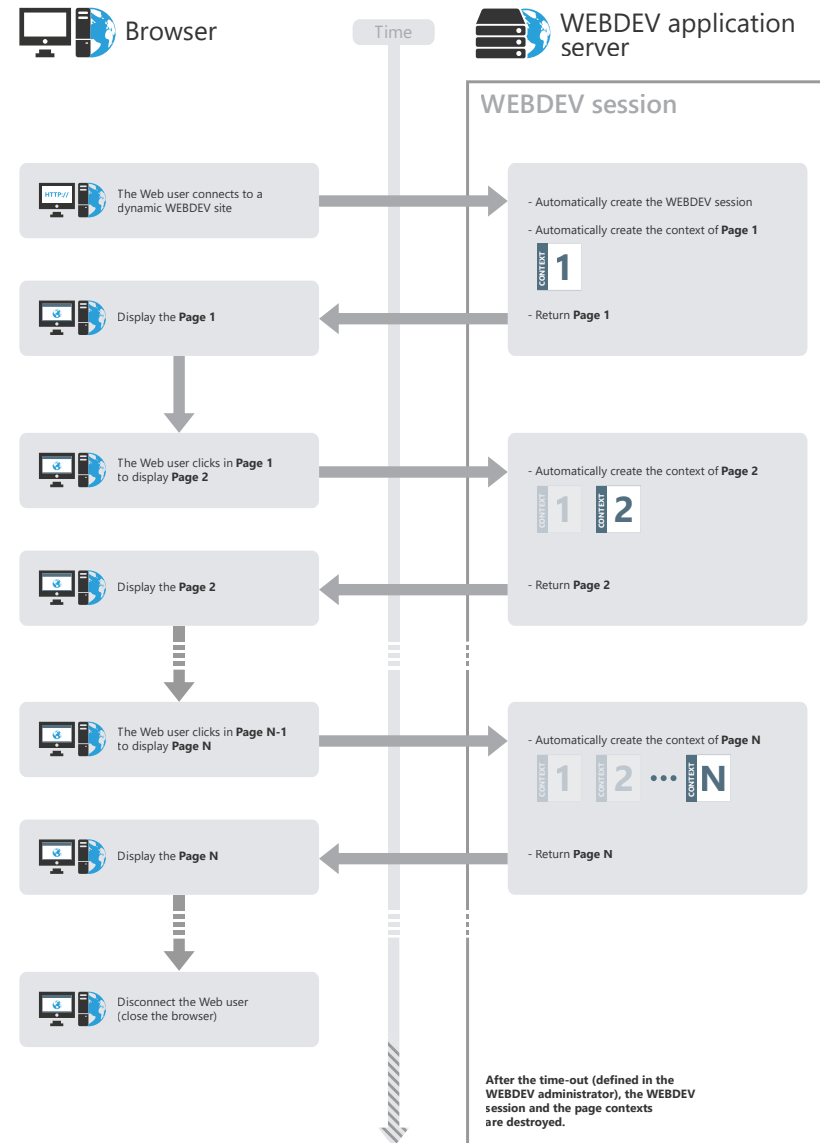
This lesson will teach you the following concepts...

- Page contexts.
- WEBDEV session.

 Estimated time: 5 mn

Principle for displaying a WEBDEV site in Session mode

The principle for displaying a WEBDEV site in Session mode is as follows:



The WEBDEV session

When the Web user connects to a WEBDEV site in Session mode, 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 or when the session timeout is exceeded. This timeout is defined in the WEBDEV administrator ("Configuration" tab, "Disconnect the inactive 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 elements that have been required to build the page viewed by the Web user:

- global variables,
- local variables,
- server processes and events,
- database connections,
- contexts of data files, ...

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 page in Session mode used to list the products.
- Creating a product form page in Session mode.
- Managing the product addition and modification.



Estimated time: 50 mn

Overview

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

- how to use WEBDEV to create pages in Session mode,
- 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. Open the WEBDEV home page (Ctrl + <).
2. In the home page, click "Tutorial" and select "Full WEBDEV Site (Exercise)".



Answer

A corrected project is available. This project contains the different pages created in this lesson. To open the corrected project, in the WEBDEV home page (Ctrl + <), click "Tutorial" and select "Full WEBDEV Site (With pages)".

Creating a page in Session mode used to list the products

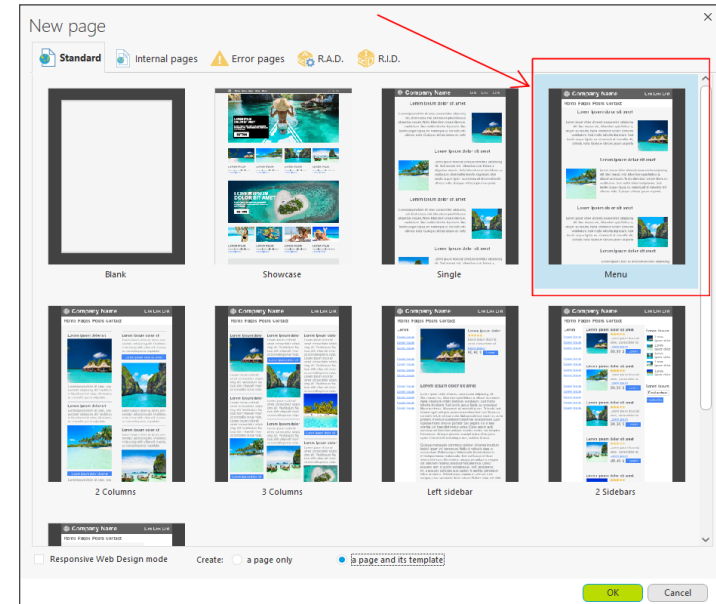
To create the page used to list the products, we are going to create a blank page then add all controls: this will allow us to present the steps required to create this type of page.

Creating a page that is using a template

► To create a blank page:

1. Click among the quick access buttons (or press Ctrl + N).
2. The element creation window appears: click "Page" then "Page".
3. The page creation wizard appears.

4. Select the "Menu" template.



Remark

A preset template is used to group part of the interface and processes in a single location. We recommend using templates to re-use and standardize the interface of your sites.

5. Make sure that option "Create: a page and its template" is checked.

6. Validate the wizard ("OK").

7. The new page is displayed in the editor.

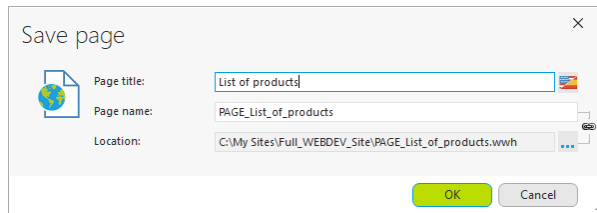


Remark

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.

8. The window for saving the page is displayed. Type the page title: "List of products". The page name "PAGE_List_of_products" is automatically proposed.



9. Validate.

► By default, this page contains the different controls used to display the possible overviews. In our case, we are going to delete these controls:

1. Select the controls found in the page (press Ctrl + A).



Remark

In the editor, by default, only the page controls are accessible. The controls and events of the template cannot be edited. All the controls in the page (including the controls from the template) are selected by pressing Ctrl + A. However, the controls from 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 modifications performed in the page: click  among the quick access buttons (or press Ctrl + S).

Creating controls

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



Remark

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.

Each area can, of course, contain different data.

► To create the Looper control:

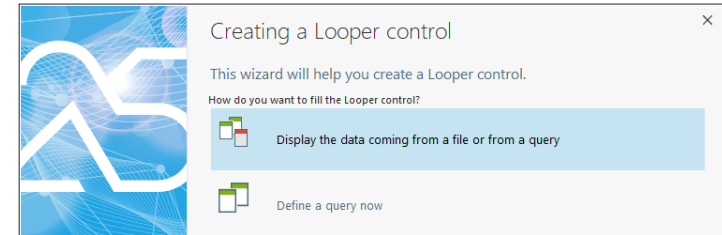
1. On the "Creation" pane, in the "Data" group, expand "Looper" and select "Looper"

2. Click in the top left corner of the page, just below the menu: the wizard for creating the Looper control starts.

3. The wizard wants to know how the Looper control will be filled:

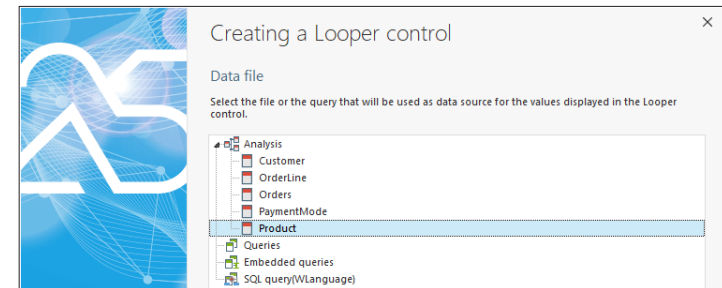
- from the database, from a data file or from a query,
- from a WLanguage variable,
- by programming (this feature will be presented in a next lesson).

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



4. Go to the next step ("Next" in the wizard).

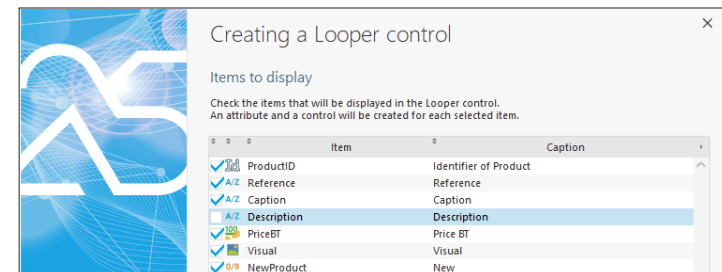
5. The next wizard screen proposes the different data files and queries found in the current project. Expand "Analysis" if necessary and select the "Product" data file.



6. Go to the next step.

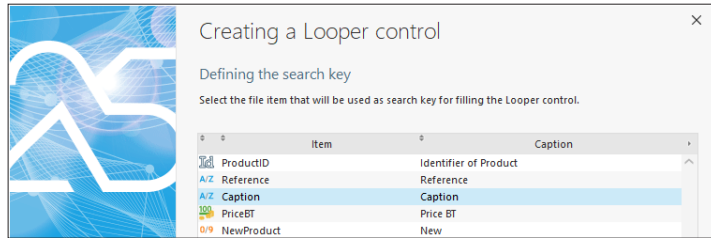
7. The wizard proposes the list of items found in the data file. By default, all items are checked in order to be displayed in the Looper control. In our case, we are going to display all the items EXCEPT for the product description.

Uncheck "Description":



Go to the next step.

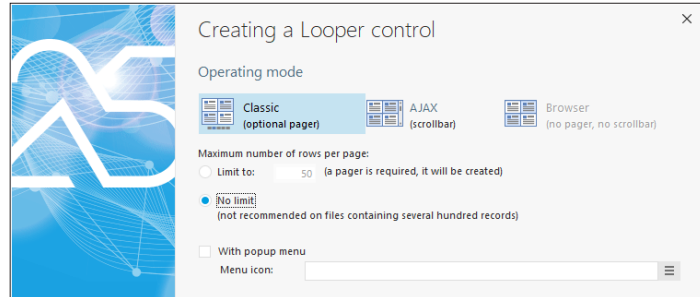
8. Then, the wizard proposes to select the search key, which defines how the data displayed in the Looper control will be sorted by default. The items proposed in the wizard correspond to the items defined as key items in the analysis. The products will be sorted according to their caption.



Select "Caption" and go to the next step.

9. We are now going to select additional parameters for creating the Looper control:

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



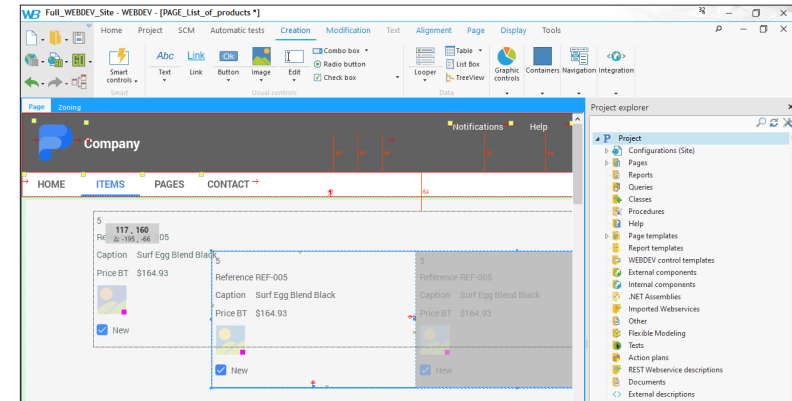
10. Go to the next step. The Looper control will display the products on 3 columns.

- Keep the default option: "Fixed number of rows, Fill in row".
- In the "Number of columns" area, replace 1 by 3.

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

► We are going to reposition the control in the page:

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



3. If necessary, reduce:

- the size of rows (the size of columns) for the control to be displayed in its entirety in the page.
- the size of the controls in the rows so that they fit (column).

Let's look for a moment at the Looper control we have just created: the data is already displayed in the control, even in the editor.

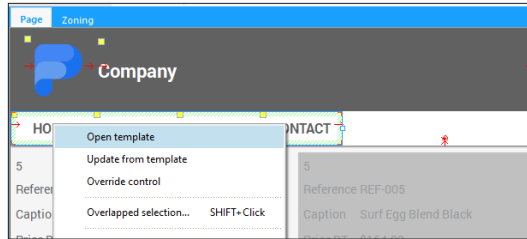
This concept is called "Live data": the content of data files found on the development computer is used in the pages or reports handled in the editor. This feature is very useful to define the size of controls found in a page for example.

► We are going to hide the control containing the product identifier. Why hide it instead of deleting it? Because this identifier will be useful in the rest of our development.

1. Select the "STC_ProductID" control. This control corresponds to the identifier (control with a number at the top of looper).
2. Open the control description window ("Description" from the popup menu).
3. In the "UI" tab, uncheck "Visible".
4. Validate the control description window.

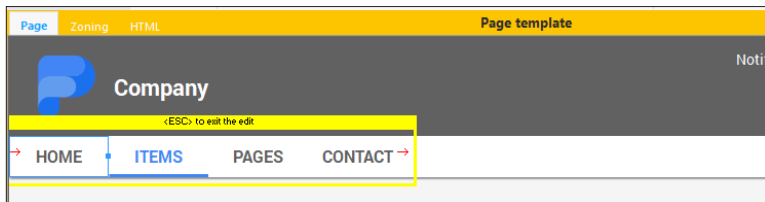
► We are now going to modify the menu of our page in order to display the option used to list the products. The menu is found in the page template. Therefore, we are going to modify the page template.

1. Display the popup menu of the menu (right-click on "HOME" for example) and select "Open the template".



2. Select the menu.

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



4. Open the option description window (select "Option description" from the popup menu).

5. In the "General" tab:

- Type the option caption: replace "Home" by "List of products".
- In the "Action of option" area:
 - Select "Display a site page".
 - Select the "PAGE_List_of_products" page.
- Validate the option description window.

6. If necessary, enlarge the control so that the caption of the option appears on a single line (use the control handles).

7. Press Esc to exit from the edit mode.



Remark

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

► To update the pages that use the template:

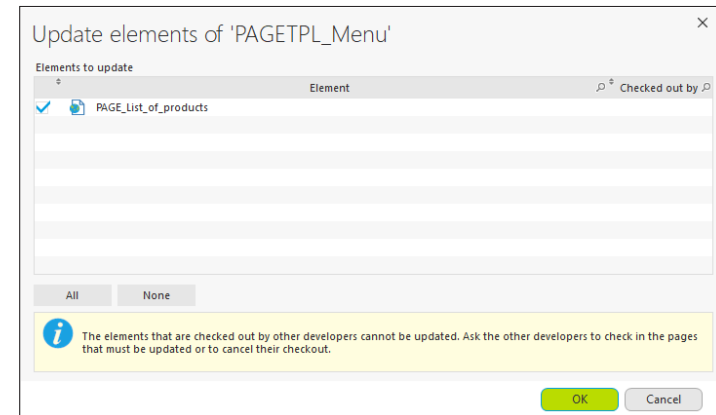
1. In the orange bar of template, click : this button is used to propagate the template modifications to all pages that use the model.



Remark

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 template update window ("OK" button).

4. Close the page template.

► In the "Menu" page template we use, the selected menu option is "ITEMS": this menu option appears in blue. However, the page we have just created displays the list of products. We want the option selected for this page to be "LIST OF PRODUCTS".

To do so, we will override the menu option in the page template to change the selected menu option. To make this change, we will override the menu, then the desired menu option.


► Perform the following actions:

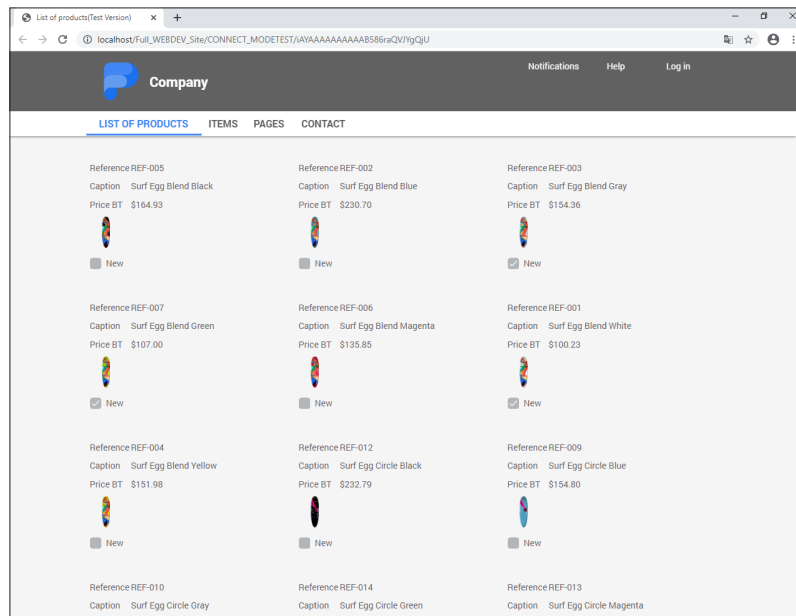
1. Open the popup menu of the menu (right click on "LIST OF PRODUCTS" for example) and select "Override control".
2. Open the popup menu of the menu again and select "Modify this menu". The menu goes into edit mode: a yellow outline appears around the menu.
3. Open the popup menu of "LIST OF PRODUCTS" and select "Override control".
4. Open the popup menu of "LIST OF PRODUCTS" and select "Put this menu option into the selected status".
5. Press Esc to exit edit mode.
6. Save the page (Ctrl + S).

Our page is ready to be checked.

Page test

► We are going to run the test of the page that was just created.

1. Click  among the quick access buttons.
2. The page is automatically displayed in the browser.



3. You can scroll through the products with the scrollbar of the page.


► Close the browser. The WEBDEV editor is redisplayed.

Modifying products via a "Product form" page in Session mode

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 product details:

1. Create a new blank page.
 - Click  among the quick access buttons.
 - The element creation window appears: click "Page" then "Page".
 - The wizard for page creation starts.
 - In the "Based on a project template" area, choose "PAGETPL_Menu" and validate the wizard.
2. The window for saving the page is displayed. Type the page title: "Product form". The page name "PAGE_Product_form" is automatically proposed.
3. Validate.

What should this page do?

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

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

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

► To declare a parameter in the page:

1. Press F2. The code editor displays the different page events.
2. Write the following WLanguage code in the "Global declarations" event:

```
PROCEDURE MyPage (gnProductID is 8-byte int)
```

3. Let's take a look at this WLanguage code:

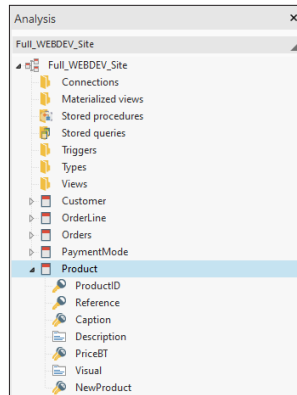
- The PROCEDURE keyword in the "Global declarations" event is used to associate a procedure to the opening of the page.
 - The procedure is named "MyPage". At run time, this keyword will be automatically replaced by the name of current page.
 - The procedure expects the gnProductID variable (that is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

Creating edit controls

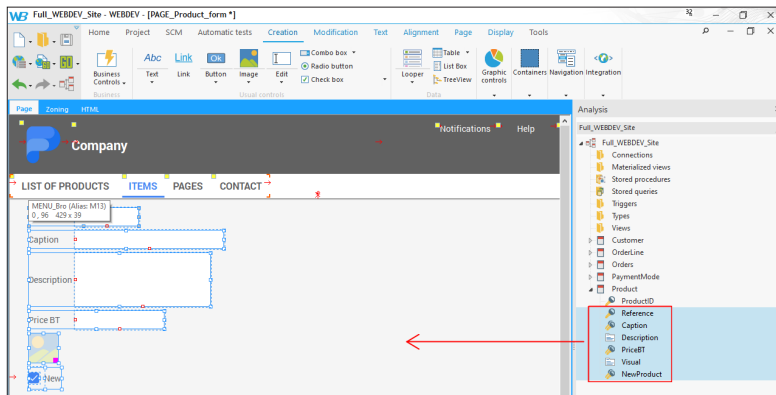
We are now going to create the edit controls used to display information about the selected product.

► To create the controls in the page:

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

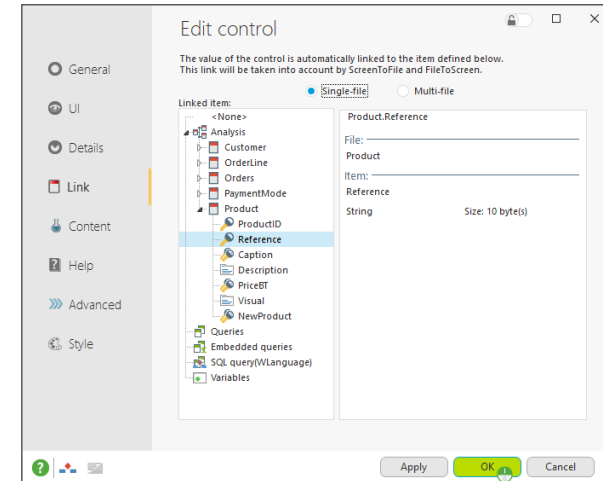


3. With the mouse, select all items displayed in the pane (except for the "ProductID" item). You have the ability to use the multi-selection by holding the Ctrl key down.
4. Drag and Drop these items to the page that was just created.



5. Different controls are automatically created in the page. These controls are linked to the corresponding item in the data file. To check this:

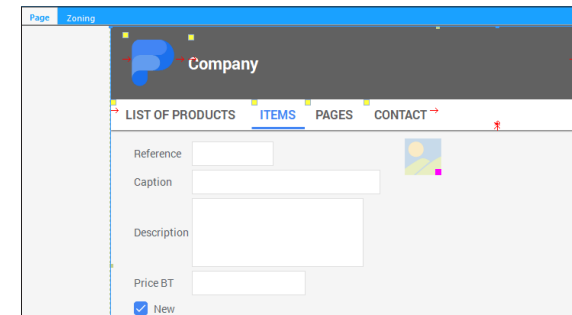
- Select the "Reference" control for example.
- Display the popup menu (right mouse click) and select "Description".
- Go to the "Link" tab in the description window:



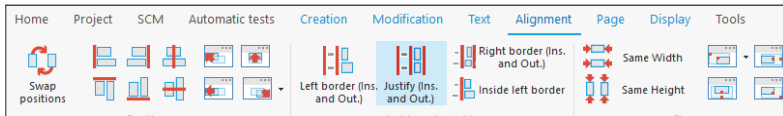
This tab shows that the current edit control is linked to the "Reference" item found in the "Product" data file.

6. Close the description window ("OK" or "Cancel").
7. Save the page (Ctrl + S).

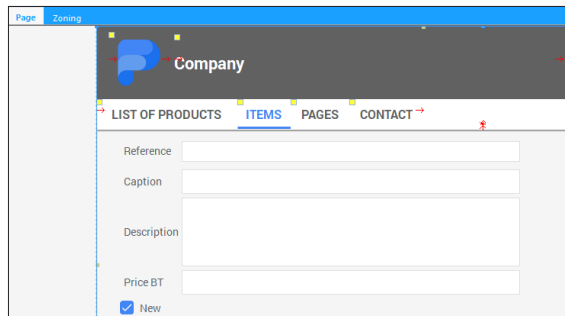
► Reposition the controls in the page in order to get the following interface:



- ▶ Enlarge the Image control used to display 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 "Caption" control then all the edit controls of the page (hold the Ctrl key down while clicking the different controls). The first selected control will be used as reference for the size of other controls.
 2. Display the "Alignment" pane of WEBDEV menu. This pane contains all alignment options available for the controls.



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



4. Save the page.

Displaying data in the page

The "Form" page must display the product that is selected in the Looper control. In the WLanguage events associated with the page, we will enter the code to:

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

- ▶ To search and display data in the page:
 1. Click the page in the editor.
 2. Press F2. The code editor displays the different page events.
 3. In the "Global declarations" event, write the following code below the code previously written:

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

4. Let's take a look at this WLanguage code:

- **HReadSeekFirst** is used to find the first record in the Product data 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**.

5. Close the code editor.

Creating buttons

The "PAGE_Product_form" page will be used to modify a product found in the list of products. We will add two Button controls: a cancellation button and a validation button.


- The "Cancel" Button control will simply display the previous page again.
- The "Validate" Button control will check the entered data and save the modified data.

Cancel button

- ▶ To create the "Cancel" Button control:
 1. On the "Creation" pane, in the "Usual controls" group, click "Button".
 2. Click below the edit controls to create the Button control.
 3. Modify the control caption (press Space to edit the caption): the new caption is "Cancel".

- ▶ Enter the WLanguage code associated with the "Cancel" Button control:
 1. Select the control and press F2. The different events associated with the control are displayed.
 2. Enter the following WLanguage code in the "Click (Server)" event of the Button control:

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

3. Let's study this code:
 - **PageDisplay** is used to display a specific page.
 - **PreviousPage** is used to find out the name of previous page.
 - This line of code is used to display the previous page
4. Save the modifications ( or Ctrl + S).
5. Close the code editor.



Remark

By default, any Button control created in a page sends the value of page controls to the server ("submit" button). This is the most common case.

Two other options are available:

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

These options can be selected in the "General" tab of the Button control description window , option "During the action".

Validate button

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

The "VALIDATE" Button control will be used to:

- check the data typed: this check consists in verifying that the different page controls have been filled by the user. This check does not require any access to the server and it can be performed in browser code.
- save the data typed in the Product data file. It is saved in server code. The data is sent to the server then it is saved in the data file.

- ▶ Write the code associated with the "Validate" button:

1. Select the "VALIDATE" Button control and press F2. The events associated with the Button control are displayed.
2. In the "Click (browser)" event of the Button control, write the following code for checking the data entered:

```
IF EDT_Reference ~= "" THEN
  Error("A reference must be typed.")
  ReturnToCapture (EDT_Reference)
END
IF EDT_Caption ~= "" THEN
  Error("A caption must be typed.")
  ReturnToCapture (EDT_Caption)
END
IF EDT_Description ~= "" THEN
  Error("A description must be typed.")
  ReturnToCapture (EDT_Description)
END
IF EDT_PriceBT = 0 THEN
  Error("A price must be typed.")
  ReturnToCapture (EDT_PriceBT)
END
```

3. Let's take a look at this WLanguage code:

- For each edit control found in our page, a check is performed to find out whether a value was typed.
- The "!=" operator is used to check the equality, excluding the case and the punctuation.
- If no value is typed, an error message asks the Web user to perform an input (**Error**). The code execution is stopped and the input is forced in the relevant edit control by **ReturnToCapture**.

4. In the "Click (server)" event of the Button control, write the WLanguage code for saving the data:


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


5. Let's take a look at this WLanguage code:

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

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

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

- **HModify** is used to update the file data for the current record.
 - **PageDisplay** is used to display another page. In our case, "PAGE_List_of_products" is redisplayed.
6. Save the modifications ( or Ctrl + S).
7. Close the code window.

Managing the product image

In the Product data 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.



Remark

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.



Remark

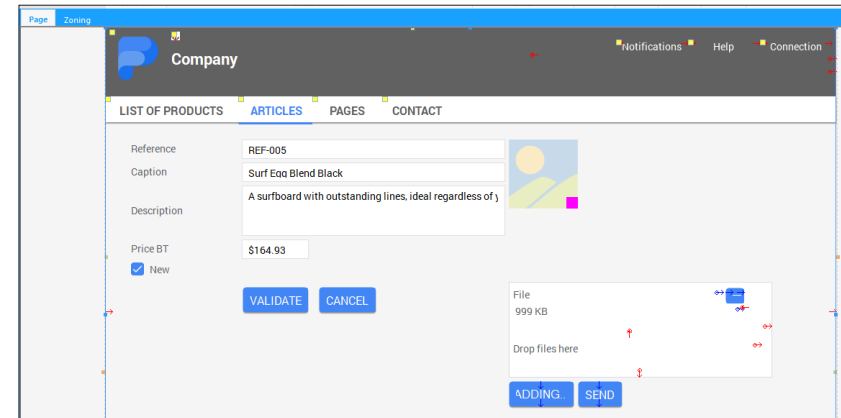
WEBDEV proposes to manage the file upload via two specific controls:

- 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 "Uploading files".
3. Click the position where the control must be created in the page (below the Image control for example).



Remark

The Upload control includes:

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

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

► To configure the Upload control:

1. Display the code of the "ADDING" Button control: select the control and press F2.
2. Different WLanguage events are associated with the Upload control. We are going to modify the event "Receiving the files uploaded" to copy the image to the site data directory.
3. Write the following code in the event "Receiving the files uploaded" (ignore compilation errors, we will correct them later):

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



Remark

This code is using "...". These 3 dots are used to perform a line break in a code line. They are used here for layout purposes.

In the code editor, you can delete them and use a single code line.

4. Let's study this code:

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

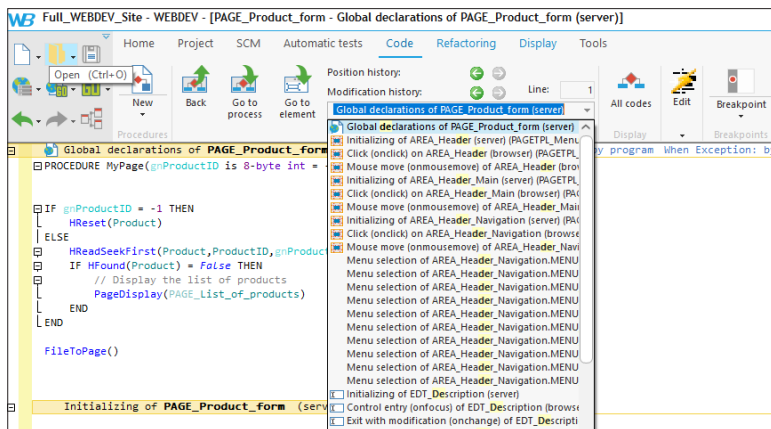


Remark

Why use a global variable?


A global variable is required here because the image path will be used in the code of "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 server ("\" for Windows, "/" for Linux). Therefore, your site is independent of the server where it is installed!
 - Then, the uploaded image is displayed in the IMG_Visual Image control.
5. When typing and saving the code, the **gsImagePath** variable is displayed in red and a compilation error occurs in the error pane: "gsImagePath' identifier unknown or inaccessible". Indeed, this global variable was not declared.
6. To declare the global variable:
- Display the "Global declarations" event of the page (for example, in the Code editor, go to the "Code" pane, and select "Global declarations" in the combo box that lists all the events).



- Type the following code after the procedure declaration:

```
gsImagePath is string
```

7. Save the modifications by clicking  among the quick access buttons. The compilation errors disappear.

8. Close the code editor.

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

► To save the image in the Product data file:

1. Display the code of the "Validate" Button control:

- Select the "Validate" Button control.
- Press F2.

2. In the "Click (server)" event, add the following code AFTER the call to **PageToFile**:

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

3. Let's take a look at this WLanguage code:

- This code checks the content of **gsImagePath** global variable. 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 Product data file is filled with the binary image content. 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).

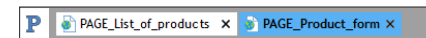
5. Close the code window.

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 will display the details via a Link control.

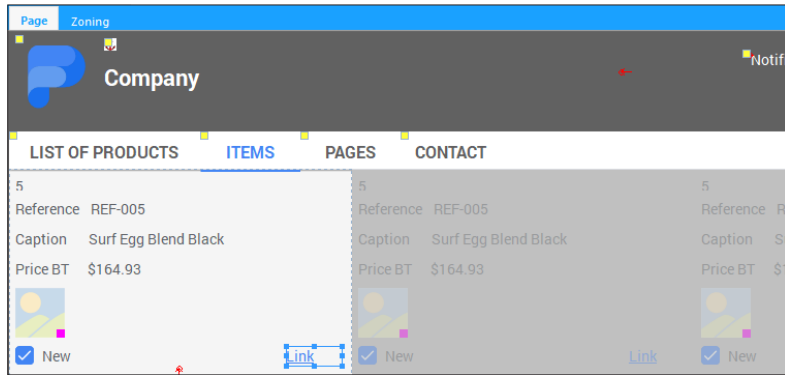
► We are going to modify "PAGE_List_of_products" in order to create a modification Link control:

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



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

3. Then, click inside the Looper control (bottom right, for example) to create the Link control.



4. Change the Link control caption (press Space to edit the caption): the new caption is "Modify".

▶ The "Modify" Link control must open "PAGE_Product_form". We are going to open this page by programming.

1. Select the "Modify" Link control and display the associated WLanguage events (F2).
2. In the code window that appears, write the following code in the "Click (server)" event:

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



Remark

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

3. Let's take a look at this WLanguage code:

- **PageDisplay** is used to open the "PAGE_Product_form" page.
- The opened page expects in parameter the identifier of product to display. This identifier corresponds to the identifier of 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 brackets are used to specify the row and LOOP_Product is used to get the current row in the Looper control.



Remark

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

```
PageDisplay (PAGE_Product_form, ATT_ProductID)
```

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

The different elements used to manage the product modification have been implemented, we are now going to run a test to make sure that everything is operating.

- ▶ Run the project test (among the quick access buttons).
 1. The editor asks for the first page of Session mode. In our case, select the "PAGE_List_of_products" page and validate.



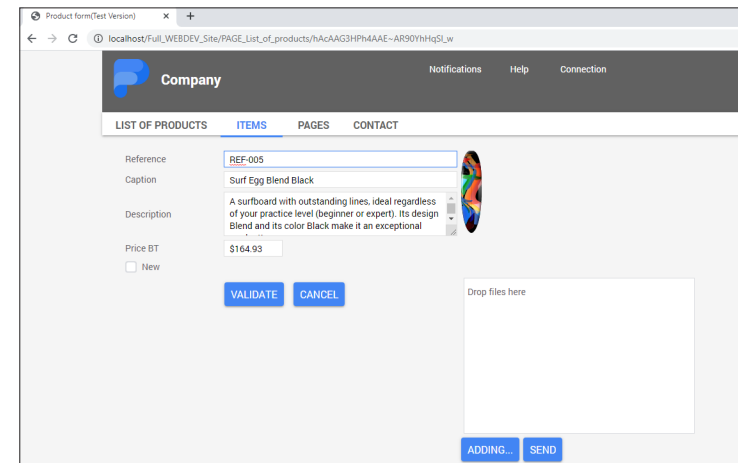
Remark

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

- during the project test.
- in the "Project explorer" pane: simply select the requested page, display the popup menu and select "First project page in Session mode".

 When you define a page as first page in Session mode, a little red "1" appears next to the page name in the "Project explorer" pane.

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



- Modify the price of a product and validate.
- The new price of 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 want to be able to add a product. To do so, there is not need to re-create a new page: we will be using the "PAGE_Product_form" page that was created beforehand and adapt it to manage the addition.

- ▶ First of all, we are going to modify the opening mode of "PAGE_Product_form" page:
 1. Position on the "Product form" page: click the "PAGE_Product_form" button found in the open documents bar.
 2. Press F2 to display the different WLanguage events of the page.
 3. In the "Global declarations" event, 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 product to modify. But when adding a record, the element identifier does not exist. To manage this case, we will be using the default value "-1".
 4. Replace the following line of code in the "Global declarations" event:

```
PROCEDURE MyPage (gnProductID is 8-byte int)
```

by the code:

```
PROCEDURE MyPage (gnProductID is 8-byte int = -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. This function initializes the item variables in the "Product" data file with the default values to manage a new record.
 - If the product identifier has a value different from -1, we retrieve the code used to open the form in modification.
6. Close the code window.
- ▶ The "Validate" Button control must also support the addition of the record.
 1. Select the "Validate" Button control in "PAGE_Product_form".
 2. Display the WLanguage events associated with the control (F2).
 3. The code in the event "Click (browser)" must not change: the checks to perform are always the same. Only the server code must be modified.
 4. In the "Click (Server)" event, replace the existing code with the following code:

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

Let's study this code:

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



Remark

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

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

5. Save the modifications ( or Ctrl + S).

6. Close the code window.

► We are now going to modify the page menu to allow the user to add a new product.

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

2. Display the popup menu of a menu option (right click) and select "Open the template".

3. In the page template, select the menu.

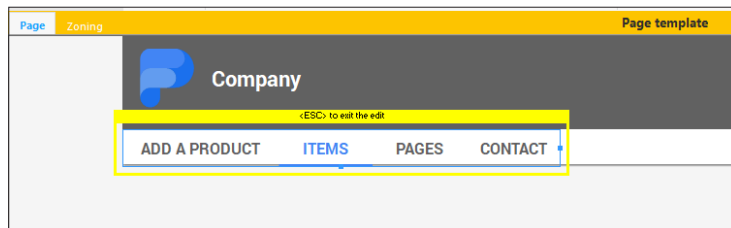
4. Click the "ITEMS" option. A yellow border appears around the menu. This yellow border indicates that the menu is in "Edit" mode: it can be modified.


5. Open the option description window ("Option description" from the popup menu).

6. In the "General" tab:

- Type the option caption: replace "ITEMS" with "ADD A PRODUCT"
- In the Action area:
 - select "Display a site page".
 - select the "PAGE_Product_form" page.

7. Validate. The menu option appears:



8. In the orange bar of template, click  to propagate the template modifications to all pages that use this template.

9. In our case, the two project pages are proposed. Validate the page template update window.

10. Close the page template.

We can now check the product addition.

Test of product addition

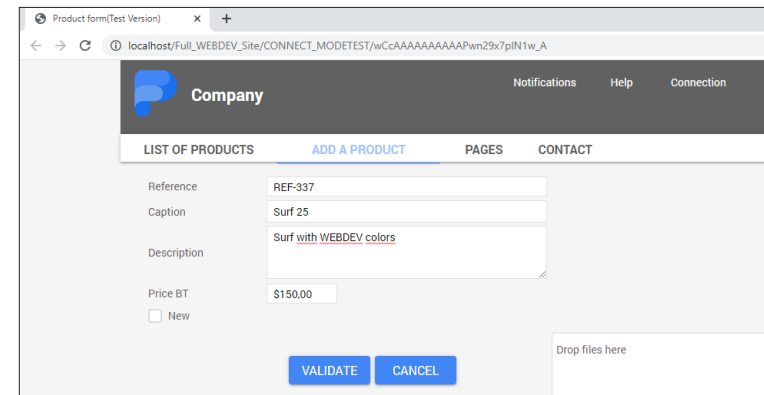
► To check the product addition:

1. Run the project test ( among the quick access buttons).

2. Click "ADD A PRODUCT".

3. Enter a new product with the following characteristics:

- Reference: REF-337
- Caption: Surf 25
- Description: Surf with WEBDEV colors
- Price: 150



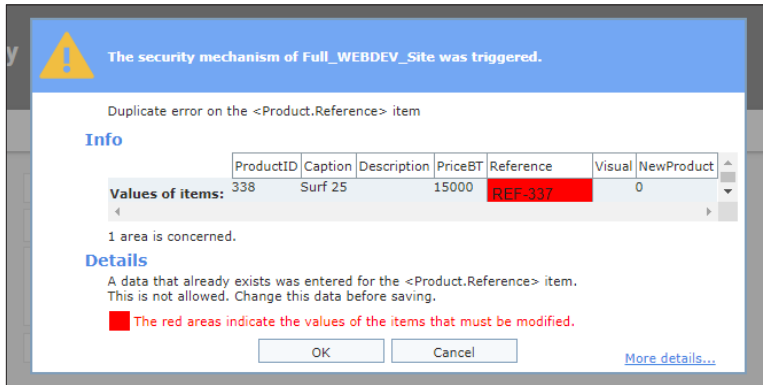
4. Validate. The new product is displayed in the list of products.

5. Click "ADD A PRODUCT" again.

6. Enter a new product with the following characteristics:

- Reference: REF-337
- Caption: Surf 25
- Description: Surf with WEBDEV colors
- Price: 150

7. Validate. A specific page is displayed:



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

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



Remark

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

WEBDEV proposes several modes for managing these errors:

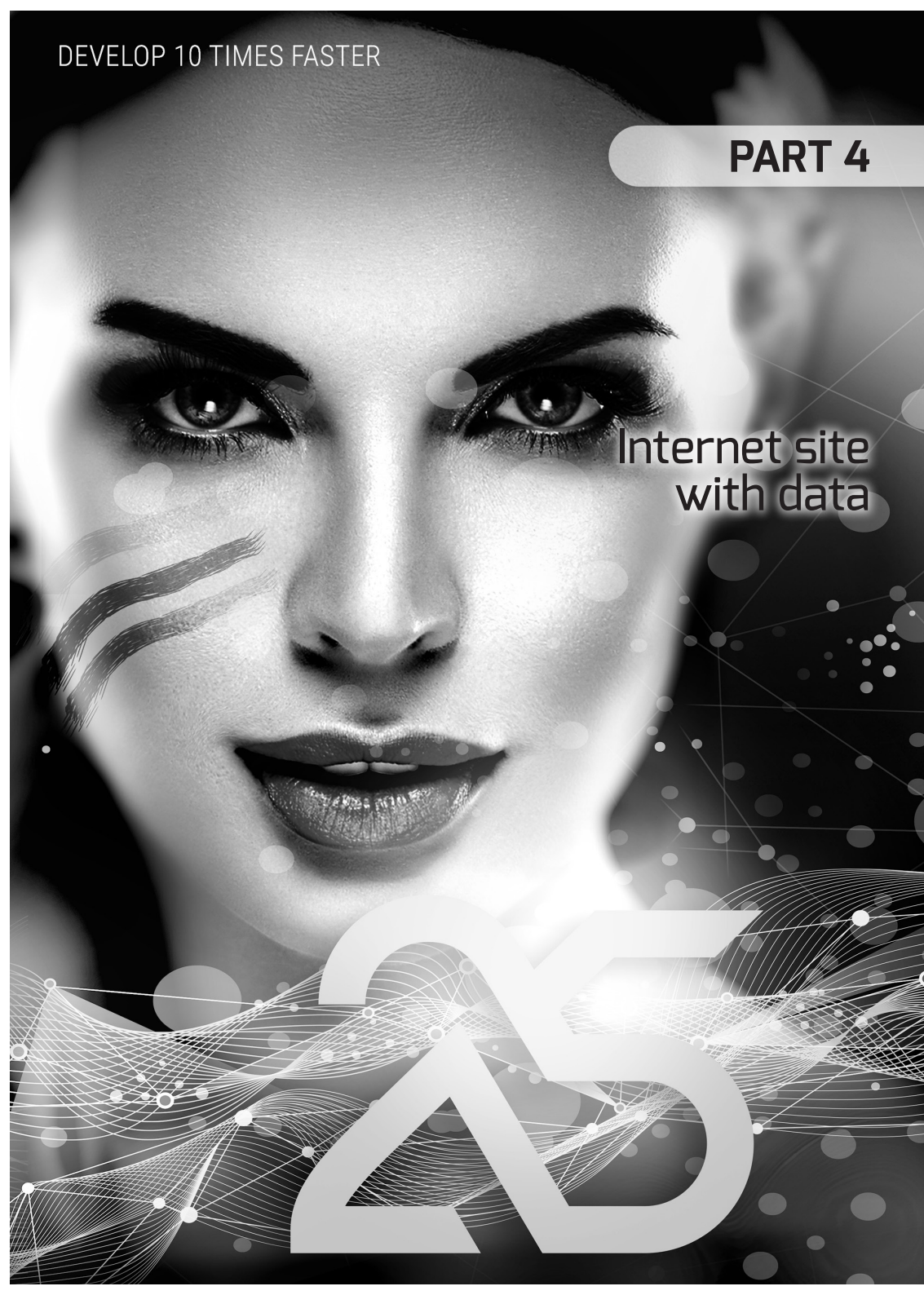
- **the automatic mode:** for each error that occurred when managing database records, a specific page is displayed to the user. This page allows the user to modify his data directly.
- **the advanced programmed mode:** a procedure or a custom page for error management is called whenever an error occurs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

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

Viewing the records

The new created records can be immediately viewed in the Looper control of "PAGE_List_of_Products" page.

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



LESSON 4.1. OVERVIEW

This lesson will teach you the following concepts...

- Overview of the site created in this part.

 Estimated time: 5 mn

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 see their details.

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



Remark

SEO

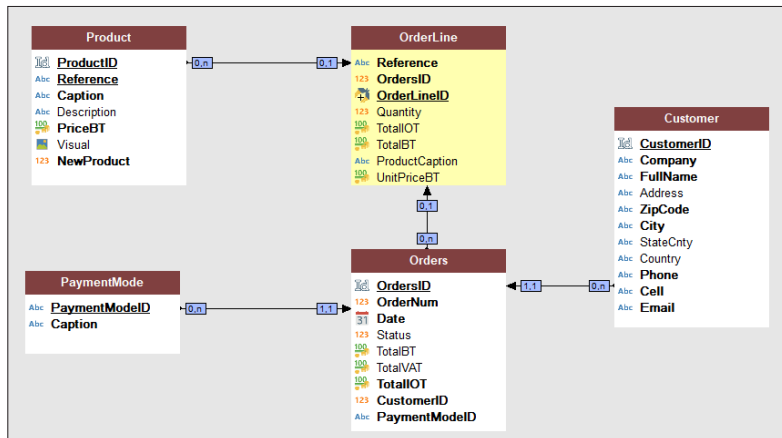
In a WEBDEV site in AWP mode, all site pages can be referenced.

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

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

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

Reminder: We are going to work on a project already associated with an analysis and with 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" paragraph).

LESSON 4.2. PRINCIPLE FOR DISPLAYING AN AWP SITE

This lesson will teach you the following concepts...

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

 Estimated time: 20 mn

Principle for displaying a WEBDEV AWP site

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

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

What is an AWP page?

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

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

Operating mode of AWP sites

For each page in AWP mode displayed in the browser, the following elements are automatically created on the server:

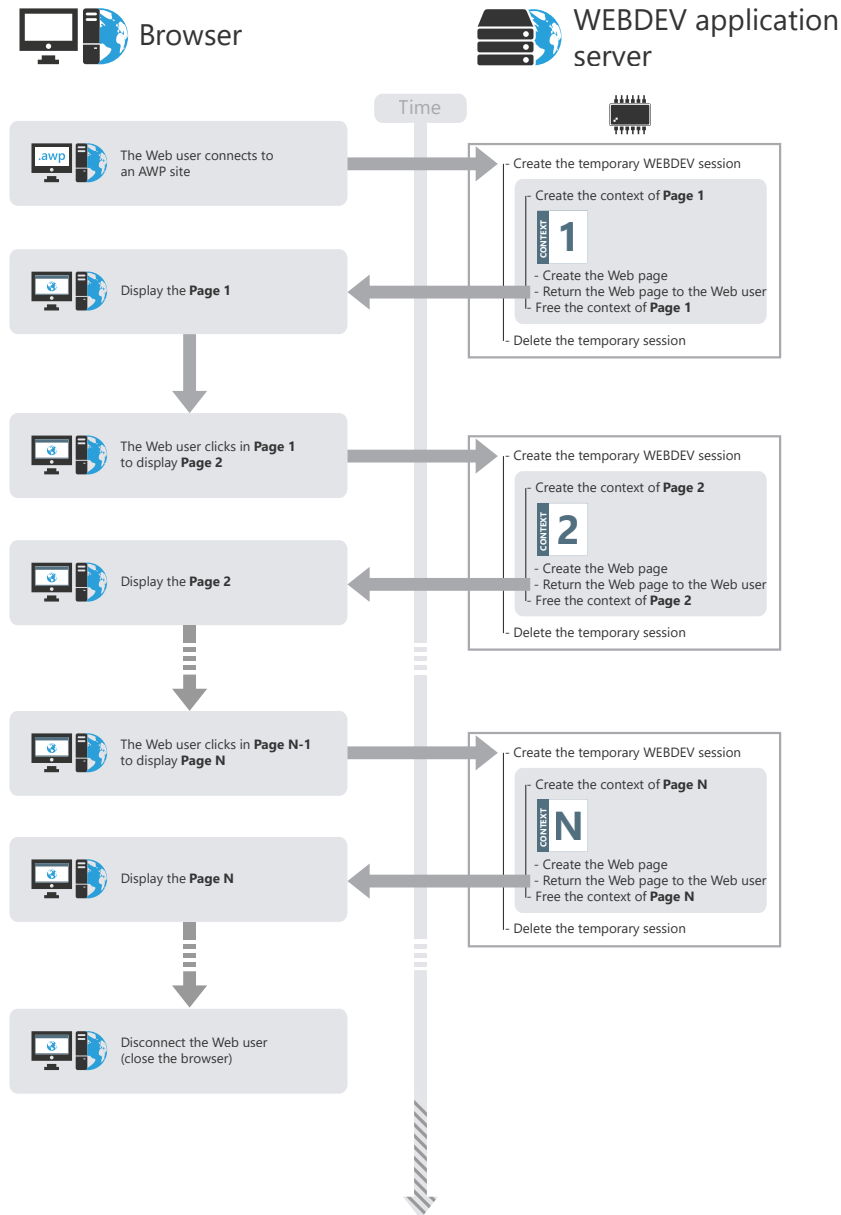
- a temporary session,
- a temporary AWP page context.

The **temporary session** contains the temporary context of the page in AWP mode. When the page in AWP mode is 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 context of the page in AWP mode** contains all elements that have been required to build the page viewed by the Web user:

- local variables,
- the server processes,
- database connections,
- the file contexts, etc.

When the page in AWP mode is sent to the Web user, these elements are destroyed.



How to share information (values) between AWP pages?

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

- 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

It is possible to pass information from one page to another via the URL. The URL has the following format: "http://Web server/.../mypage.awp?NameParam1=Value1&NameParam2=Value2".

This method allows a better page referencing 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 pages in AWP mode, via the **AWP contexts**. An AWP context is created on disk on the server. This context is available as long as pages in AWP mode are displayed and as long as the timeout of AWP contexts is not exceeded. The timeout of AWP contexts is defined in the WEBDEV administrator ("Configuration" tab, "Duration of AWP contexts" option).

To manage AWP contexts, use functions such as *DeclareAWPContext*, *FreeAWPContext*, etc.

See the online help for more details.

LESSON 4.3. CREATING AWP PAGES

This lesson will teach you the following concepts...

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



Estimated time: 50 mn

Overview

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

We are going to perform these operations in the "Full_WEBDEV_Site" project. We worked on this project in the previous section.

- ▶ To open this project:
 - If you have opened the exercise "Full WEBDEV Site" in the previous section, open the project you have been working on:
 1. Open the WEBDEV home page (Ctrl + <).
 2. In the home page, click "Tutorial" and select "Full WEBDEV Site (Exercise)".
 3. WEBDEV informs you that this project has already been opened and proposes opening the local version or overwriting this project with the initial version. To retrieve your changes, click "Open the local copy".
 - If the "Full WEBDEV Site" exercise was not opened in the previous section:
 1. Open the WEBDEV home page (Ctrl + <).
 2. In the home page, click "Tutorial" and select "Full WEBDEV Site (Exercise)".
- Caution:** In this case, there may be some differences regarding the images in this tutorial and it will not be possible to link the Internet and Intranet part of the site.




Answer

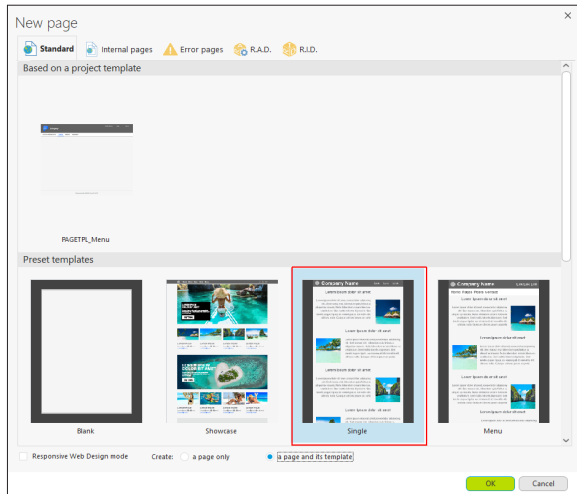
A corrected project is available. This project contains the different pages created in this lesson. To open the corrected project, in the home page, click "Tutorial" and select "Full WEBDEV Site (Answer)".

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.

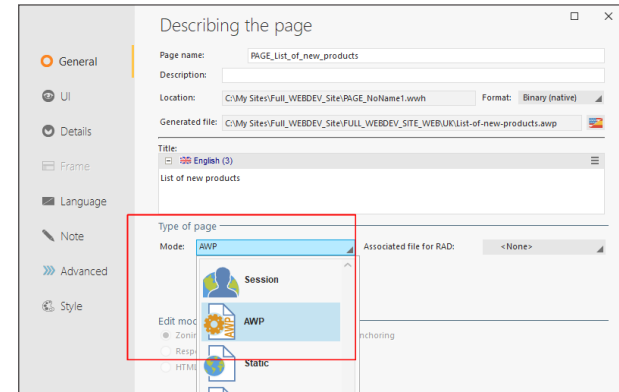
Creating the page

- ▶ To create a page used to list new products:
 1. Create a new page:
 - Click  among the quick access buttons.
 - The element creation window appears: click "Page" then "Page".
 - In the wizard for page creation, click the "Single" preset page.



- Make sure that option "Create: a page and its template" is checked.
 - Validate the wizard.
2. The new page is displayed in the editor. The save window is displayed.
 3. Type the page title: "List of new products". The page name ("PAGE_List_of_new_products") is automatically proposed.
 4. Validate.
- The page was created with the controls proposed by default. Delete these controls:
1. Select the controls found in the page (press Ctrl + A).
 2. Delete the controls (Del key).
- 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. Go to "General", "Type of page", "Mode", and select "AWP".



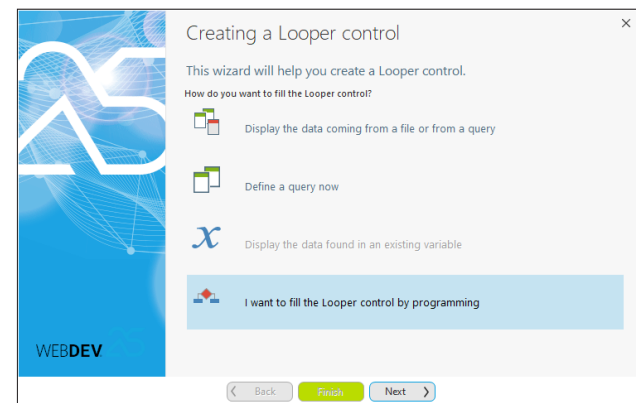
3. Validate.
4. Save the modifications by clicking among the quick access buttons.

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. In this page, the Looper control will be filled by programming.

Creating the Looper control

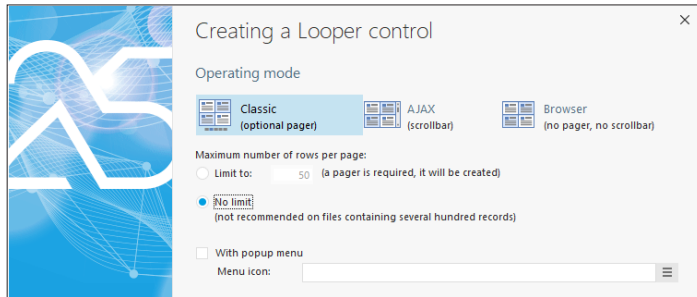
- To create the Looper control:
1. On the "Creation" pane, in the "Data" group, expand "Looper" and select "Looper". The control currently created follows the mouse movement.
 2. Click in the top left corner of the page: the wizard for creating the Looper control starts.
 3. In the assistant, select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



Go to the next step.

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

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



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

- Keep the proposed display direction: fixed number of columns, fill in row.
- The Looper control will display the new products on 5 columns.
In the "Number of columns" area, replace 1 by 5.

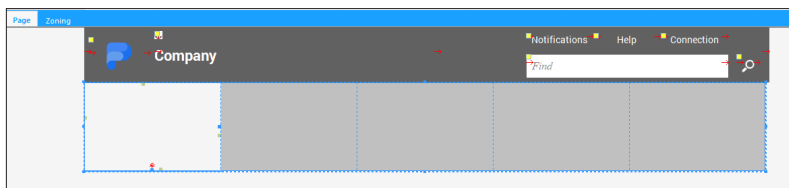
6. Go to the next step.

7. The last step is used to specify the control name (LOOP_NewProducts for example).

8. 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.

► With the handles, reduce the width of main cell in order for the 5 cells to fit across the page width.



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

Each row will display:

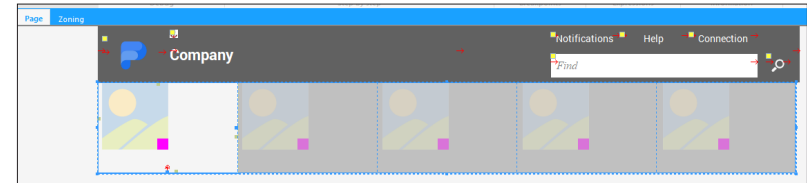
- An Image control used to display the product image.
- A Link control used to display the product name and to open the detailed form.
- A Static control used to store the product identifier.

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

Creating controls inside the Looper control

► 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 see the available area.
3. Click the top left corner of first row in the Looper control. The Image control is created.



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

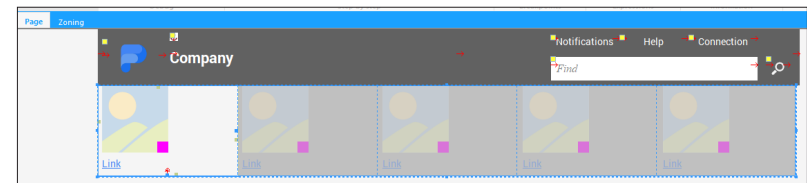
5. Modify:

- the control name: IMG_Visual.
- the display mode of image: select "Homothetic" and check "Without image enlargement" and "Center image in the control".
- the transformation mode of image. In our case, the product image is stored in memo format in an item of the Product data file. Choose a display calculated on the server to optimize bandwidth and check "High quality".

6. Validate.

► To create the Link control:

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



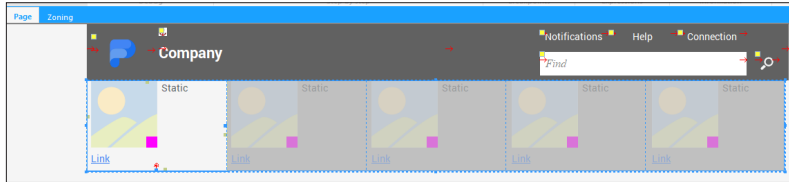
3. Display the description of Link control to modify its characteristics (double-click the control).

4. This control is named "LINK_View_Product".

5. Validate the control description window.

6. Select the Link control and increase its width (with the handles) so that it occupies the row width.

- ▶ To create the Static control containing the product identifier:
 1. On the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Simple static".
 2. Click the Looper control, to the right of the Image control to create the Static control.



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

We are going to create attributes to fill the controls in the Looper control and to modify their characteristics on each row.

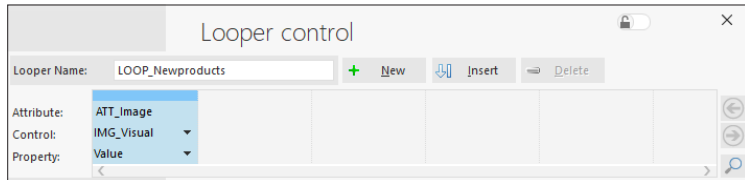


Remark

A Looper control includes:

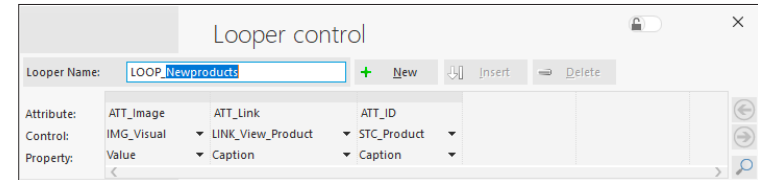
- controls, that are repeated on each row.
- attributes. An attribute is used to define the control characteristic that will be modified on each row. For example, if the value and the color of PRICE control must change on each row, you will have to define two different attributes for the same control.

- ▶ To create attributes:
 1. Display the description window of Looper control (double-click the control).
 2. In the top section of window, no attribute is created. We are going to define 3 attributes (one for each looper control).
 3. Create a new attribute by clicking the "New" button:
 - Rename the attribute to "ATT_Image".
 - Select the "IMG_Visual" control.
 - Select the "Value" property. Indeed, the image value will be modified for each row.



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



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

The Looper control is created now. Let's now program the fill operation for the Looper control.


Fill of the Looper control

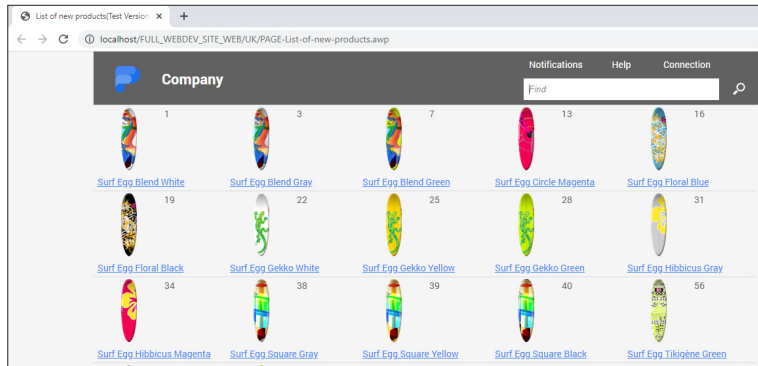
The Looper control is filled when initializing the page.

- ▶ To fill the Looper control:
 1. Display the WLanguage events associated with the page (click inside the page and select "Code" in the popup menu, or press F2).
 2. Write the following code in the "Initializing" event of the page:

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

3. Let's take a look at this WLanguage code:
 - This code browses the Product data 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 different row attributes are updated with the data of current record:
 - the "ATT_Image" attribute value corresponds to the image found in the "Visual" item of the "Product" data file.
 - the value of the "ATT_Link" attribute caption is the content of the "Caption" item found in the "Product" data file.
 - the value of the "ATT_ID" attribute caption is the content of the "ProductID" item found in the "Product" data file.
4. Save the modifications (or Ctrl + S).
5. Close the code window.

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




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

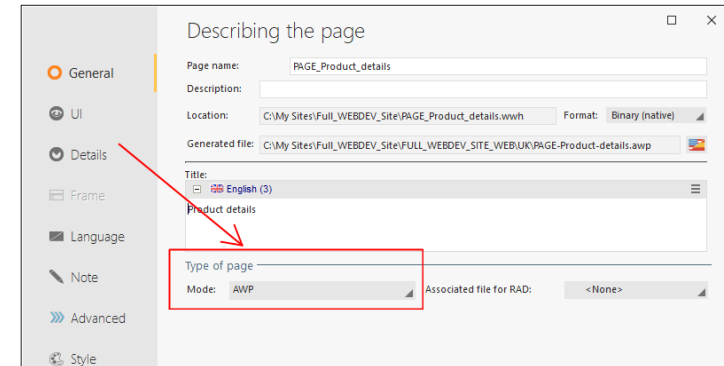
Viewing the details of a new product

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

Creating the page

- ▶ To create a page used to display the product details:
 1. Create a new page.
 - Click  among the quick access buttons.
 - The window for element creation is displayed: click "Page" then "Page".
 - The page creation wizard starts.
 - In the "Based on a project template" section, choose "PAGETPL_Single" and finish the wizard.
 2. The window for saving the page is displayed. Type the title: "Product details". The name ("PAGE_Product_details") is automatically proposed.
 3. Validate.

- ▶ Like the page used to list new products, this page must be generated in AWP mode:
 1. Display the description window of the page (select "Description" from the popup menu).
 2. In the "Type of page" area, select the "AWP" mode.



3. Validate.
4. Save the modifications ( or Ctrl + S).

What should this page do?

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

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

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

- ▶ To declare a parameter in the page:
 1. Press F2. The code editor displays the different page events.
 2. Write the following code in the "Global declarations" event:

```
PROCEDURE MyPage(gnProductID is 8-byte int)
```

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

Creating controls

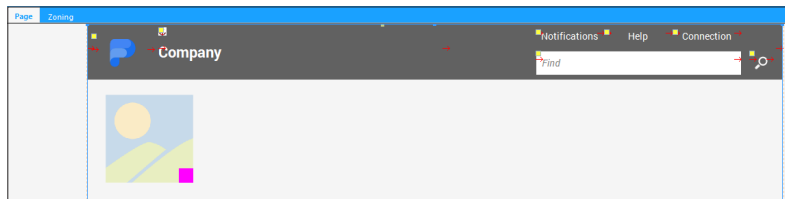
We are now going to create the controls used to display information about the selected product. In the previous part, we explained how to create controls by Drag and Drop from the analysis pane. In this part, we are going to create the controls one by one then associate them with the corresponding item in the data file.

We are going to create the following controls:

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

► To create the Image control:

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



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

- This control is named "IMG_Product_Image".
- Choose a display calculated on the server to optimize bandwidth and check "High quality". Indeed, the display mode of image is "Homothetic". Check "Without image enlargement" and "Center image in the control".



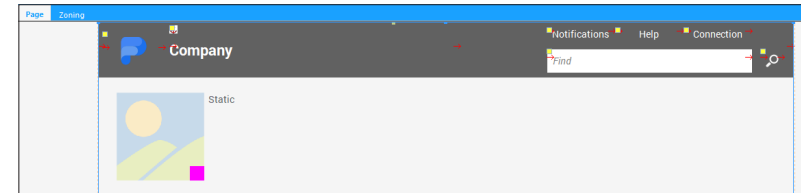
Remark

With the options "Homothetic", "Center image in the control" and "Without image enlargement", the image size will adapt homothetically to the area defined for the image. The proportions will be respected and the image will not be enlarged.

4. Validate the control description window.

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

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



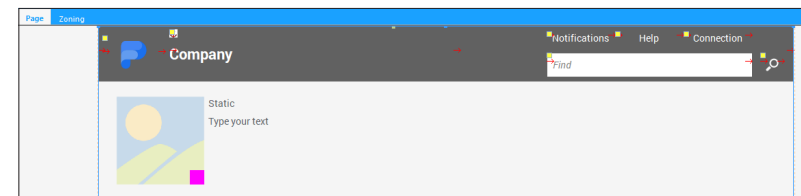
Remark

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 see the control border and to easily position the controls in relation to the other ones.

This feature is very useful for the controls without border.

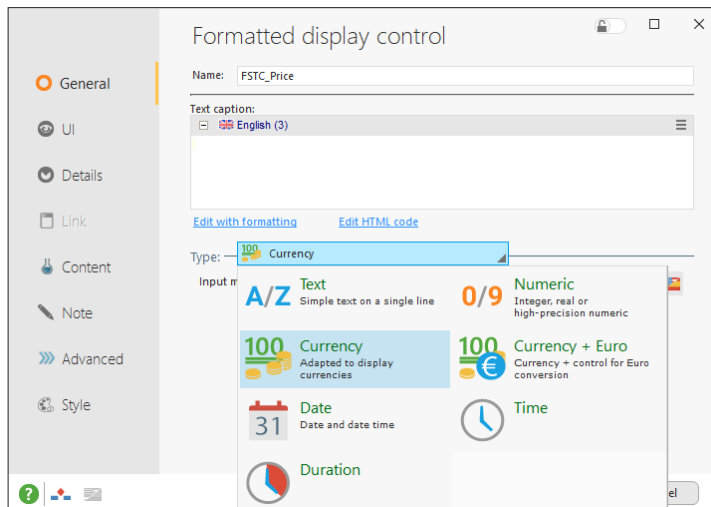
3. Rename the Static control: "STC_Title".
 4. Enlarge the control horizontally (using the sizing handles).
- We are going to display the product description in a rich Static control (called Rich Text Area). To create this control:
1. On the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Rich Text Area". You also have the ability to click the "Text" icon directly.
 2. Click inside the page (to the right of image for example, below the STC_Title control).



3. Rename the control: "RTA_Description".

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

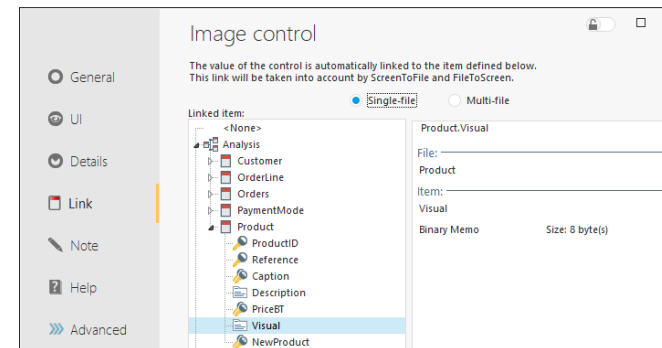
- ▶ To create a formatted display control:
 1. On the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Formatted Display control".
 2. Click in the page, below the product description: the control is automatically created.
 3. Reduce the control width using the sizing handles.
- ▶ We are going to modify the characteristics of this control:
 1. Display the description window of the control that was just created: double-click the control for example.
 2. In the "General" tab:
 - Modify the control name: "FSTC_Price".
 - Delete the caption.
 - Modify the control type: select "Currency".



3. Validate.

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

- ▶ To link the "IMG_Product_Image" control to the corresponding item:
 1. Select the "IMG_Product_Image" control.
 2. Display the popup menu (right mouse click) and select "Description".
 3. Go to the "Link" tab in 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 Product data file.
 4. In "Linked item", expand "Analysis" then "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" data file.
 - RTA_Description control linked to the "Description" item of the "Product" data file.
 - FSTC_Price linked to the "PriceBT" item of the "Product" data file.
 - STC_Reference control linked to the "Reference" item of the "Product" data file.
- ▶ Save the page (Ctrl + S).

Displaying data in the page

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

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

► To display data in the page:

1. Press F2. The code editor displays the different page WLanguage events.
2. In the "Global declarations" event, write the following code below the code previously written:

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

3. Let's take a look at this WLanguage code:

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

4. Close the code editor.

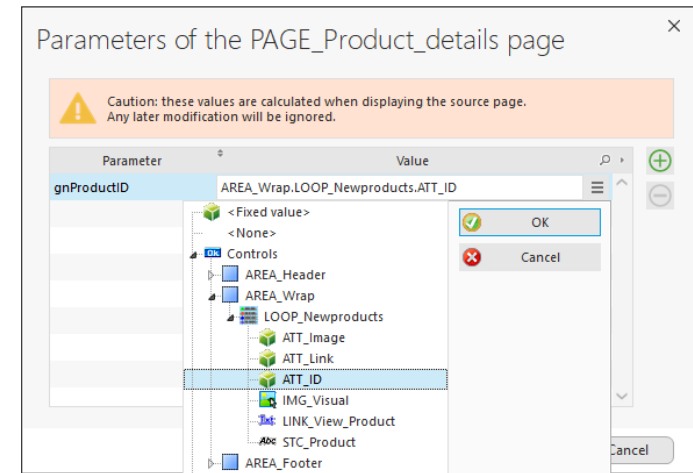
Displaying the form from the list of products

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 control.

► First, we will modify the "PAGE_List_of_new_products" page so that the Link control opens the product description window.

1. Position the cursor on the "List of products" page: click "PAGE_List_of_products" in the open documents bar.
2. Display the description window of Link control: double-click the control.
3. In the "General" tab of the description window, in "Link action":
 - Select "Display a site page".
 - Select the "PAGE_Product_details" page.
 - Click the "Parameters" button. We are now going to define the parameter that will be passed to the page in order to display the details of selected product.
4. In the window that appears, you will find the name of the parameter declared in the event "Global declarations" of the page (gnProductID). All you have to do is select the control containing the parameter value. In our case, the product identifier is found in the "ATT_ID" attribute.

5. Expand and select "ATT_ID":



6. Click "OK" and validate the parameters description window.




Remark

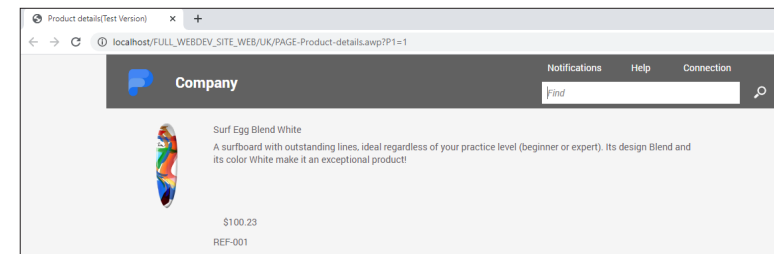
Caution: In a Looper control, the value is not contained in the control but in the attribute that is associated with it for the value characteristic.

7. Validate the description window of Link control.

8. Save the modifications ( or Ctrl + S).

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

1. If necessary, open "PAGE_List_of_new_products" in the page editor (use the documents bar).
2. Click  among the quick access buttons.
3. The page is automatically displayed in the browser.
4. Click the link to display the product details.



► Close the pages to stop the test.

Managing styles

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



Remark

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 input area.



Remark

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

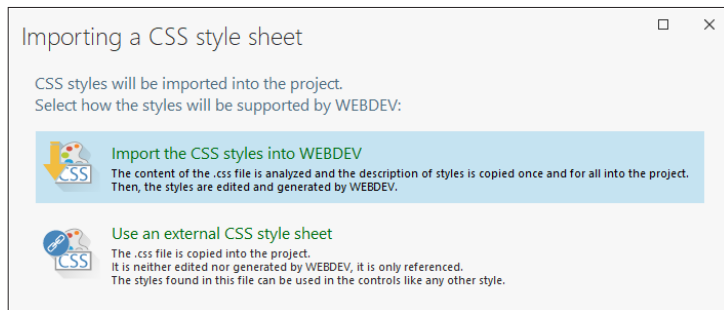
The skin provides the images, the fonts, the texture, the shape of buttons and the styles available for the project.

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

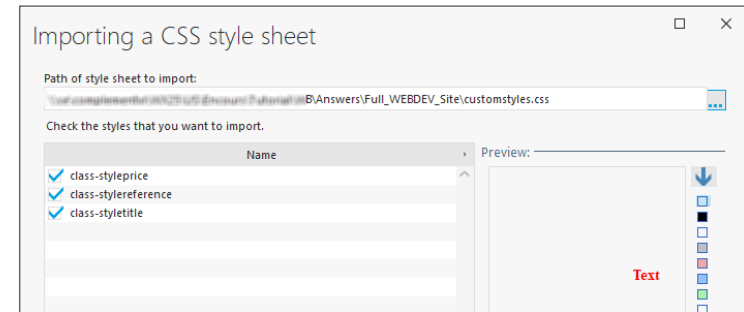
Importing a CSS style sheet

► To import the CSS style sheet into the project:

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



- Import CSS styles into WEBDEV: 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 company).
3. Select "Import the CSS styles into WEBDEV" and click "Next".
 4. Select the "customstyles.css" file. This file is found in the directory "\\Tutorial\WB\Exercises\Full_WEBDEV_Site". The various styles found in the style sheet are displayed.



- A preview of 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 proposed styles and validate. The styles are imported and they are available in WEBDEV.

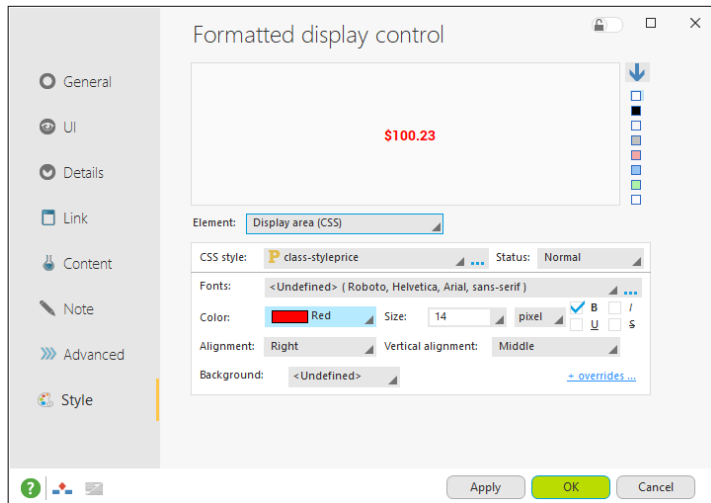
Applying CSS styles


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

1. Position on the "PAGE_Product_details" page.
2. Display the description window of "FSTC_Price" control that displays the product price (double-click the control).


3. In the "Style" tab:

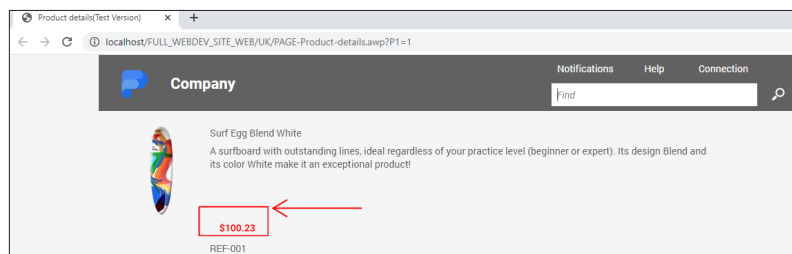
- Select the "Display area (CSS)" element.
- Select the "class-styleprice" style in the "CSS style" list:



4. Validate. 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. If necessary, open "PAGE_List_of_new_products" in the page editor (use the open documents bar).
2. Click  among the quick access buttons.
3. The page is automatically displayed in the browser.
4. Click the link to display the product details. The price is now displayed in red.



► Close the pages to stop the test.

Linking the Internet site and the Intranet site

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

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

To do so, we are going to modify the template of 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:

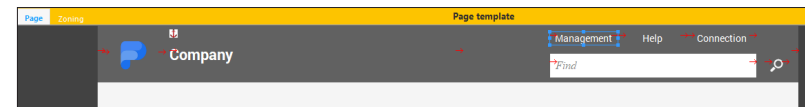
1. Select "PAGE_List_of_new_products".
2. Open the popup menu of one of the links at the top of the page and select "Open template".
3. Display the description of Notifications control (double-click the control).
4. Modify the control caption: Management.



Tip

In your programs, we advise you to choose meaningful names for your elements (in this case, the Link control is named "LINK_Management"). Thus, your code will be more readable.


5. In the "Link action" area, for the "Type" option, select "Run browser click code only".
6. Validate.



► We will now write the WLanguage code associated with the Link control:

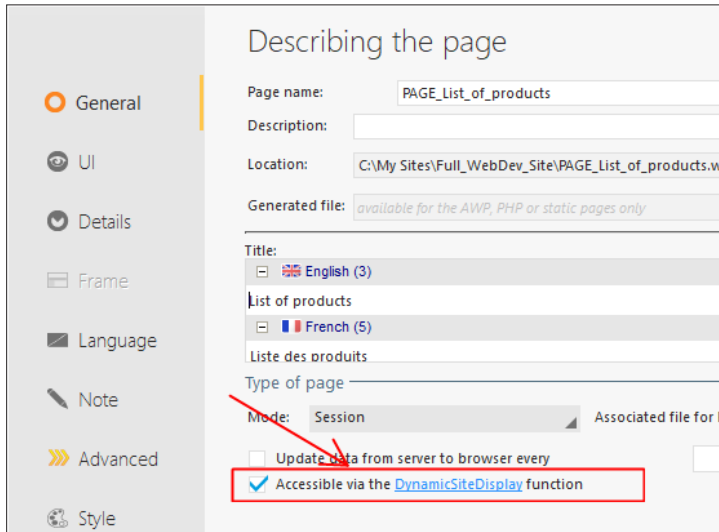
1. Display the events associated with the Link control (F2 on the control).
2. Write the following code in the "Click (browser)" event:

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

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

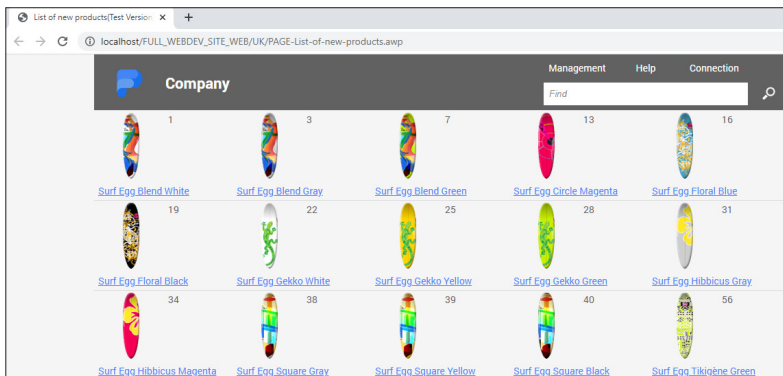
► We are now going to modify the "PAGE_List_of_products" page so that it can be opened by **DynamicSiteDisplay**:

1. Open the "PAGE_List_of_Products" page in the editor.
2. Display the description window (select "Description" from the popup menu).
3. In the "General" tab, check "Accessible via the DynamicSiteDisplay function".

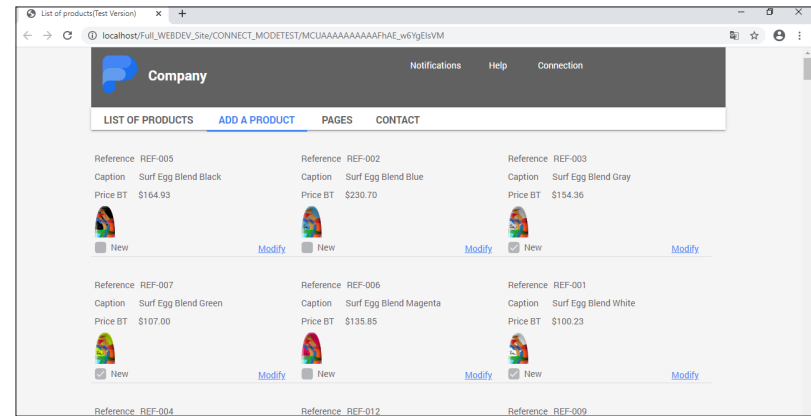


4. Validate.

- We are now going to run the project test via the AWP pages.
1. If necessary, open "PAGE_List_of_new_products" in the page editor (use the documents bar).
 2. Click among the quick access buttons.
 3. The page is automatically displayed in the browser.



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



► Close the pages to stop the test.

Creating a page used to display the general sales conditions

We are now going to create a type of page that is often used in the e-commerce sites: a page displaying the general sales conditions. To do so, we will be using the Rich Text Area control, allowing you 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 element creation window appears: click "Page" then "Page".
 - The page creation wizard starts.
 - In the "Based on a project template" section, choose "PAGETPL_Single" and finish the wizard.
 2. The window for saving the page is displayed. Type the title: "General sales conditions". The name ("PAGE_General_sales_conditions") is automatically proposed.
 3. Validate.

Creating the control

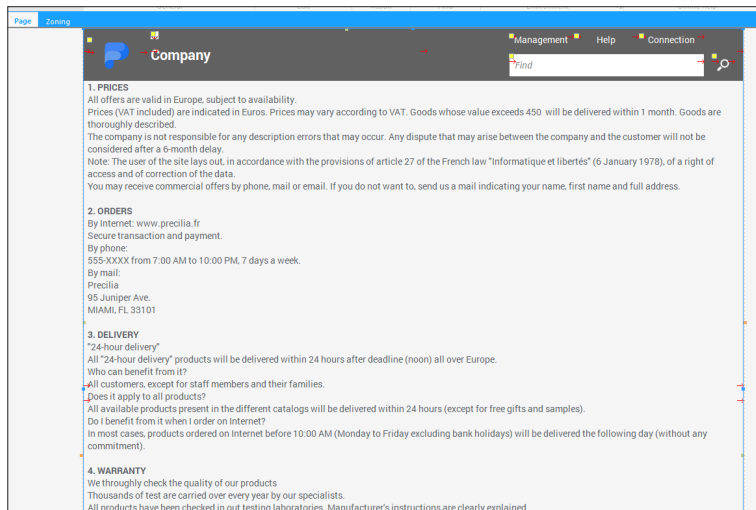
- To create a rich Static control:
1. In the ribbon, on the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Rich Text Area".
 2. Click in the page (top left corner for example). The control is automatically created.


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

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

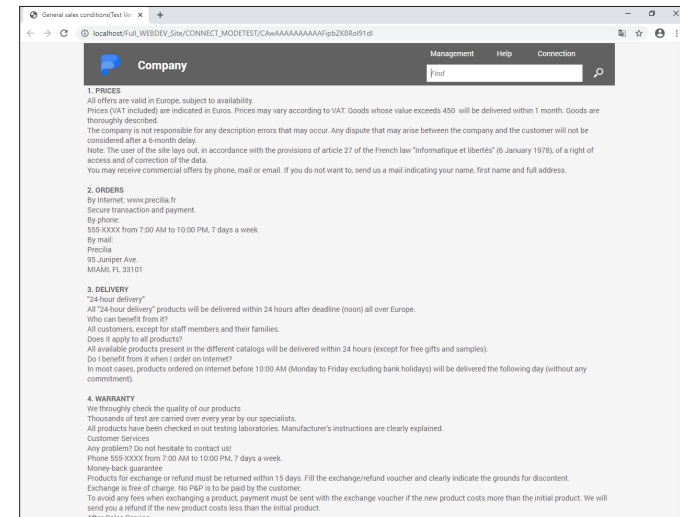
► Now, we are going to display the general sales conditions in this control. The file containing the text was prepared beforehand.

1. Copy the text in the GSC.txt file. This file is found in "\Tutorial\WB\Exercices\Full_WEBDEV_Site" in the WEBDEV setup directory.
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.
 - Write the titles of different points in bold.



6. Try the different options: all 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, ...

PART 5

Let's continue
the development



LESSON 5.1. MULTICRITERIA SEARCH

This lesson will teach you the following concepts...

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



Estimated time: 1h30 mn

Overview

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

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

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

1. Display the WEBDEV home page if necessary (Ctrl + <).
2. In the home page, click "Tutorial" and select "Full WEBDEV Site (With pages)".



Answer

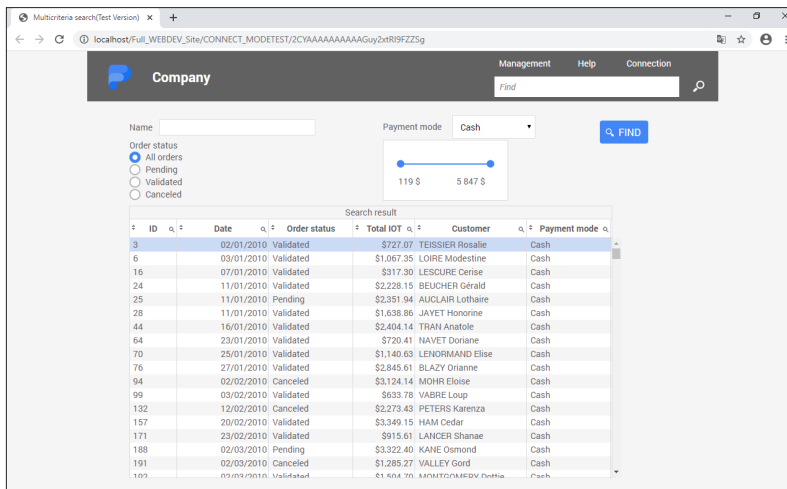
A corrected project is available. This project contains the different pages created in this lesson. To open the corrected project, in the home page, click "Tutorial" and select "Full WEBDEV Site (Answer)".

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

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

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

The interface of "PAGE_Multicriteria_search" will be as follows:



This interface includes:

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

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



Remark

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.

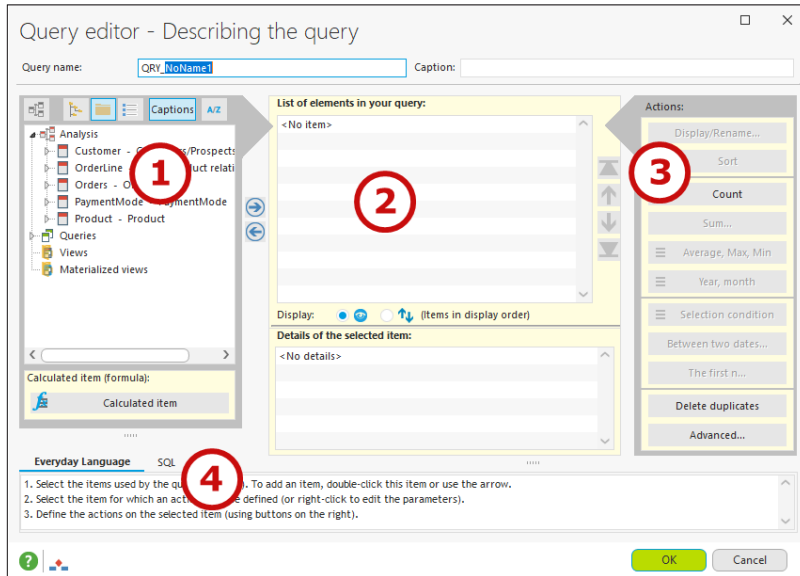
Creating the query used to find orders

Creating the query

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

1. Click among the quick access buttons. The element creation window appears: click "Query". The query creation wizard starts.
2. Select the "Select" option.
Indeed, this query will be used to select records. Go to the next step.
3. The query description window is displayed.

Let's take a look at the interface of query editor:



This interface includes:

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

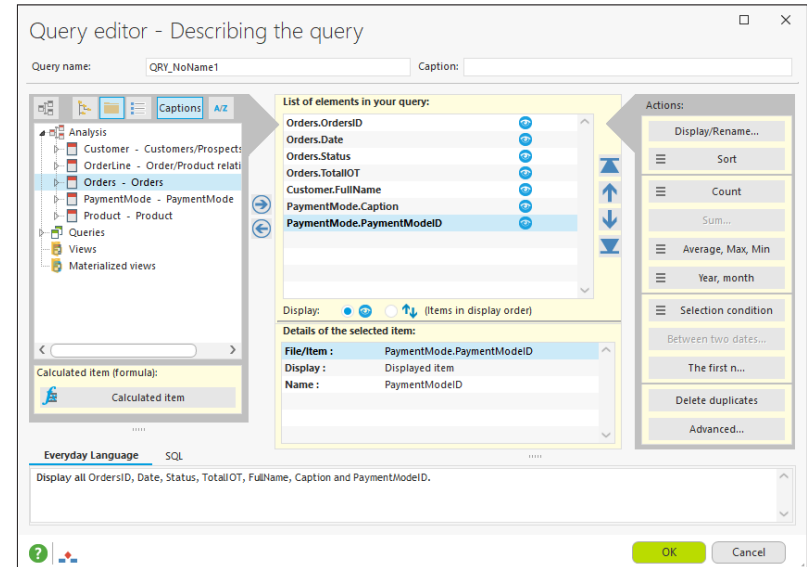
► To build the query, we are going to select the elements that will be displayed in the result.

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

We want to display:

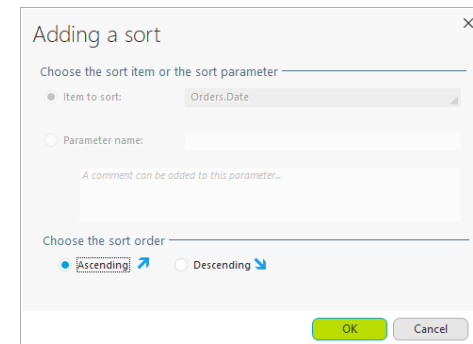
- information regarding the order. In the "Orders" data file, double-click on the items: OrdersID, Date, Status and TotalIOT.
- information regarding the customer who placed the order. In the "Customer" data file, double-click on the "FullName" item.
- information regarding the payment mode of order.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

The description window of query is as follows:



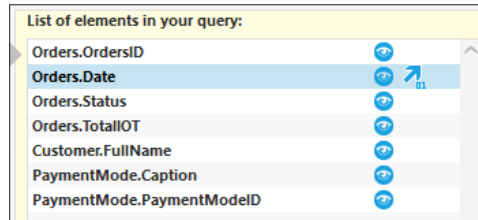
2. The data will be sorted by date.

- Select the "Orders.Date" item in the center of the screen, then click "Sort" (in the list of actions) and select "Sort on the selected item".

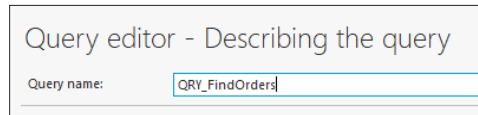


- In the window that is displayed, specify an ascending sort on the item and validate.

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



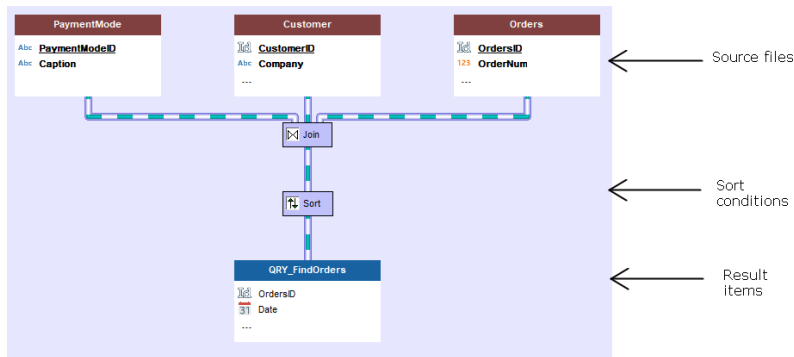
4. Give a name to the query: type "QRY_FindOrders" instead of "QRY_NoName1" in the "Query name" area:



5. Validate the query description window ("OK" button).

6. Validate the window for saving the query.

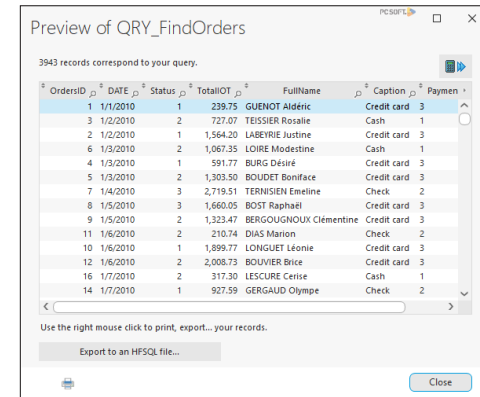
7. The graphic query representation is displayed:



Query test

Like all 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:



Remark

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

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

3. The result lists ALL 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.

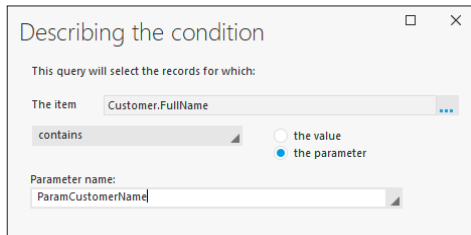
Using parameters in the query

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

- Name of the customer.
- Order status.
- Order payment mode.
- Price of the order.

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

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



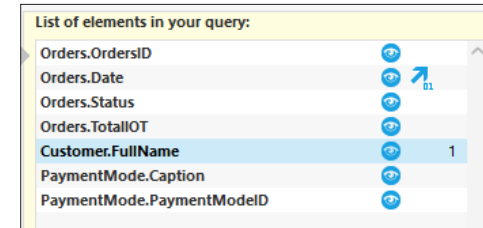
- Select "Contains".
- Check "the parameter".
- Specify the parameter name: "ParamCustomerName".



Remark

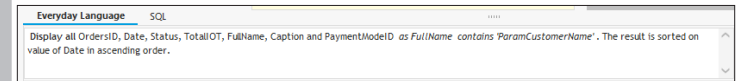
It is recommended to use the prefix "param" on query parameters. This allows you to easily find them in the code editor. When you search for a query parameter, simply enter 'param' and the completion of the code editor will propose all the parameters.

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

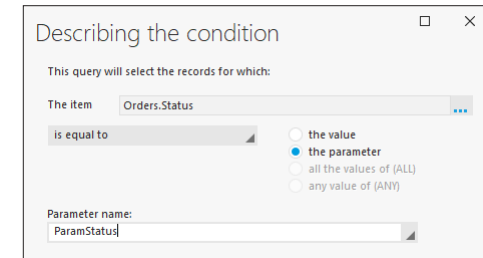


Remark

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



- ▶ Let's now see the second parameter: the order status:
 1. Select the Orders.Status item (in the middle of the screen).
 2. Expand "Selection condition" and select "New condition".
 3. In the window that is displayed, we are going to specify that the selection condition is equal to a parameter:

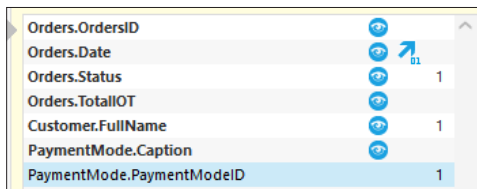


- Select "Is equal to".
- Select "the parameter".
- Specify the parameter name: "ParamStatus".

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

► We are now going to define a condition on the payment mode. The "PaymentMode.PaymentModelID" item must not be displayed in the query result but a condition must be applied to it. To do so, it will be made invisible.

- 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".
- To define a selection condition on the "PaymentMode.PaymentModelID" item:
 - Select the "PaymentMode.PaymentModelID" item (in the middle of the screen).
 - Expand "Selection condition" and select "New condition".
 - In the window that is displayed, specify that the selection condition corresponds to a parameter:
 - Select "Is equal to".
 - Select "the parameter".
 - Specify the parameter name: "ParamPaymentModelID".
- Validate the definition of selection condition.



► The last selection condition to define affects the order amount. In fact, we are going to define two selection conditions in order to specify a minimum price and a maximum price.

- Select the "Orders.TotalIOT" item in the list of query elements.
- Display the popup menu of item (right mouse click) and select "Selection condition .. New condition".
- In the window that is displayed:
 - Select the condition "Is greater than or equal to".
 - Click "the parameter".
 - Specify the parameter name: "ParamMinPrice".
- Validate the definition of selection condition.
- Define once again another condition on the same "Orders.TotalIOT" item: display the popup menu of item (right mouse click) and select "Selection condition .. New condition".
- In the window that is displayed:
 - Select the condition "Is less than or equal to".
 - Click "the parameter".
 - Specify the parameter name: "ParamMaxPrice".
- Validate the definition of selection condition.

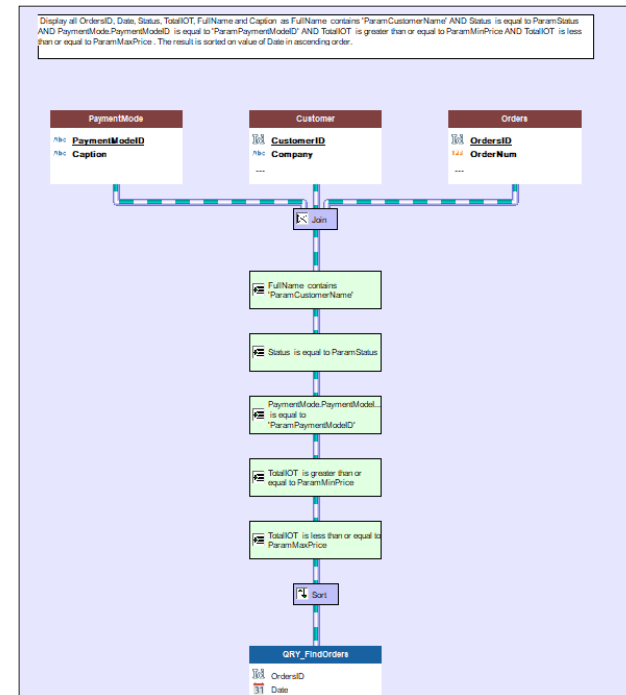


Remark

The definition of the two selection conditions on the item (is greater than or equal to and is less than or equal to) can be replaced by a single condition "Is included between".

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


8. Validate the description window of query. The query graph is modified to take into account the selection conditions that have been defined.



9. Save the query by clicking [Save] among the quick access buttons.

Test of query with parameters

► To run the test of query with parameters:

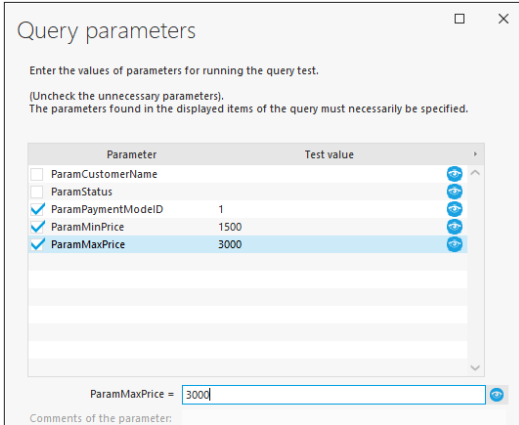
1. Click .
2. A window is displayed, allowing you to type the different query parameters.



Remark

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 ParamCustomerName is ignored, the query will take into account the orders of all customers.

Remark: If all parameters are unchecked, the query returns the entire result.



Parameter	Test value
<input type="checkbox"/> ParamCustomerName	
<input type="checkbox"/> ParamStatus	
<input checked="" type="checkbox"/> ParamPaymentModelID	1
<input checked="" type="checkbox"/> ParamMinPrice	1500
<input checked="" type="checkbox"/> ParamMaxPrice	3000

ParamMaxPrice =

Comments of the parameter:

3. Type the following data:

- Uncheck ParamCustomerName and ParamStatus.
- Select the ParamPaymentModelID parameter. In the lower section of the screen, type "1".
- Select ParamMinPrice. In the lower section of the screen, type "1500".
- Select ParamMaxPrice. In the lower section of the screen, type "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 query result.

Creating the page proposing the multicriteria search

Creating a page

► To create a page used to display the result of multicriteria search:

1. Create a new blank page.
 - Click  among the quick access buttons.
 - The element creation window appears: click "Page" then "Page".
 - The page creation wizard starts.
 - In the "Based on a project template" section, choose "PAGETPL_Single" and finish the wizard.
2. The window for saving the page is displayed.
3. Type the title: "Multicriteria search". The name ("PAGE_Multicriteria_search") is automatically proposed.
4. Validate.

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

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

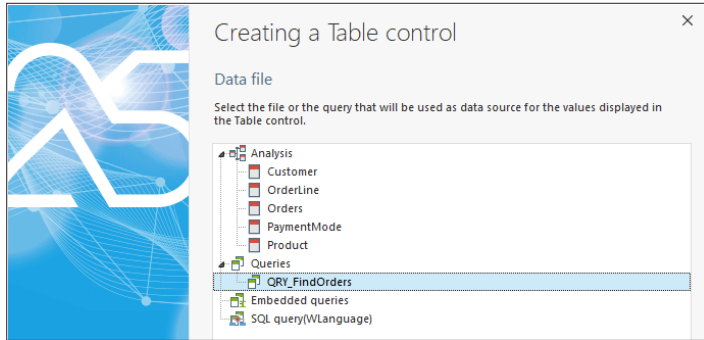
Creating the Table control

► To create the Table control used to display the search result:

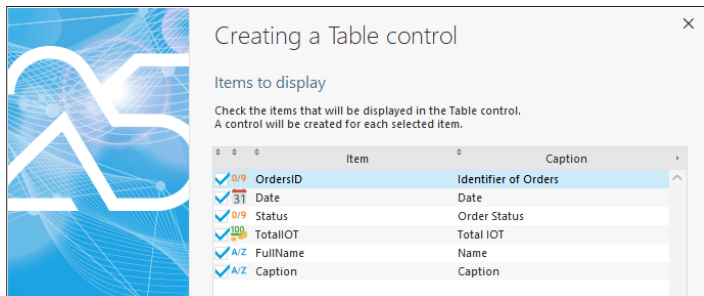
1. Create a Table control: on the "Creation" pane, in the "Data" group, expand "Table" and select "Table".
2. Click in the "PAGE_Multicriteria_search" page: the Table control creation wizard starts.
3. The Table control will be based on the "QRY_FindOrders" query (that was created beforehand). Select "Display the data found in a file or in an existing query". Go to the next step of the wizard.

4. Select the query that will be used as data source for the Table control:

- Expand the "Queries" group.
- Select the "QRY_FindOrders" query.

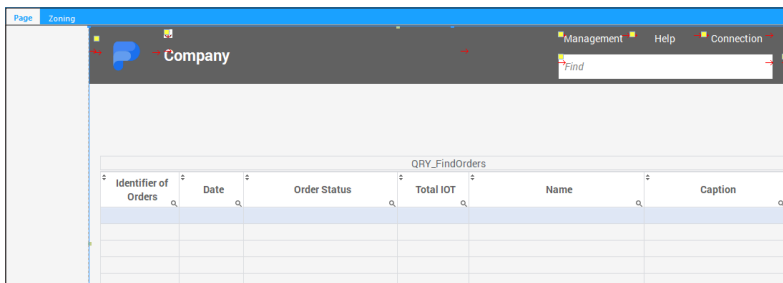


- Go to the next step of the wizard.
- Select all proposed items: all of them will be displayed in the Table control.



Go to the next step of the wizard.

5. Keep the default options in the different wizard screens and validate the creation of Table control.
6. The Table control is automatically created in the page.
7. Modify (if necessary) the position of Table control so that it is entirely displayed in the page.



► For better readability, we are going to modify the Table control description.

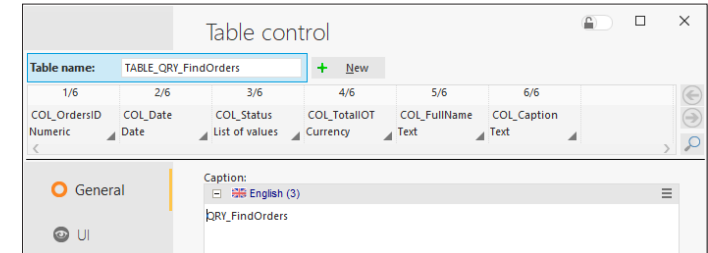
1. Open the Table control description (double-click the control).



Remark

The description window of a Table control includes two sections:

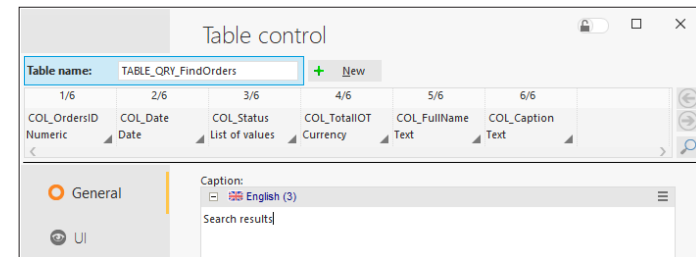
- the upper section, presenting the name of control, the name of columns as well as their type.
- the lower section, containing the different description tabs.



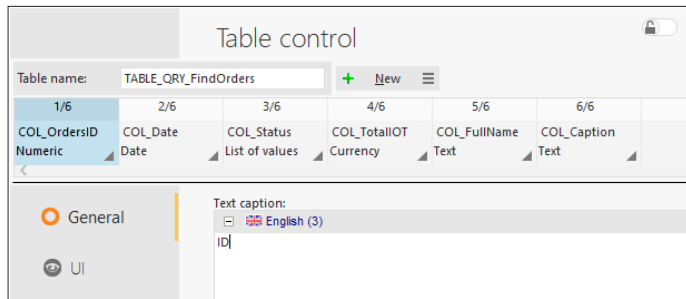
If the name of Table control is selected, the lower section presents the characteristics of Table control.

If a column is selected, the lower section presents the characteristics of columns.

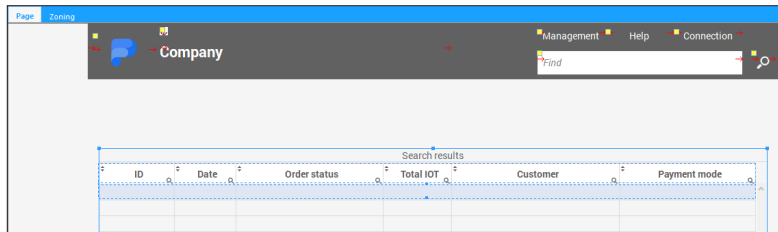
2. Click the name of Table control. The caption of Table control is displayed in the lower section. We are going to modify this caption. Replace the proposed caption by "Search results".



3. Click the "COL_OrdersID" column. The column title is displayed in the lower section of the screen. Replace the "Identifier of Orders" caption by "ID".



4. Click the "COL_FullName" column. Replace the "Name" caption by "Customer".
5. Click the "COL_Caption" column. Replace the "Caption" caption by "Payment mode".
6. Validate the description window of Table control. The control is automatically updated with the modifications performed.
7. Reduce the size of "ID" column in order for all columns to be displayed in the Table control.
8. Increase the size of Customer and PaymentMode columns in the Table control.



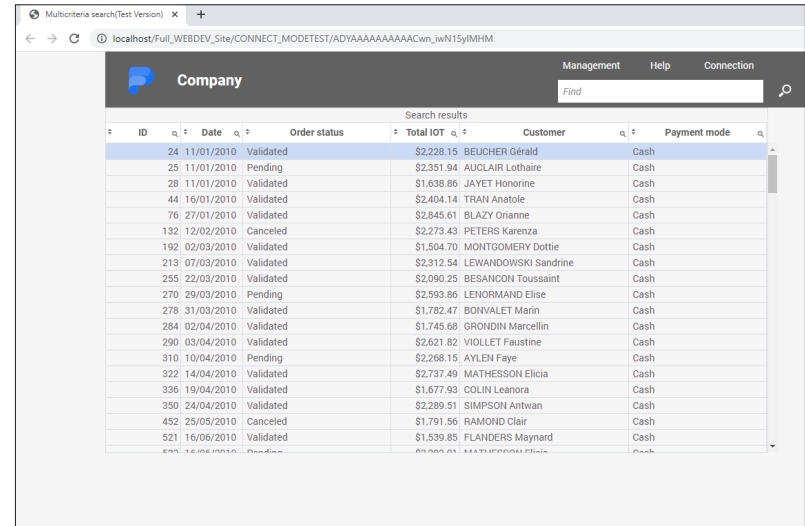
9. Save the page by clicking among the quick access buttons. We are going to check the sizes of columns by running the page.

Remark **Live Data and controls based on queries**

The Live Data is not displayed in the controls that use a query as data source for the following reason: the data displayed in the control depends on the query result therefore it is known at run time only.

- We are going to run a first test of this page:
 1. Click among the quick access buttons.

2. Only orders paid in cash and whose price is found between \$1,500 and \$3,000 are displayed, like in the last query test run in the editor, when the parameters were specified in the test window of query.



3. Close the browser to go back to the editor.

► Let's take a look at the events associated with the Table control:

1. Select the Table control and press the F2 key.
2. The "Initializing TABLE_QRY_FindOrders" event contains the following code:

```
// The control is using a query with parameters to display the data.
// The parameters of this query must be defined before or
// when initializing the control.
// The query will be automatically run if at least one parameter is
// defined.
//
// See the online help for more details:
// Query with parameters, Use in a Table, List Box or Combo Box
// control
//
// Parameters of 'QRY_FindOrders' query
//MySource.ParamCustomerName = <Value of ParamCustomerName>
//MySource.ParamStatus = <Value of ParamStatus>
MySource.ParamPaymentModeID = "1"
MySource.ParamMinPrice = "1500"
MySource.ParamMaxPrice = "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 typed by the user, via controls.

3. To build our page, we are going to switch all 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 in the top right corner of 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.

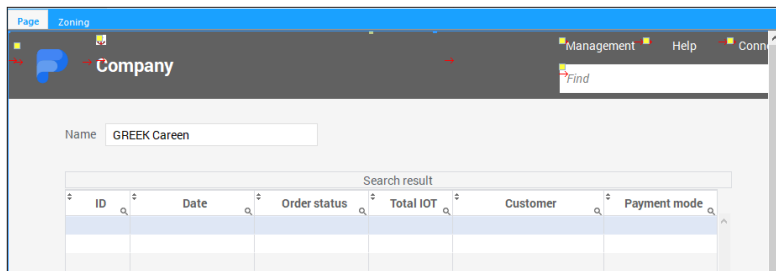
- ▶ Move (if necessary) the Table control in the page and reduce its height in order to leave space for creating the different controls for criteria selection.

First parameter: Customer name

In order for the user to be able to type a customer name to find, we are going to create an edit control.

- ▶ To create the edit control:

1. Display the "Analysis" pane if necessary: on the "Home" pane, in the "Environment" group, expand "Panels" and select "Analysis". The different data files described in the "Full_WEBDEV_Site" analysis are displayed in the pane.
2. Click the  icon on the left of the "Customer" data file: the items found in the data file are listed.
3. Select the "FullName" item located in the Customer data file, then drag it and drop it in the "PAGE_Multicriteria_search" page.
4. The edit control is automatically created. Position this control above the Table control.



- ▶ We are now going to pass the value typed in the edit control in parameter to the query:

1. Select the Table control and press the F2 key.
2. In the "Initializing" event of the Table control, replace the line:

```
//MySource.ParamCustomerName = <Value of ParamCustomerName>
```

by:


```
MySource.ParamCustomerName = EDT_FullName
```

In this code, EDT_FullName is the name of edit control that was just created. The value of this control is associated with the ParamCustomerName parameter expected by the query.

3. Close the code editor.

- ▶ Before running the test, we are going to create a Button control to re-display the content of Table control according to the value selected in the Radio Button control:

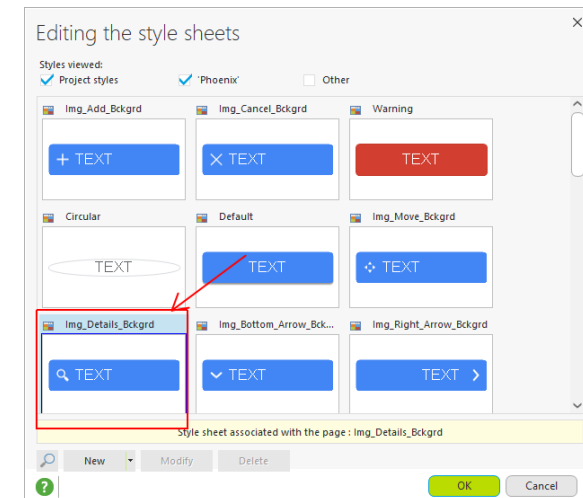
1. Create a Button control:

- on the "Creation" pane, in the "Usual controls" group, click .
- then, click at the top right of the page.

2. Modify the control caption (press Enter on the keyboard). The new caption is "Find".

3. Modify the control style:

- In the popup menu of control (right mouse click), select "Choose a WEBDEV style".
- In the window that appears, select "Details (With background)".



- Validate ("OK" button).

4. Resize the control if necessary:

- select the control.
- select "Adapt the size" from the popup menu.

5. Display the code associated with this control: press F2.

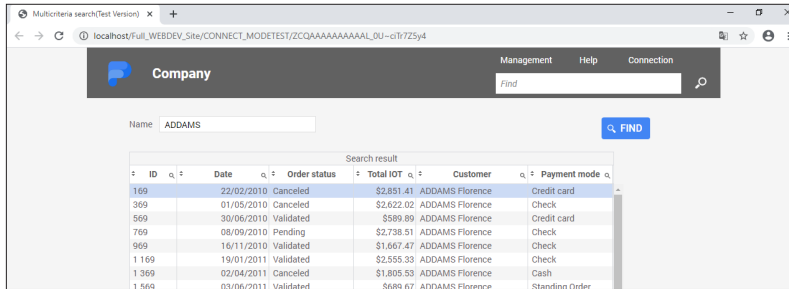
6. Write the following code in the "Click (server)" event:

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

In this code, the *taInit* constant is used to re-run the "Initializing" event of the Table control (the event 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. Save the page by clicking among the quick access buttons.
 2. Click among the quick access buttons.
 3. In the page that is displayed, type the customer name ("ADDAMS") and click "FIND". The content of Table control is modified: all orders placed by the customer named "ADDAMS" are displayed.



4. Close the test page.

Second parameter: Order status

Three states can be assigned to an order:

- pending,
- paid,
- canceled.

In our analysis, the order status is saved in the "Status" item found in the "Orders" file. This item is a radio button.

To allow the user to select one of these three states, we are going to use the Radio Button control associated with the "Status" item of "Orders" data file.



Remark

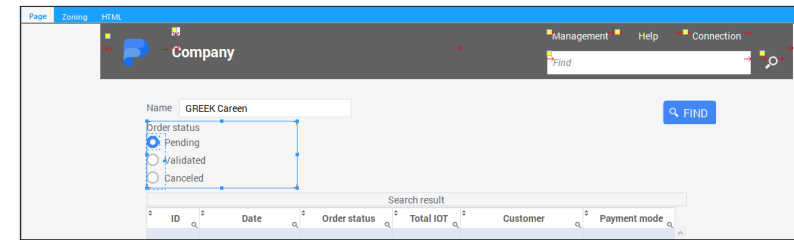
The radio buttons are also called "option boxes". They are used to select a single option among the proposed ones.

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 next to the "Orders" data file: the items found in the data file are listed.
 2. Select the "Status" item in the Orders data file, then drag and drop this item to "PAGE_Multicriteria_search".

- 3. The Radio Button control is automatically created. Position this control above the Table control.



- ▶ We are now going to use the value selected in the Radio Button control to pass it to the query as parameter:

1. Display the events associated with the Table control:
 - Select the Table control.
 - Display the popup menu (right mouse click) and select "Code".
2. In the "Initializing" event of the Table control, replace the line:

```
//MySource.ParamStatus = <Value of ParamStatus>
```

by:

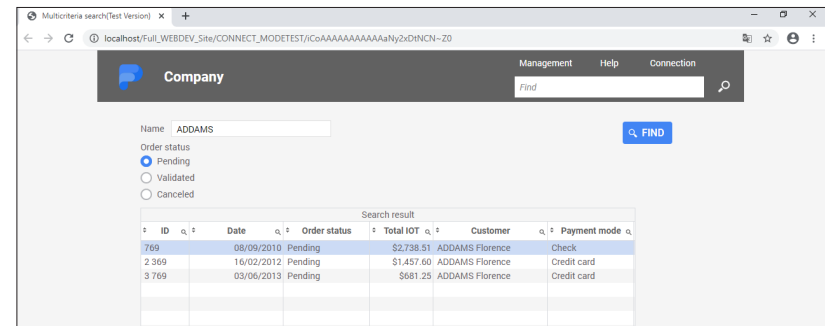
```
MySource.ParamStatus = RADIO_Status
```

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 ParamStatus parameter expected by the query.

- 3. Close the code editor.

- ▶ We are now going to check how the first two parameters are passed:

1. Click among the quick access buttons.
2. Type the name "ADDAMS" and select "Pending".
3. Click "FIND": only the orders placed by the customer named ADDAMS and waiting for payment are listed.



- ▶ In this example, only the orders placed by a customer and in a specific status can be listed. It may also be interesting to list all orders placed by a customer regardless of their status. We are going to modify our page in order to perform this process.

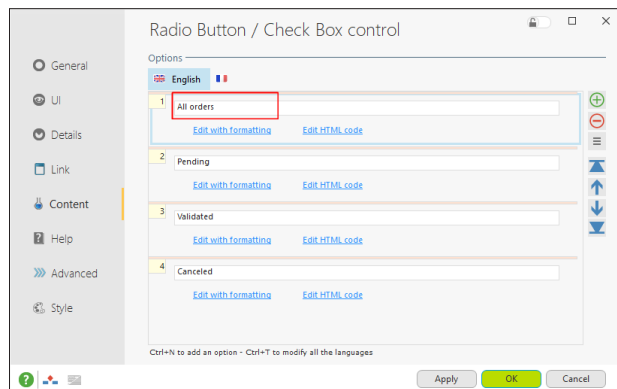
To manage this case, we must:

- Add an additional status to display all orders,
- Manage the additional status.

- ▶ Close the browser.

- ▶ To add an option into the Radio Button control:

1. Select the Radio Button control that was created beforehand.
2. Open the control description window (double-click the control for example).
3. In the "Content" tab, in the list of options, insert the "All orders" option at the top of the list:
 - Click the "+" button.
 - Type "All orders".
 - Use the arrow pointing to the top (on the right) to move the option up.



4. Validate the control description window.
5. Enlarge the control in the editor in order for all options to be displayed (move the Table control if necessary).

- ▶ The new option "All orders" means that the query parameter "ParamStatus" will be ignored. To ignore a query parameter, simply assign the NULL value to this parameter. We will manage this case for ParamStatus.

1. Select the Table control in the editor and display its events (press F2 or select "Code" from the control popup menu).
2. In the "Initializing" event of the Table control, replace the line:

```
MySource.ParamStatus = RADIO_Status
```



by:

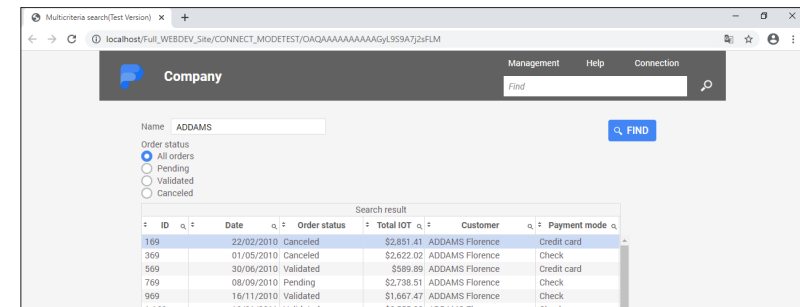
```
SWITCH RADIO_Status
// All orders
CASE 1
    MySource.ParamStatus = Null
OTHER CASE
    MySource.ParamStatus = RADIO_Status-1
END
```

In this code, if the RADIO_Status control corresponds to 1 (case of "All orders" option), the parameter of 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. Save the page by clicking  among the quick access buttons.
2. Click  among the quick access buttons.
3. Type the name "ADDAMS" and select "All orders".
4. Click "FIND": all orders placed by the customer named ADDAMS are listed, regardless of their status.



5. Close the browser.

Third parameter: Payment mode

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

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



Remark

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.

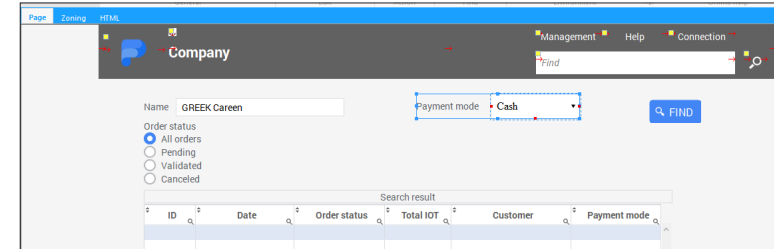
The elements displayed in a combo box can be defined when creating the control in the editor. These elements:

- 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 found in a file or in an existing query" and go to the next step.
5. Select the "PaymentMode" data file and go to the next step.
6. The item that will be displayed in the Combo Box control is "Caption". Select "Caption". Go to the next step.
7. Select the search key: "Caption". Go to the next step.
8. Select the return value: "PaymentModeID". This return value is very important because it will be passed in parameter to the query. Go to the next step.
9. Keep the default options in the different wizard steps and validate the creation of Combo Box control.
10. The Combo Box control is automatically created in the page.

► Modify the caption of Combo Box control (press Enter). 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 events associated with the Table control: select the Table control and press F2.
2. In the "Initializing" event of the Table control, replace the line:

```
//MySource.ParamPaymentModeID = 1
```

by:

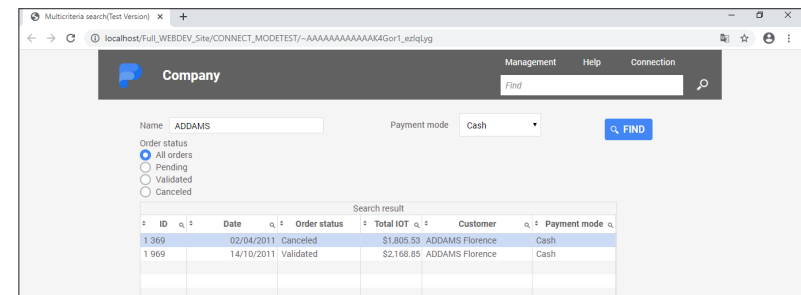
```
MySource.ParamPaymentModeID = COMBO_PaymentMode
```

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 ParamPaymentModeID 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. Type the name "ADDAMS".
3. Change the status of orders via the radio button and modify the payment mode via the Combo Box control, then click the "Find" button. The content of Table control is modified.



4. Close the browser.

Last parameter: Total amount of order

The last query parameter corresponds to the order amount taken into account. We have a parameter that corresponds to the minimum amount and a parameter that corresponds to the maximum amount. The interface must allow the user to type a price interval. To do so, we will be using a Range Slider control.

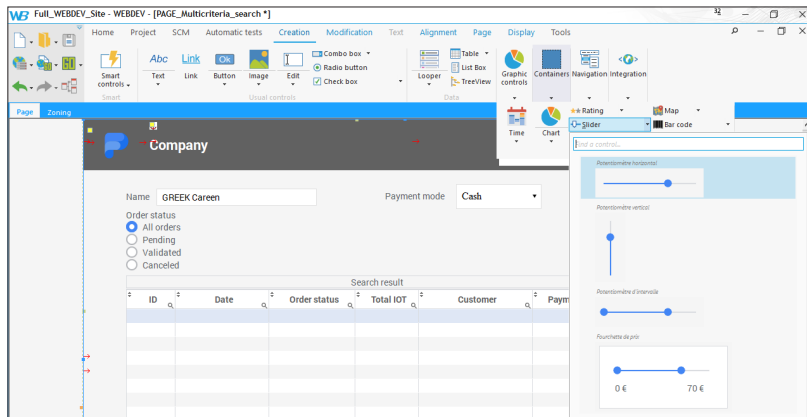


Remark

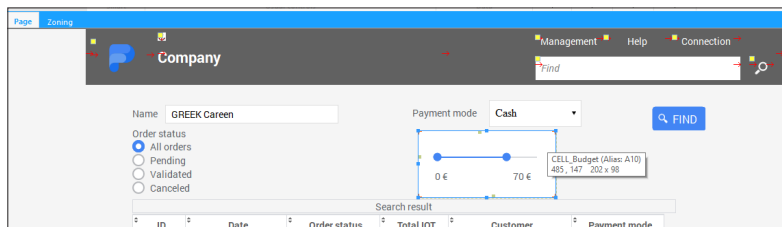
A Range Slider control is a graphic control that can be easily included in a WEBDEV site and that is used to select the bounds of an interval (start value and end value).

► To manage the product price:

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



2. Click inside the page, between the "Payment mode" control and the "FIND" button. The range slider is automatically created.



► The control that was just created is an advanced control that includes the slider and two controls displaying the bounds. The programming required to display the bounds is already performed in the control. We are going to initialize the Slider control so that it proposes the bounds corresponding to the data found in the Orders data file.

1. Select the Slider control and display its events (F2).
2. Insert the following code (before the existing code lines) into the "Initializing" event 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" data file according to the browse 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 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 greatest amount. The only difference is the name of the function used: **HReadLast**. This function will allow you to read the last record of the Orders data file according to the amount, i.e. the record with 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 events associated with the Table control: select the Table control and press the F2 key.
2. In the "Initializing" event of the Table control, replace the lines:

```
//MySource.ParamMinPrice = "1500"
//MySource.ParamMaxPrice = "3000"
```

by:

```
MySource.ParamMinPrice = SLIDER_Price..LowerValue
MySource.ParamMaxPrice = SLIDER_Price..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).

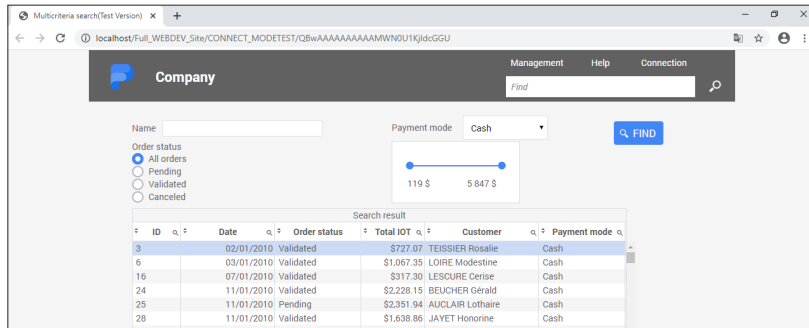
► We are now going to check how the parameters are passed:

1. Click among the quick access buttons.

2. Define the different search parameters:

- Customer name,
- Status,
- Payment mode,
- Amount.

3. Click "FIND". The content of Table control is modified.



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 Table control will be returned by the server.



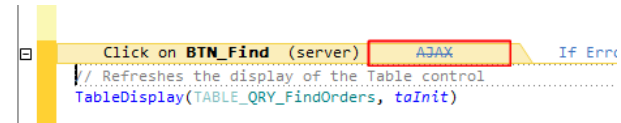
Remark

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

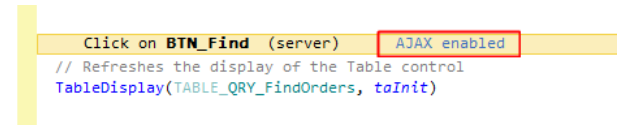
- **Ajax in one click:** simply click in the code editor to transform an event into Ajax event.
- **Ajax controls** (Table control, Looper control). The Ajax controls are used to display data dynamically loaded from the server. No need to download everything at once!
- **Ajax by programming** via specific WLanguage functions: **AJAXExecute** and **AJAXExecuteAsynchronous**.

► To implement the Ajax mode on the "Find" button:

1. Select the "Find" button and display its events (F2).
2. In the "Click (server)" button event, the AJAX link appears to be struck out in the code bar.



3. Click the AJAX link: the "AJAX enabled" link appears, indicating that the event is automatically changed into an Ajax event.



4. Close the code editor.

► Run the page test:

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

Overview

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



Answer

If you have not created the pages in the previous lessons, you can still follow the report creation section by opening a corrected project: in the WEBDEV home page (Ctrl + <), click "Tutorial" and select "Full WEBDEV Site (With pages)". A full corrected project is also available: in the WEBDEV home page (Ctrl + <), click "Tutorial" and select "Full WEBDEV Site (Answer)".

Principle for printing on Internet

In fact, we should not talk about "printing" on Internet. Indeed, when a document must be "printed", a file is generated beforehand (in HTML, PDF or XML format) and it is transmitted to the browser. As soon as the file transfer is completed, the Web user decides whether the document that was received will 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 sales department.

Generating documents in HTML, PDF or XML format

Printing from HTML, PDF or XML documents consists in creating a document and in displaying it on the browser of Web user. To create this document, all you have to do is use the WEBDEV report editor. A formatted file can be "generated" by your WEBDEV site from a data source. The file can be in HTML, PDF, RTF or XML format.

The main benefit of this method is that the generated file can be transmitted to the browser. The Web user can print the document or store it on his computer.

Printing in a file can be used for an Internet site and for an Intranet/Extranet site (for example: to transmit an order form in PDF format to the Web user, ...).

What is a report made of?

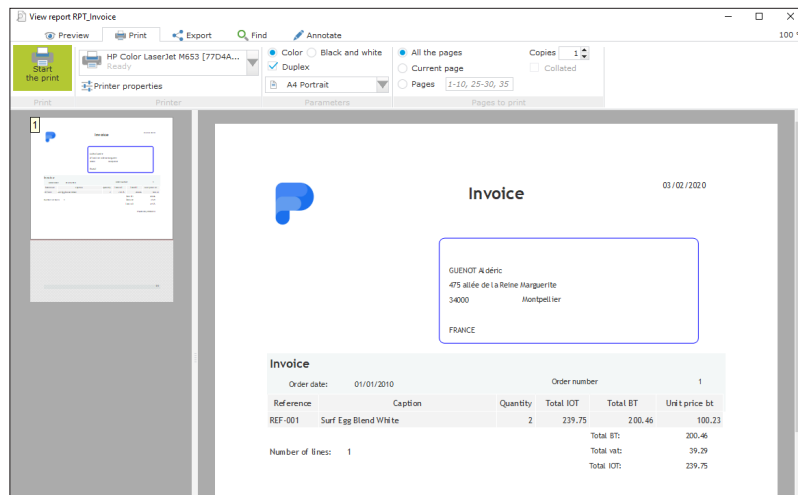
In our example, the invoice that must be printed will correspond to a report. A report is a project element used to group and format the data to print.

To create and use reports, WEBDEV proposes a specific editor: the report editor.

A report includes:

- different blocks. The blocks are used to define the areas where the data will be displayed. The available blocks are as follows:
 - Start of document: block displayed on the first page only.
 - Page header: block displayed at the top of each page.
 - Body: block containing the data. This block is printed as long as there is data to print.
 - Page footer: block displayed at the bottom of each page.
 - End of document: block displayed on the last page only.
- controls used to display the data.

Let's see the report that we want to create:



Creating the "Invoice" report

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

- The order characteristics: date and order number.
- The personal customer details: name, address, zip code, city and country.
- The characteristics of order lines:
 - Ordered quantity,
 - 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 WEBDEV project (Table control, Looper control, ...).





Remark

WEBDEV proposes to create reports from several data sources: data files, queries, controls, text files, ...

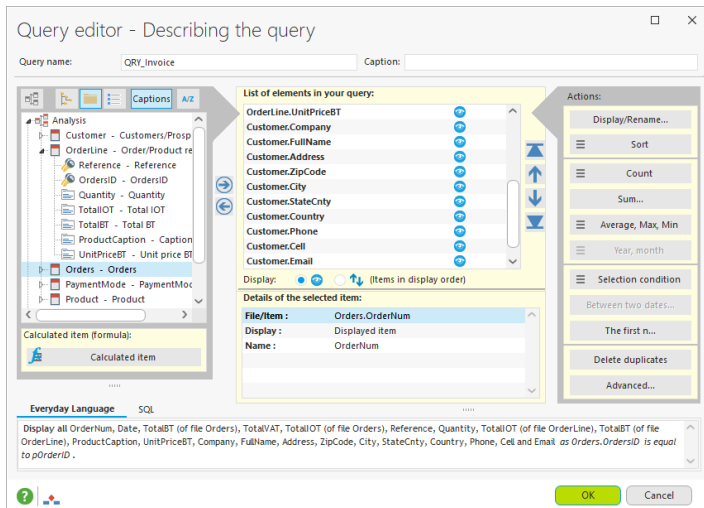
In most cases, we advise you to group the data to print via a query and to create a report based on this query. To add an information into the report, all you have to do is add the corresponding item into the query.

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 report.
 1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Query". The query creation wizard starts.
 2. Select the "Select" option. 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 query is displayed.
- ▶ First of all, give a name to the query: type "QRY_Invoice" instead of "QRY_NoName1" in "Query name".
- ▶ To build the query, we are going to select the elements that will be displayed in the result. The query will contain the content of Orders data file, the content of OrderLine data file and the content of Customer data file.
 1. In the left section of the screen, select the Orders data file and click the blue arrow (): the items of Orders data file are displayed in the middle of the screen.
 2. Repeat this action for the OrderLine and Customer data files.

The description window of query is as follows:




At this time, this query is used to select all orders and the corresponding order lines. We are now going to reorganize the query items. Indeed, this order will be used to create the different controls of the report based on this query. We want the product caption to be positioned just after the product reference.

- ▶ To reorganize the order of items:
 1. Select the ProductCaption item of the OrderLine data file.
 2. Click  on the right of the list of items. The item moves up.
 3. Position the item just after the product reference (OrderLine.Reference).

We want to select the data corresponding to a single order whose identifier is known. Therefore, we are going to define the order number in parameter.

- ▶ To manage the "Order identifier" parameter:
 1. Select the Orders.OrdersID item (in the middle of the screen).
 2. Click in the fourth column: select "New condition".



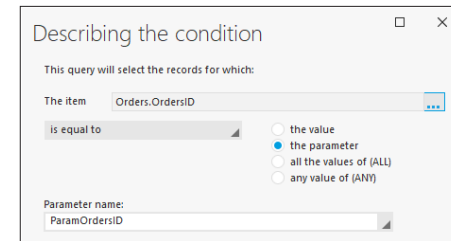
Remark

When describing a query, the section that lists the query elements includes 4 columns:

- The item name,
- The ability to display (or not) the item in the query result,
- The sort management for the item,
- The number of conditions associated with the item.

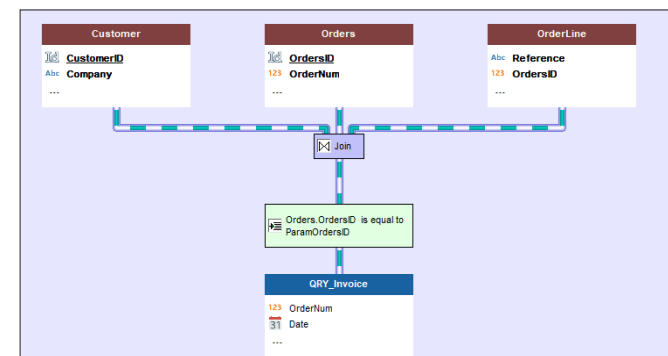
To access one of these characteristics, all you have to do is click in the corresponding column.

3. In the window that is displayed, we are going to specify that the selection condition corresponds to a parameter:



Perform the following operations:

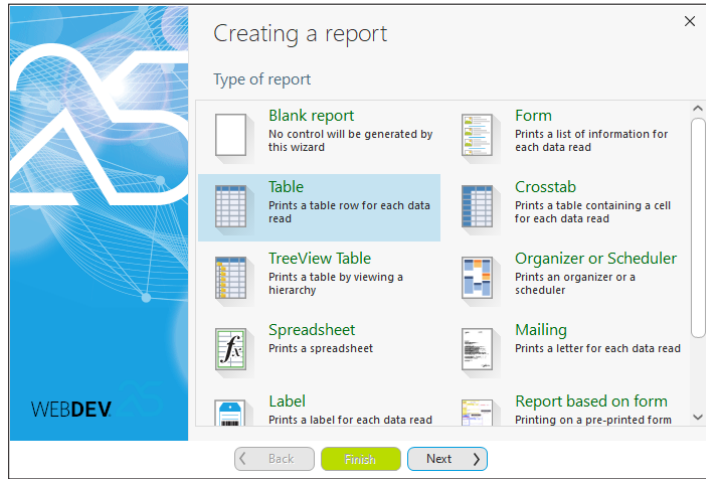
- Select "Is equal to".
 - Check "the parameter".
 - Specify the parameter name: "ParamOrdersID".
4. Validate the condition description window. The number "1" is displayed on the right of Orders.OrdersID item, indicating that a selection condition was defined.
 5. The order identifier will not be viewed so let's make it invisible: click the eye found on the item line and select "Don't display".
 6. Similarly, make the following items invisible:
 - Orders.Status,
 - Orders.CustomerID,
 - Orders.PaymentModelID,
 - OrderLine.OrdersID,
 - Customer.CustomerID.
 7. Validate the query description window ("OK").
 8. Validate the save information
 9. The graphic query representation is displayed:



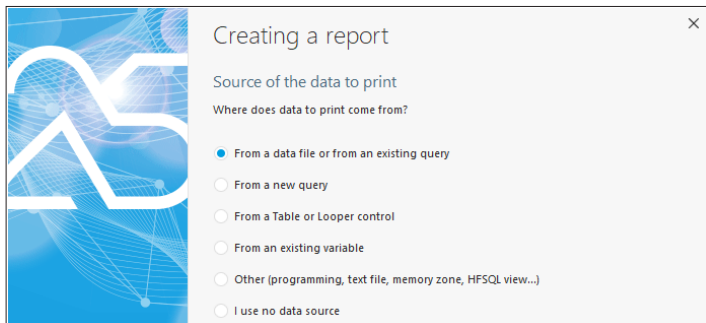
Creating the report based on a query

► To create a report:

1. Click among the quick access buttons.
2. The window for creating a new element is displayed: click "Report" then "Report". The report creation wizard starts.
3. The report creation wizard proposes several types of reports:



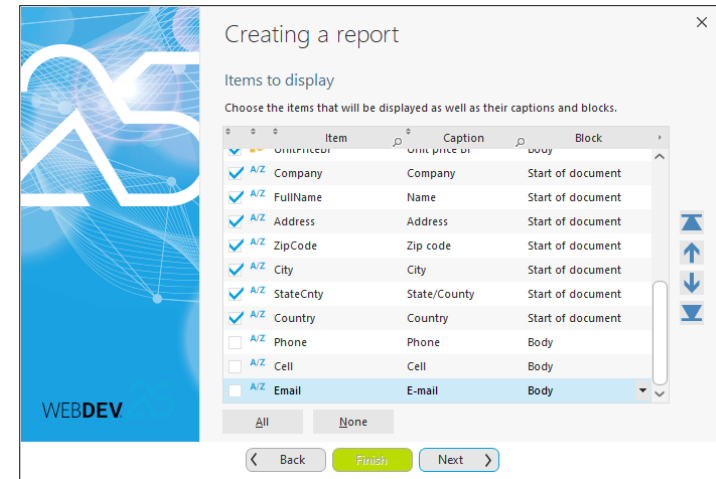
4. Select "Table". Go to the next step.
5. Select the data source of report. The report will be based on the query that was just created. Select "From a data file or from an existing query". Go to the next step.



6. In the list of data files and queries, select the "QRY_Invoice" query. Go to the next step.
7. The wizard asks you to specify whether a break is required. No break will be used in this report. This concept will be presented later in this tutorial. Answer "No". Go to the next step.

8. You are going to specify the order in which the items will be printed and how they will be distributed in the different blocks:

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



The following table presents the different assignments of items in the order presented in the wizard:

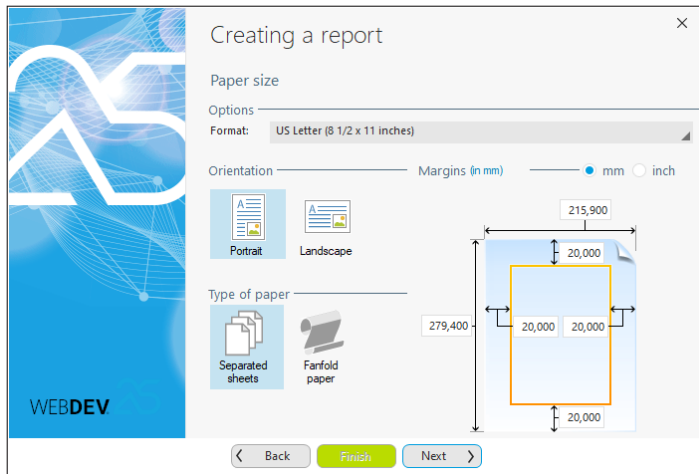
Item	Block
OrderNum	Start of document
Date	Page header
TotalBT	End of document
TotalIVAT	End of document
TotalIOT	End of document
Reference	Body
Quantity	Body
TotalIOT_Or	Body
TotalBT_Or	Body
Product caption	Body
UnitPriceBT	Body

Company	Start of document
FullName	Start of document
Address	Start of document
ZipCode	Start of document
City	Start of document
StateDep	Start of document
Country	Start of document
Phone	Uncheck
Cell	Uncheck
Email	Uncheck

9. Go to the next step.

10. The wizard proposes to create a counter, a sum or an average 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 step is used to define the report layout.



We will keep the default values with the "Portrait" orientation.



Remark

Print margins

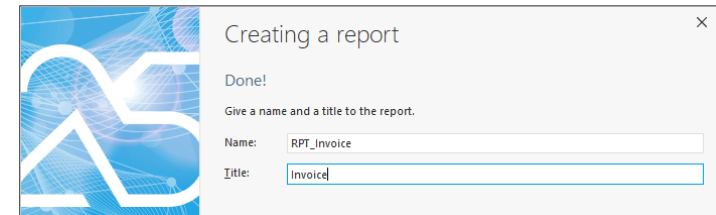
When choosing the print margins, don't forget to take into account the physical margins of printers. The physical margins of printers are margins where no print is allowed. Furthermore, the physical margins differ according to the printer models.

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 "Phoenix" 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.

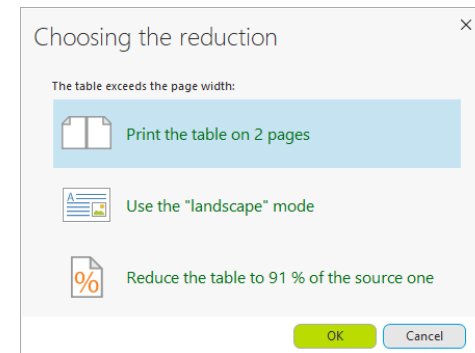
- Type the name: "RPT_Invoice".
- Type the title: "Invoice".



15. Validate.

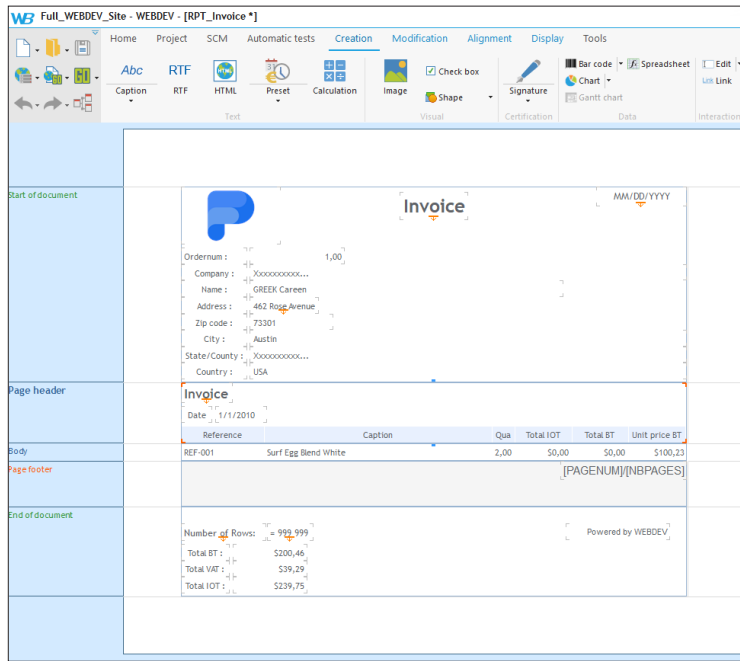
16. The table as it is defined does not fit on an A4 page in portrait mode. WEBDEV proposes to:

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



Choose "Reduce the table".

- 17. Validate the save information.
- 18. The report is displayed in edit in the report editor:



- 19. The different order lines are grouped in a table.

► Run this report by clicking among the quick access buttons.

- 1. The report editor asks for the print destination. The print destination can be:
 - Report viewer: the report is displayed on the screen in a specific window.

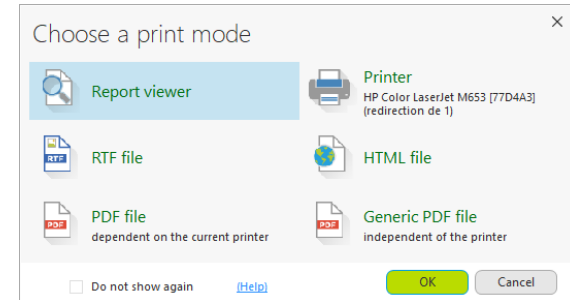


Remark

The report viewer is only available when developing the site. At runtime, the user will not be able to use the report viewer. The only available preview will be the display of generated file by the browser.

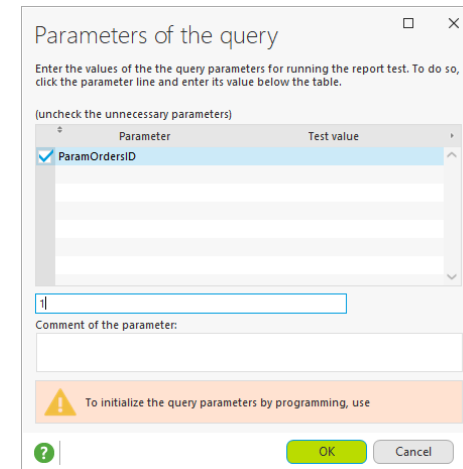
- Printer: the report is printed on the default printer.

- Specific file: the report is printed in a file to the selected format.



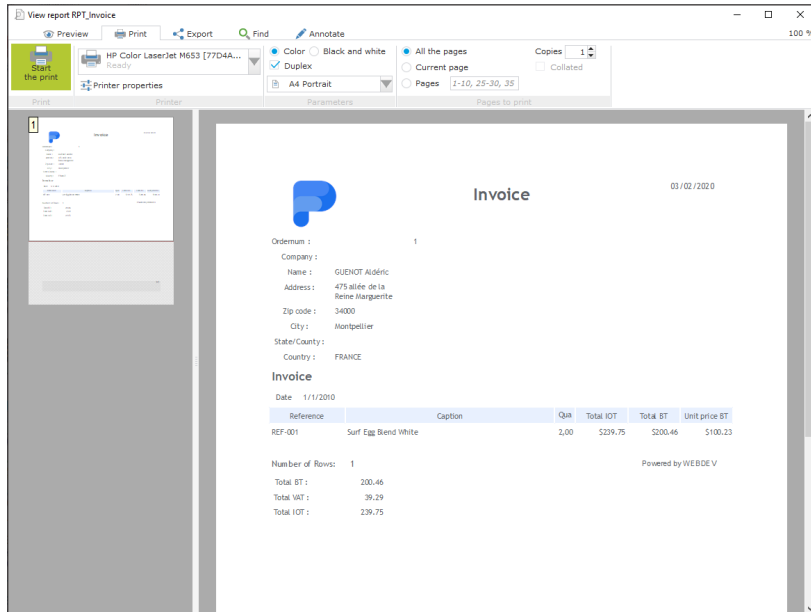
Select "Report viewer" and validate.

- 2. The report editor asks for the parameters of query used by the report. Don't forget that a parameter was used to specify the number of the order to print. For the example, type the test value "1".



Validate.

3. The report is displayed on the screen.



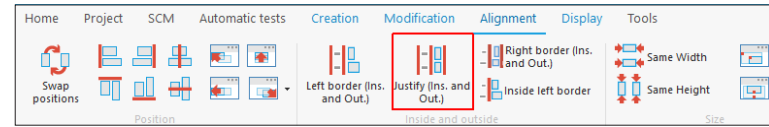
4. Close the report viewer.

Modifying the "Invoice" report

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

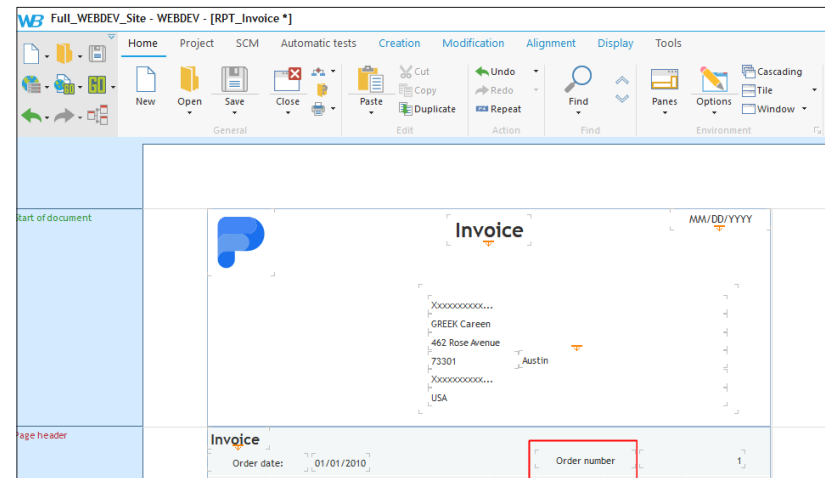
- ▶ We want to position the information regarding the customer and the order in the page header:
 1. Delete the captions found in front of the customer details (Name, ...).
 2. Position the item containing the city beside the zip code.
 3. Enlarge (with the handles) the control containing the company name: the control size must be identical to the size of Zip Code and City controls.
 4. Align the controls:
 - Select the Company control.
 - Press Ctrl and select (with the mouse) the controls containing the address, the state and the country.

• On the "Alignment" pane, click "Justify (Ins. and Out.)".



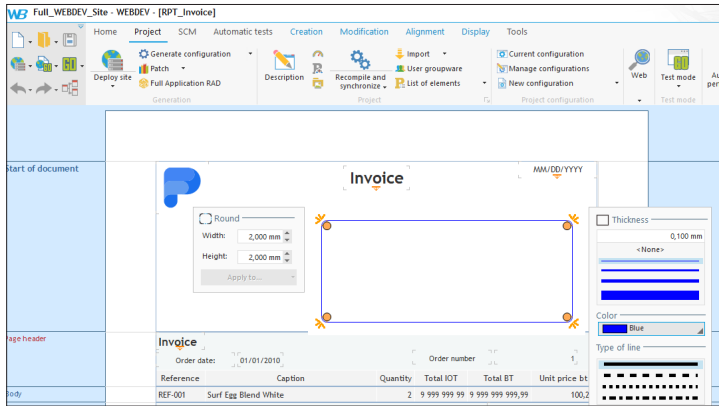
5. Select the customer details and move them (with the mouse) to the right of report.

6. Move the order number: position it at the top of the "Page header" block.



- ▶ We are going to create a border around the customer details. To do so, we are going to create a Static control stacked over the customer details. Only the border of this static will be visible.
 1. Create a Static control: on the "Creation" pane, in the "Text" group, click "Static".
 2. Click inside the "Start of document" area, 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 is stacked over the customer details. This operation is used to enlarge the Static control without moving the controls found below (containing the customer details).
 4. In the popup menu of Static control, select "Edit the caption" and delete the caption. Click in the report to validate the modification.

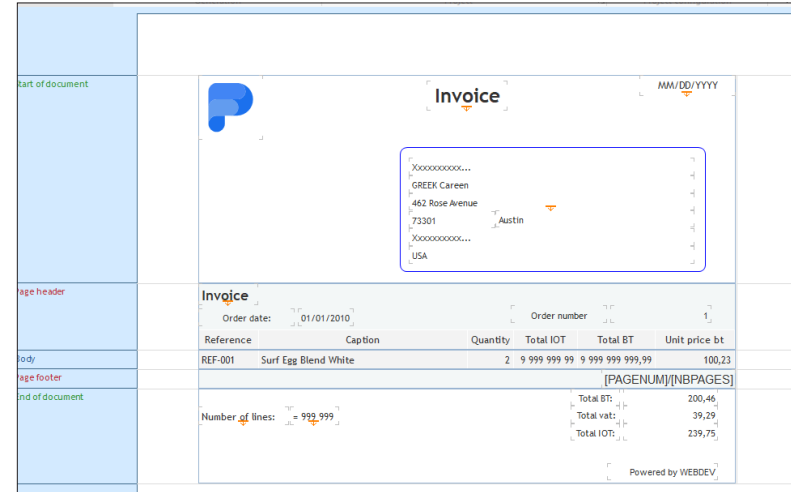
- ▶ To configure the border:
 1. Select the Static control that was just added.
 2. Press R and click at the top right corner of the control.
 3. Select "Edit the border". The configuration options are displayed:



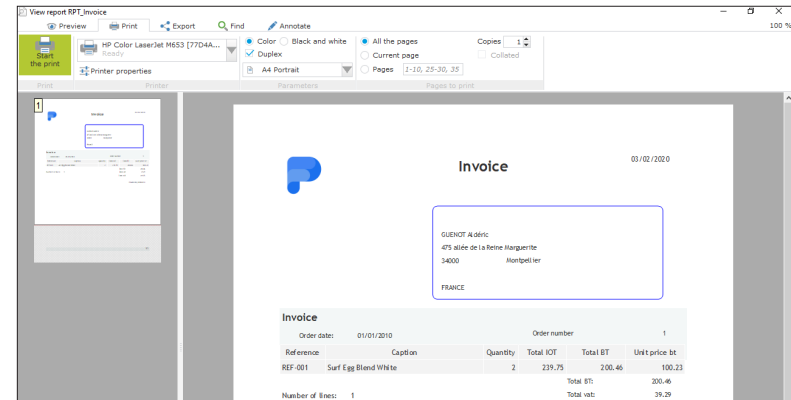
4. Select:
 - the "Blue" color in the context-sensitive colors.
 - a simple border (thin line).
 - a rounding set to 2 mm in width and in height.

- ▶ 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 at the bottom right corner of the table (above the text "Powered by WEBDEV").

3. The report is displayed in the report editor:



4. Our "Invoice" report is created. Save the report by clicking among the quick access buttons.
5. Check the changes made by running the report in "Report Viewer" mode (click among the quick access buttons).



6. Close the report viewer.

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 report will be printed in PDF on the server and the generated PDF document will be downloaded or displayed on the Web user's computer. Then, the user will be able to print the document on their printer.

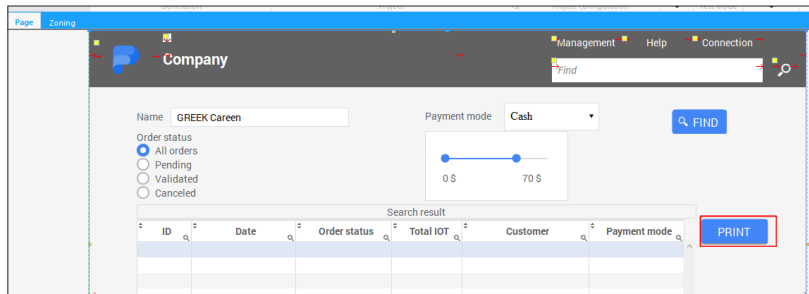
In our site, "RPT_Invoice" will be printed from a Button control in the page used to find an order. This Button control will print the invoice for the selected order.

Remark: To complete this part of the tutorial, the page "PAGE_Multicriteria_Search" must have been created in Lesson 5.1. Multicriteria search.

Implementing the print

To print the "RPT_Invoice" report:

1. Locate the "PAGE_Multicriteria_Search" page: click "PAGE_Multicriteria_Search" in the documents 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 control.
4. Select the control and press Enter. The caption becomes editable.
5. Replace "Button" by "Print" and validate.




6. Open the code of this button (F2) and write the following code in the "Click (server)" event:

```
iDestination(iGenericPDF)
iInitReportQuery(RPT_Invoice, ...
    TABLE_QRY_FindOrders.COL_OrderID[TABLE_QRY_FindOrders])
iPrintReport(RPT_Invoice)
FileDisplay(iLastFile(), "application/pdf")
```

Let's study this code:

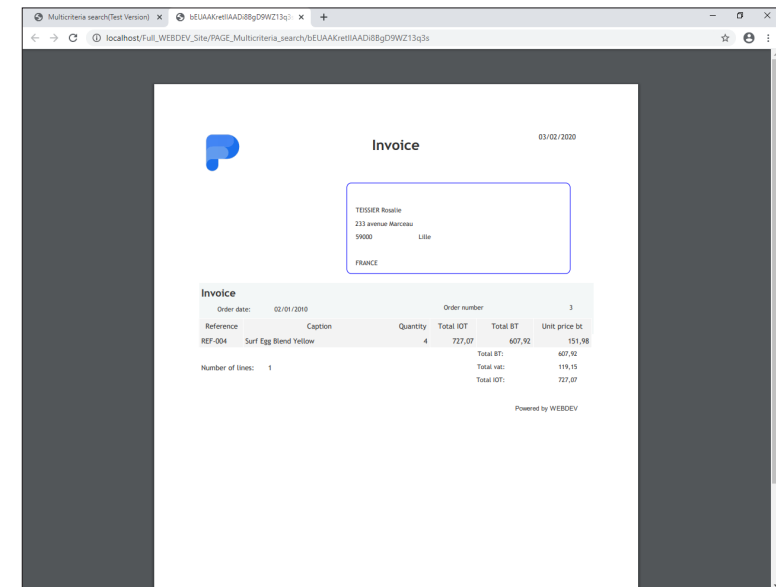
- **iDestination** is used to define the print format (PDF format in this case).
- The "RPT_Invoice" report being based on a query with parameters, the parameter must be passed to the query before running the report. This operation is performed by **iInitReportQuery**. In our case, the parameter corresponds to the number of the current order, displayed in the Table control.

- **iPrintReport** is used to generate the "Report_Invoice" report in the specified format (PDF in our case).
 - **FileDisplay** is used to display the invoice to the user:
 - **iLastFile** returns the path of the last file generated by a report.
 - "application/pdf" if the MIME type of returned file. When this type is specified, the browser can display the file directly or it can choose the application that will be used to perform the display on the computer of Web user.
7. Close the code editor.
 8. Save the page ( or Ctrl + S).

Print test

► Now, all we have to do is run a real test:

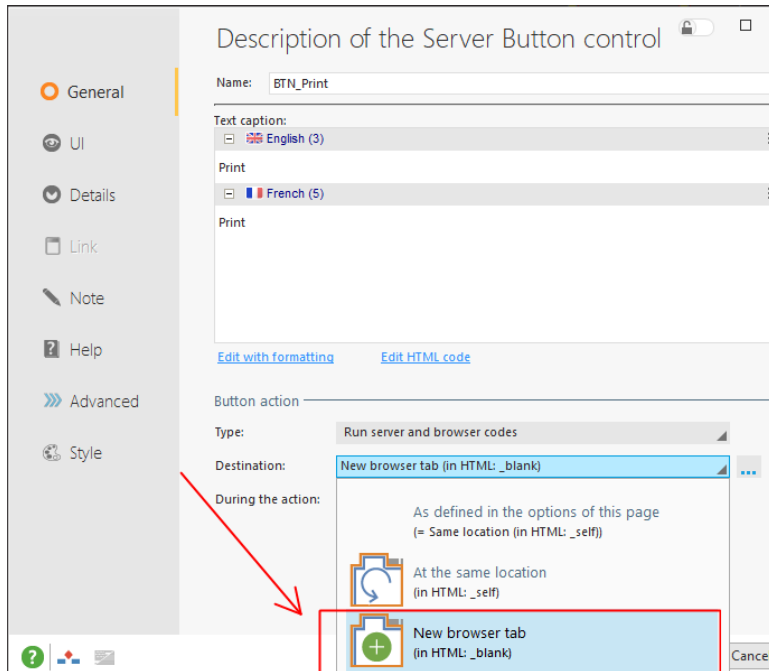
1. Run the test of "PAGE_Multicriteria_Search" page.
2. Specify the criteria and start a search.
3. Select one of the orders displayed in the Table control.
4. Print the order via the "PRINT" button.



5. The browser opens the PDF file. The browser has replaced the current page by the PDF file. We are going to change this behavior.
6. Close the browser.

► To open the PDF file in another tab or browser:

1. Double-click the "Print" button in the "PAGE_Multicriteria_Search" page. The description window is displayed.
2. In the "General" tab, in the Destination control, select "New browser tab".



3. Validate.



Remark

New tab or new window?

The display in a new tab or in a new window can be chosen neither by the user nor by the site developer. It is the browser that chooses whether a new tab or a new window is opened. The behavior can differ according to the browser used!

LESSON 5.3. SENDING AN EMAIL

This lesson will teach you the following concepts...

- How to send an email from a WEBDEV site.



Estimated time: 20 mn

Overview

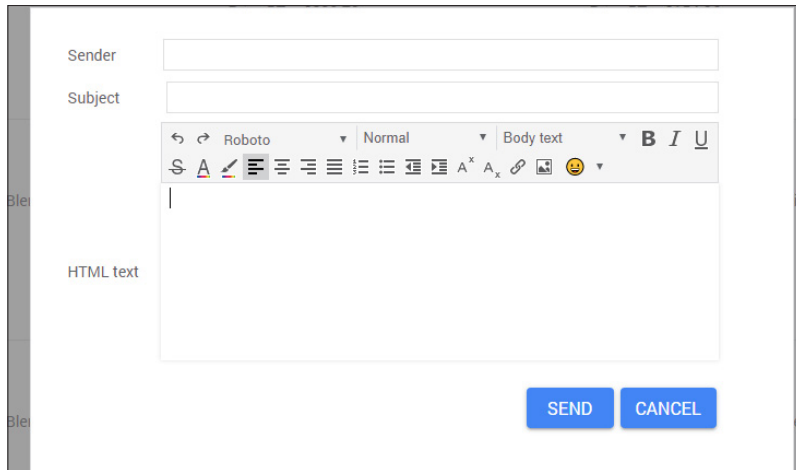
Several WLanguage functions allow you to manage the incoming and outgoing emails. You also have the ability to access the email characteristics:

- sender, recipients,
- outgoing date, subject, message,
- attachments...

WEBDEV proposes several methods for managing emails:

- Management via Lotus Notes, Outlook or MS Exchange:
 - The **Lotus Notes or Outlook messaging software**: these programs allow you to send and receive emails.
 - The **"Simple Mail API"** (also called SMAPI or Simple MAPI): this management mode of emails is used by most of the Microsoft applications, especially by Microsoft Exchange.
- Management via the POP3, IMAP and SMTP protocols:
 - The **POP3** protocol: this protocol for receiving emails is recognized by all service providers. It is used to communicate with the server directly, available at your ISP. This protocol is used to list the incoming messages and to read them.
 - The **IMAP** protocol: this protocol for receiving emails allows you to leave the emails on the server so that they can be viewed from different messaging clients or webmail.
 - The **SMTP** protocol: this protocol for sending emails is recognized by all service providers.

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



To do so, we will be using the SMTP protocol. Indeed, this mode is commonly used all over the world.

See the online help for more details about the other methods.



Answer

If you have not created the pages from the previous lessons, you can follow this lesson by opening a corrected project: in the WEBDEV home page (Ctrl + <), click "Tutorial" and select "Full WEBDEV Site (With pages)".

A full corrected application project is also available: in the WEBDEV home page, on the "Home" pane, click "Tutorial" and select "Full WEBDEV Site (Answer)".




Examples

The "Sending emails" example (unit example), supplied with WEBDEV, gives more details about the email management. This example is accessible from the WEBDEV home window.

A popup page to send emails

The popup page that we want to create will contain all controls allowing the user to type the different email elements. A "Send" button will group all 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 Popup page via the handles.
 6. Save the page ( or Ctrl + S).

Creating the controls used to type the email characteristics


To write an email, the user must have:

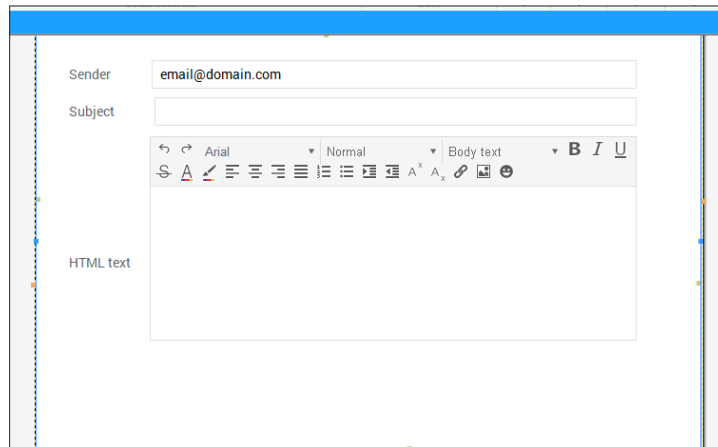
- A control used to type the sender address.
- A control used to type the email subject.
- A control used to type the email text.

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

Remark: the recipient being the site developer, the corresponding address will be found in the send code of email. Similarly, the parameters of SMTP server will be typed in the code directly.


- ▶ To create the edit control corresponding to the sender address:
 1. On the "Creation" pane, in the "Usual controls" group, expand "Edit".
 2. Select the preset "Email" edit control.
 3. Click in the page: the edit control is automatically created.
 4. Modify the control caption: "Sender".

- ▶ To create the edit control corresponding to the email subject:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click below the "Sender" control: the edit control is automatically created.
 3. Modify the control caption: "Subject".
- ▶ For the message body, we will be using an HTML edit control: the user will have the ability to format the email text via a specific toolbar.
 1. On the "Creation" pane, in the "Usual controls" group, expand "Edit".
 2. Select a preset "HTML text" edit control.
 3. The control shape appears under the mouse cursor.
 4. Click below the "Subject" control: the edit control is automatically created.
 5. Enlarge the control in order for several lines to be visible.
 6. Open the control description window (double-click the control).
 - On the "General" tab, modify the HTML toolbar display mode ("HTML toolbar" option). This toolbar must always be visible.
 - Validate the control description window.
 7. Reposition the control if necessary.
- ▶ Align the different controls found in the page.



We will now create the Button control to send emails.

Sending the email

- ▶ To create the Button control to send emails:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click the location where you want to create the Button control (for example, at the bottom of the page).
 3. Select the control and modify its caption (press Enter for example). The new caption is "Send".
 4. Edit the code of this control (F2).
 5. Type the following codes in the "Click" server event:
 - the code for opening the connection and for starting the SMTP session:

```
MySession is EmailSMTPSession
MySession.ServerAddress = "smtpserver"
// Type the address of SMTP server
MySession.Name = "user_name"
// Type the user name (if necessary)
MySession.Password = ""
// Type the password (if necessary)

// Starts the SMTP session
IF EmailStartSession(MySession) = False THEN
  ToastDisplay("Unable to connect to the SMTP server.", ...
    ErrorInfo())
  RETURN
END
```

This code is using an advanced **emailSMTPSession** variable. The different properties of this variable are used to define the characteristics of SMTP session. Then, **EmailStartSession** associated with this variable is used to start the session.

- the code for preparing the email:

```
Suggestion is Email
Suggestion.Sender = EDT_Sender
Suggestion.Subject = EDT_Subject
Suggestion.HTML = EDT_HTML_Text
Suggestion.Message = HTMLToText(EDT_HTML_Text)
Suggestion.Recipient = "developer@mysite"
```

This code is using an **Email** variable. The different properties of this variable are used to define the characteristics of email to send. This code associates the content of different page controls to the properties of **Email** variable.

- the code for sending the email:

```
// Send the email
IF EmailSendMessage(MySession,Suggestion) = False THEN
  ToastDisplay("Sending error",ErrorInfo())
ELSE
  ToastDisplay("Thank you for your suggestion.")
END
```


The email is sent by **EmailSendMessage**. All you have to do is pass in parameter the variable containing the characteristics of SMTP session and the variable containing the characteristics of email to send.

If the email is sent, a Toast message is displayed, indicating that the email was sent. A Toast message corresponds to a future message.

- the code for closing the SMTP session:

```
// Close the SMTP session
EmailCloseSession(MySession)
```

This code closes the session with **EmailCloseSession**. Close the code editor.

- ▶ Save the page and its code ( or Ctrl + S).



Remark

The process for sending emails is given for information only. In a real site, we would have to check the parameters typed, process the errors, save a log file, ...

Improving the page

We are going to improve our popup page:

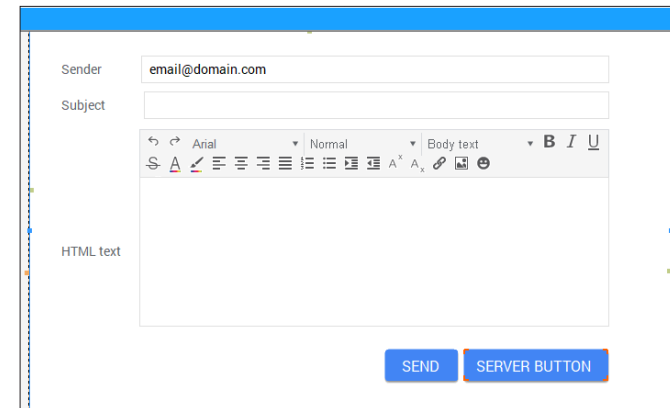
- Add a Button control to close this popup page.
- Start the popup page from the "PAGE_List_of_products" page.

Closing the popup page

- ▶ To add a Button control to close the popup page:

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

2. Click the position in the window where you want to create the Button control (for example at the bottom, to the right of the "SEND" Button control).



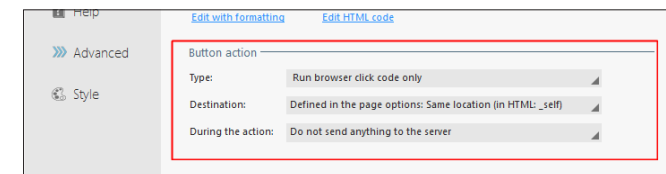
3. Select the control and press Enter. The caption becomes editable. Type "Cancel" and validate.
4. Display the events associated with the control (F2).
5. Type the following code in the browser click code:

```
PopupClose ()
```

PopupClose is used to close the popup.

- ▶ This function is a Browser function, run on the browser only: no return to the server is required. Therefore, we are going to modify the type of "Cancel" button consequently:

1. Display the description window of "Cancel" Button control (double-click the control for example).
2. In the "General" tab, in the "Button action" area, for the "During the action" option, select "Do not send anything to the server".



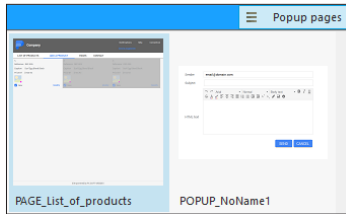
3. Validate the control description window.

Let's now see how the popup page can be opened.

Opening the popup page

The popup page for sending an email will be opened from the menu of "PAGE_List_of_products" page.

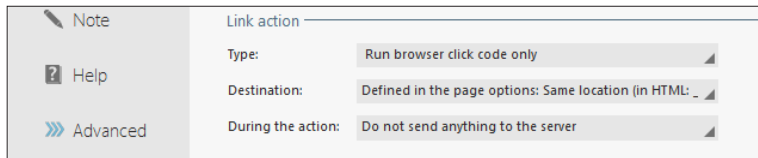
- Display the "PAGE_List_of_products" page:
 1. In the editor, in the page bar, expand "Popup pages".



2. In the list that is displayed, click "PAGE_List_of_products".

► We will now create a Link control to send a suggestion:

1. On the "Creation" pane, in the "Usual controls" group, click "Link".
2. Then, click the top section of the page (above the "Notifications" control): the Link control is created.
3. Modify its caption: "Send a suggestion" and validate.
4. Open the control description window (double-click the control).
5. In the "Link action" area:
 - The type must correspond to "Run the browser click code only".
 - The "During the action" option must correspond to "Do not send anything to the server".



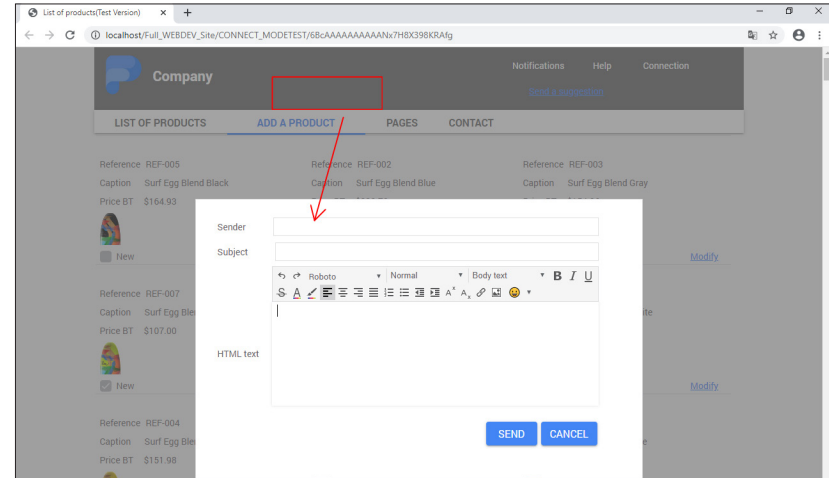
6. Validate the control description window.
7. Display the events associated with the Link control (F2).
8. Type the following code in the browser click code:

```
PopupDisplay (POPUP_NoName1)
```

PopupDisplay is used to display the popup.

Remark **Why didn't we open the popup from the menu found in the page template?**
 The popup page that was created is linked to the "PAGE_List_of_products" page. It can only be used from this page.
 To use a popup from a page template, the popup must be created from the page template.

- Save the page and its code (or Ctrl + S).
- Run the page test (among the quick access buttons) and open the popup for sending a suggestion.



- Close the browser.

Remark **Why does the link not appear in the browser at the specified location?**
 In the page editor, the Link control is positioned above the search control, but at runtime it appears above the option "ADD A PRODUCT". This difference is due to the anchoring of the Link control.
 Indeed, in the page editor, various red arrows are displayed in the page header. These arrows indicate that the controls are anchored. We will see how anchoring works in detail in the Advanced concepts lessons. The Link control we created is not anchored and it follows the movement of the controls around it.

- To anchor the Link control to the right (and keep it from moving):
 1. Select the Link control.
 2. In the popup menu (right click), select "Anchor".
 3. In the window that appears, select and validate.

LESSON 5.4. IDENTIFYING THE USER: THE USER GROUPWARE

This lesson will teach you the following concepts...

- What is the user groupware?
- Integrating the user groupware.
- Configuring the user groupware.
- Checking the user groupware.



Estimated time: 20 mn

Overview

A site can be used by different contributors with different profiles. It is sometimes necessary to define different access levels according to the Web user (customer, salesman, manager for example).

Let's take a simple example: when implementing an e-business site, the site proposes the following features:

- Price list view,
- Price list modification,
- Order entry,
- 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 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 in order to limit the access to the page for adding products.



Answer

If you have not created the pages from the previous lessons, you can follow this lesson by opening a corrected project: on the "Home" pane, in the WEBDEV home page (Ctrl + <), click "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.



Remark

When the integrated user groupware is not used, a standard connection page is proposed. If the user types his login and password, he is allowed to access the site.

When the integrated user groupware is used, a specific control template is made available to the developer. This control template is used to include in the page a link allowing the user to connect. We are going to present this solution in this lesson because it is flexible and it can be easily included.

3. Validate. A message is displayed, indicating that a "supervisor" user is created by default.



Remark

A single user exists by default, the supervisor. During the first site start, connect yourself by using the: "supervisor" login and the "supervisor" password.

4. Validate this message. The user groupware is integrated in the project.



Remark

In this example, we will keep all default options. Several options can also be configured. See the online help for more details.

Configuring the user groupware

The configuration of user groupware is performed when running the site. This configuration consists in defining the different users as well as their rights on the site pages and controls.



Remark

The configuration of users can be performed:

- when developing the application. The necessary data files (regarding the users and their rights) can be installed with the site.
- when the site is installed, by the administrator.

Including the customer section in the site

In this example, we 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.



Remark

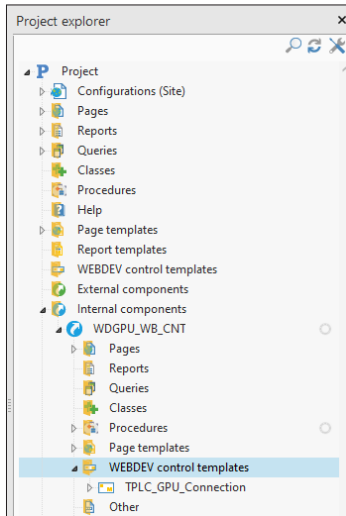
A control template is a specific page containing several controls. All types of controls can be found in this page. A control template is a file whose extension is "WDT".

The first benefit of a control template is the reusability. A control template found in a project can be re-used in any project page. The modifications are performed once only in the control template and the modifications are automatically applied by WEBDEV.

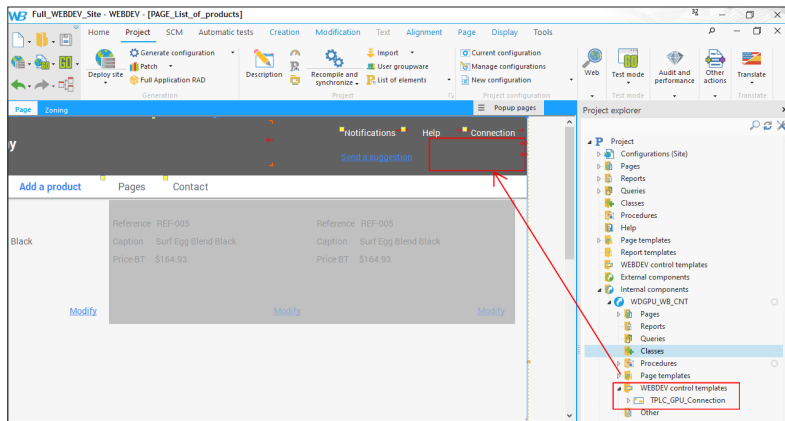
The control templates can be overloaded: code can be added, the controls can be moved in the page that is using the control template. The controls can be modified.

- To include the connection link:
1. Open the "PAGE_List_of_Products" page in the editor.
 2. Go to the "Project explorer" pane, "Internal components", and expand "WDGPU_WB_CNT". This internal component was included in the project when implementing the integrated groupware. It contains all elements required to its management, especially the control template used to connect.

3. Then, expand "WEBDEV control templates".



4. Select the control template named "TPLC_GPU_Connection" and perform a Drag and Drop into the "PAGE_List_of_products" page. Drop the control template above the menu bar.

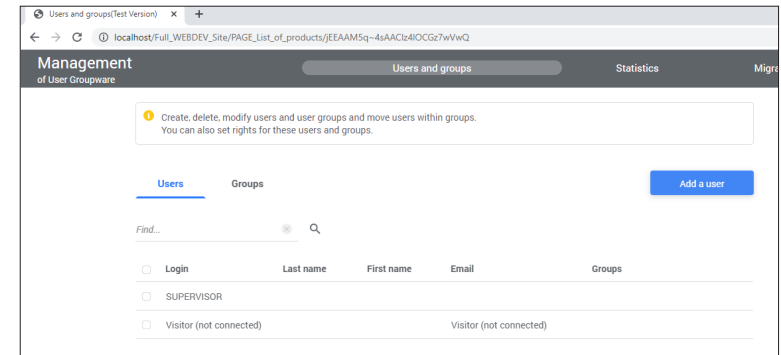


5. The control template is integrated. The "Connection" link appears.

Site test

▶ Let's now run the test of our site:

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



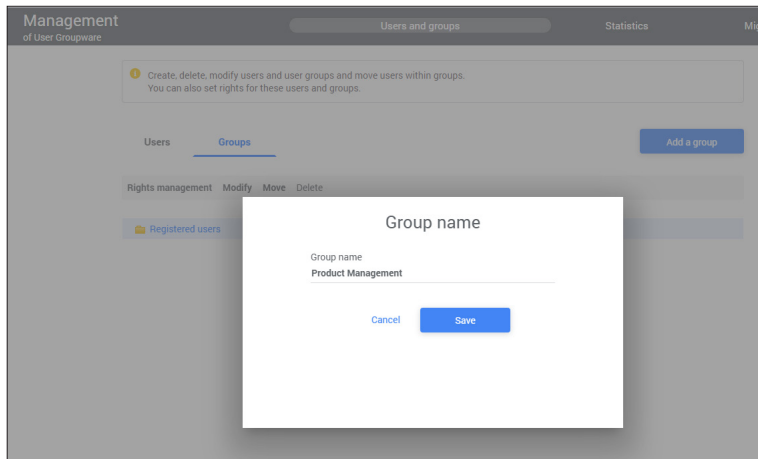
Creating users and groups

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

Then, we are going to create a user named Alison, associated with the "Product Management" group.

► To create a new group of users:

1. Click the "Groups" tab.
2. Click "Add a group". The screen for entering a new group is displayed.
3. Type the group name: "Product Management".



4. Click on "Save". The "Product Management" group appears in the list of groups defined for the user groupware.

► To create a user:

1. Click on the "Users" tab.
2. Click on "Add a user". The screen for entering a new user is displayed.

3. Type the following information:

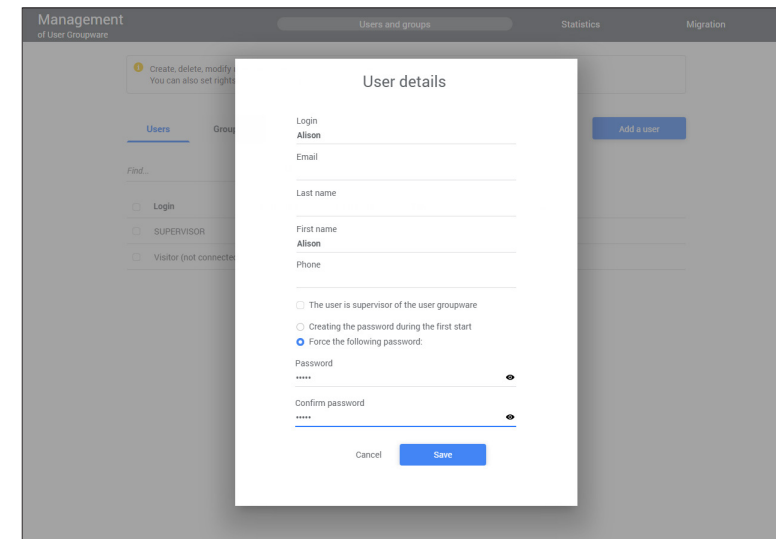
- Login: Alison
- First name: Alison
- Select "Force the following password" and enter the password: Alison.
Remark: You can give the user the ability to choose his password during the first connection.



Remark

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.



4. Click on "Save". The user named "Alison" appears in the list of users defined for the user groupware.

► To associate the user named "Alison" with the "Product Management" group:

1. Select the user named "Alison" in the page.
2. Check the box at the beginning of row: the buttons "User groups" and "Delete user" appear.
3. Click "User groups".
4. In the popup that appears, check "Product Management" then click "Validate".
5. The association is performed. The name of the "Alice" user group appears in the lists of users.

Defining rights

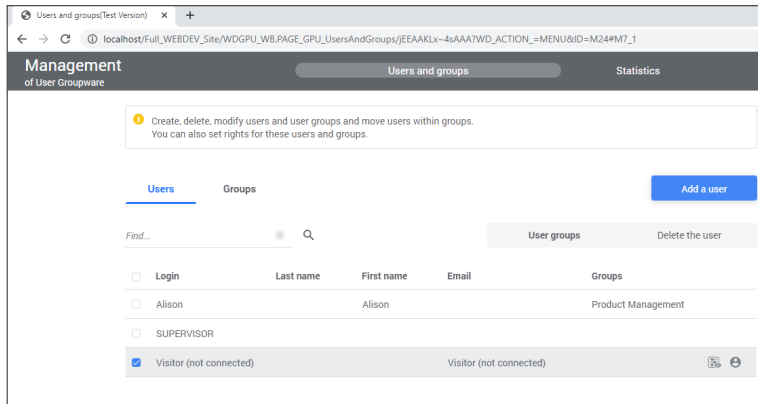
We will now define the access rights to the menu for adding a product.

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

► We are now going to define rights for the unconnected users (which means not identified by the groupware). These rights will be used by default when starting the site, as long as the user is not connected.

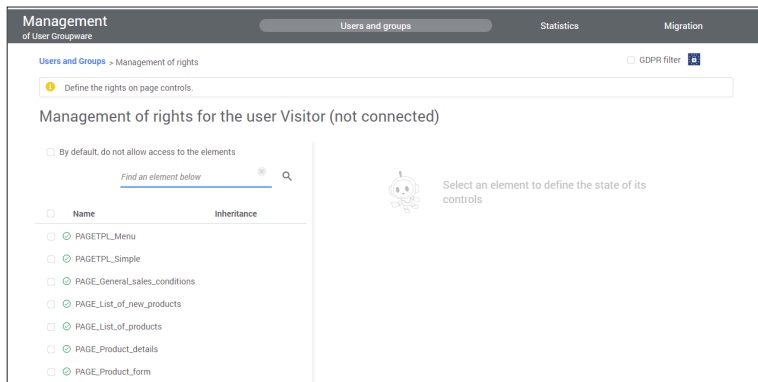
► To define the rights:

1. To prevent all users from accessing the product addition page, select "Visitor (not connected)" (click the check box at the top of row). Each unconnected user is automatically associated with this user.



2. Click to set the rights.

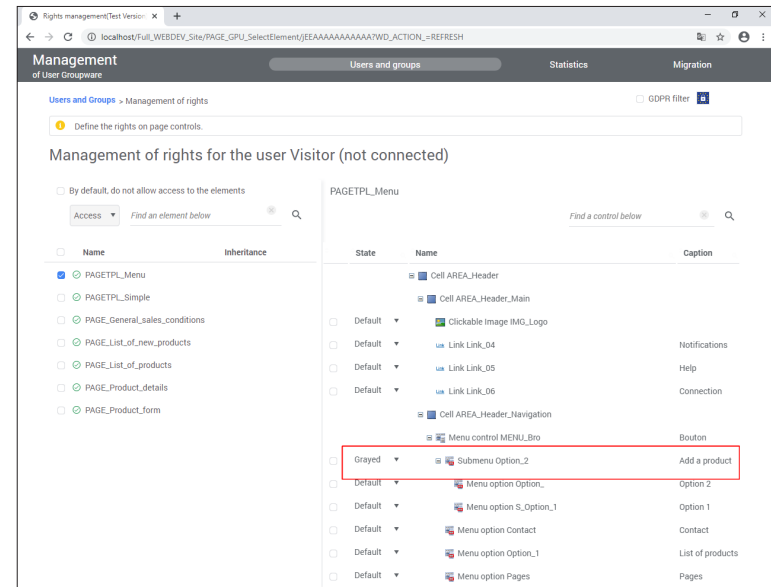
3. The page that is displayed is used to select each application page, page template or report.



- For each page, page template or report, it is possible to specify whether or not the element will be accessible to the user (or group) previously selected.
- For each page or page template, you can 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 pages that use the template.
2. The names of the different page controls appear on the right.
3. For "Sub-menu Option_2", select "Grayed" in the "State" combo box.



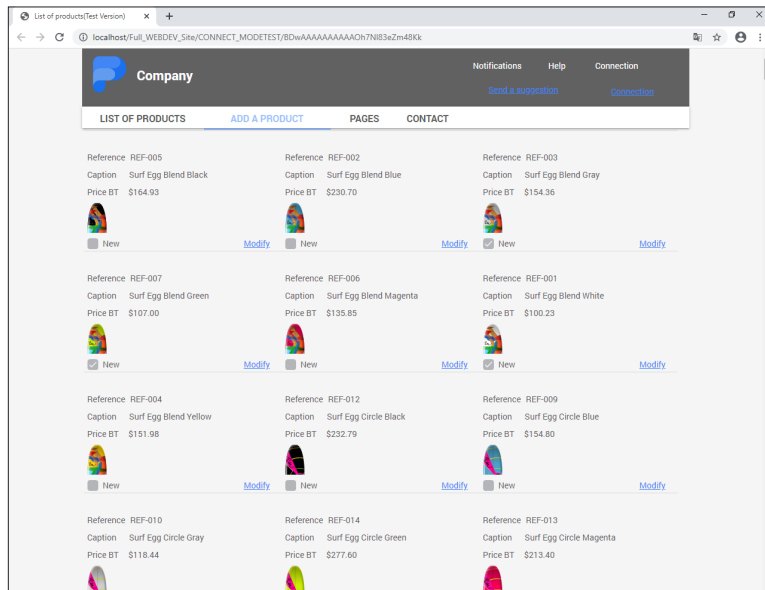
4. Close the browser.

Site test

► We are now going to run the site test.

1. Run the project test (among the quick access buttons).

2. By default, the option "ADD A PRODUCT" is grayed.



3. Click "Connection" and use the login "Alison" (and the password "Alison"). Validate.
4. The option "ADD A PRODUCT" is no longer grayed.
5. Close the browser.

Disabling the management of user groupware

The user groupware will no longer be used in the rest of this tutorial. Therefore, you can disable the management of user rights in this project:

1. On the "Project" pane, in the "Project" group, click "User groupware".
2. In the window that is displayed, in the "Integration" tab, select "No user groupware".
3. Validate.

LESSON 5.5. MANAGING THE MULTILINGUAL FEATURE

This lesson will teach you the following concepts...

- What is a multilingual site?
- Creating a multilingual site, step by step.



Estimated time: 50 mn

What is a multilingual site?

A multilingual site is a site that proposes an interface in several languages: English, French, German or any other language.

The same site created with WEBDEV can propose up to 64 different languages.

We are going to handle a project that can be run in English or in French, according to the user's choice.

The following steps are used to transform a site into a multilingual site:

- Choosing the project languages.
- Localizing the project elements (pages, reports, controls, ...).
- Localizing the messages found in the code.
- Programming the change of language in the site.

We are going to apply these different steps to the "Full_WEBDEV_Site" project. This project, available in English, will be translated into French.



Answer

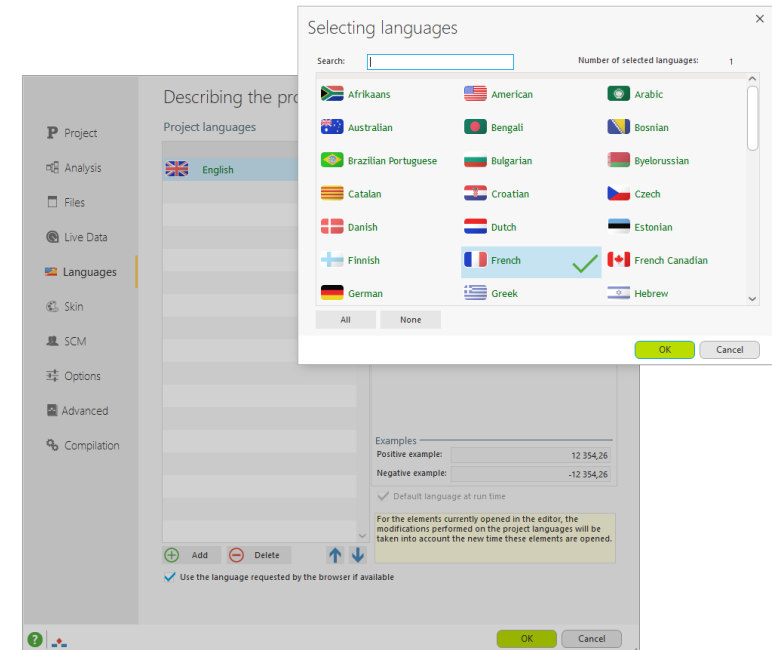
If you did not perform the operations in the previous lessons, you can follow this lesson by opening a corrected project: in the WEBDEV home page (Ctrl + <), click "Tutorial" and select "Full WEBDEV Site (With pages)".

Choosing the project languages

► The first operation consists in choosing the project languages.

1. Display the project description: on the "Project" pane, in the "Project" group, click "Description".
2. Click the "Languages" tab. Our project will support English and French.

3. Click the "Add" button. The window for selecting languages is displayed.

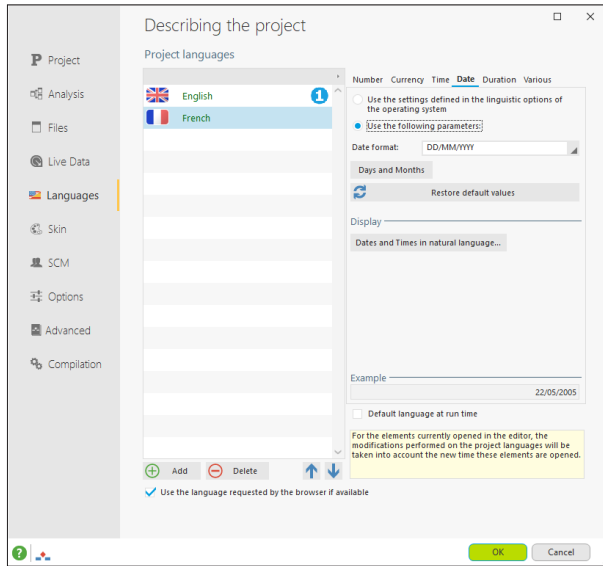


4. Click "French". A checkmark is displayed on the right of language.
5. Validate. The "French" language appears in the list of project languages.

► The "Languages" tab can also be used to configure the linguistic options regarding the numbers, the currencies, the dates, ... for the selected language. Let's see an example:

1. Click the "French" language.
2. Select the "Date" tab.
3. Specific linguistic options are used by default: you have the ability to define the date format as well as the translation used for the days and months. If you select "Use the settings defined in the linguistic options of the operating system", the parameters used in deployment will be the server parameters (and not the user parameters).

4. Keep "Use the following parameters".



Remark 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 project elements opened in the editor (pages, reports, ...) are closed and the additional languages are added to these elements.



Remark UI errors may occur. We will ignore them for now. These errors will be processed later in this tutorial.

Localizing the project elements

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

We are going to modify some elements in the "PAGE_List_of_products" page to present the different methods that can be used.

We are going to see how to modify:

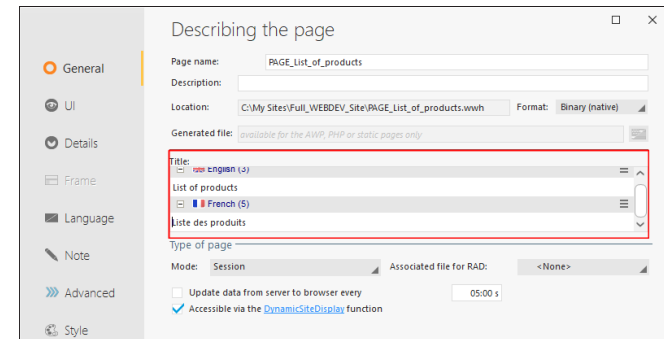
- the logo image used in the PAGE_List_of_products page.
- the captions of controls in the PAGE_List_of_products page.
- the menu options.
- a message displayed by the WLanguage code.

- Open the "PAGE_List_of_products" page in the editor (double-click its name in the "Project explorer" pane for example).

- Validate the update of template if necessary.

- First of all, check whether the PAGE_List_of_products page is associated with the different languages defined in the project:

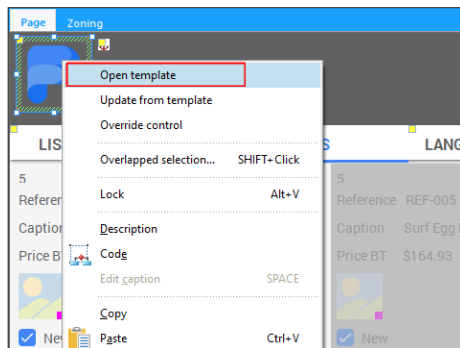
1. Display the page description ("Description" from the popup menu of the page).
2. Select the "Language" tab: the two languages selected in the project are displayed.
3. Select the "General" tab: the page title must be translated.




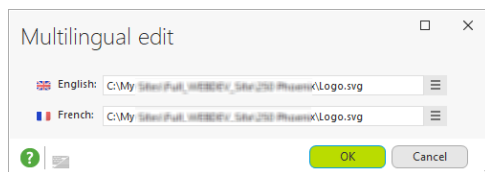
4. Replace the text found in the French section by "Liste des produits".
5. Validate the window.


Localizing an image

- ▶ To change the logo image used in the PAGE_List_of_products page according to the runtime language:
 1. Open the "PAGE_List_of_products" page if necessary.
 2. The logo image is found in the template associated with the page. Therefore, the associated page template must be opened:
 - Click the logo and display the popup menu.
 - Select "Open the template".



- The page template appears, enclosed by an orange border.
3. Display the description window of logo (double-click the control).
 4. In the "General" tab, on the right of "Image" section, click . Select "Multilingual" from the menu that is displayed.
 5. The window for managing multilingual images is displayed.





6. A different image can be used for each language:
 - Click the  button.
 - Select "Browse" from the menu that is displayed.
 This feature is very useful if you are using images containing text. The same image is used in our case.
 7. Validate the window for multilingual edit.
 8. Close the description window of control.
- ▶ As we are 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 option, or from the page editor.

- ▶ 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 option: display the popup menu and select "Option description".
 5. In the description window, type the option in French: "Liste des produits".
 6. Validate.

The options can also be translated in the page editor.


- ▶ To translate the "Add a product" option:
 1. On the "Display" pane, in the "Options" group, expand "Language displayed" and select the language that 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. Type the caption in French: "Ajouter un produit".
 6. Press Enter to validate the caption.
 7. Press Esc to exit from the edit mode.
 8. Save the page template ( or Ctrl + S).
 9. Restore the English display mode: on the "Display" pane, in the "Options" group, expand "Language displayed" and select "English".
 10. Update the pages that use the page template by clicking the  icon in the orange bar. Validate the update window.
 11. Close the page template displayed in the editor.

Localizing controls

A control can display various information to the user:

- a caption,
- an image, ...

This information must be translated. This information is accessible in the different tabs of the description window of control.

- ▶ To translate the "Modify" link found in the "PAGE_List_of_products" page:
 1. Select the "Modify" link.
 2. Display the description window of control ("Description" from the popup menu).
 3. Type the caption in French: "Modifier".
 4. Validate.
 5. Save the page ( or Ctrl + S).

Localizing a programming message

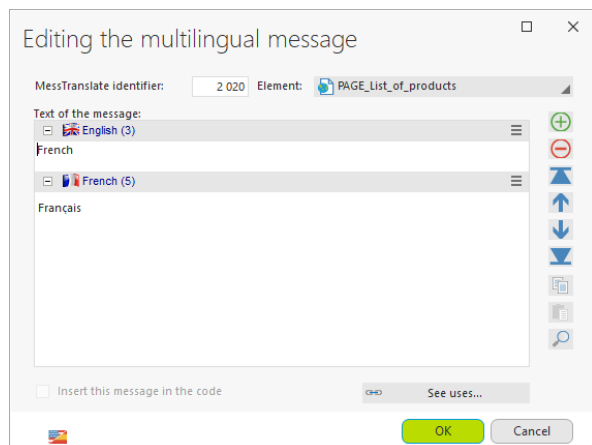
All messages found in your program can be translated into several languages.


Let's see how to translate a programming message:

1. Display the code editor (press F2 on the PAGE_List_of_products" page).
2. Write the following code in the "Global declarations" event:

```
sMyString is string = "French"
```

3. To translate this type of message, position the cursor in the "French" string and press Ctrl + T. On the "Code" pane, in the "Languages" group, you can also expand "Translate strings" and select "Translate messages".
4. The window for editing a multilingual message is displayed. This window allows you to translate all messages of your program into all project languages.
5. In the "French" section, type "Français" and validate.



6. The  icon as well as a digit appear in the code editor. These icons indicate that the multilingual message exists in 2 languages.
7. Close the code editor.

The translation tools

Some application elements have been translated manually.

Several methods can be used to translate this information:

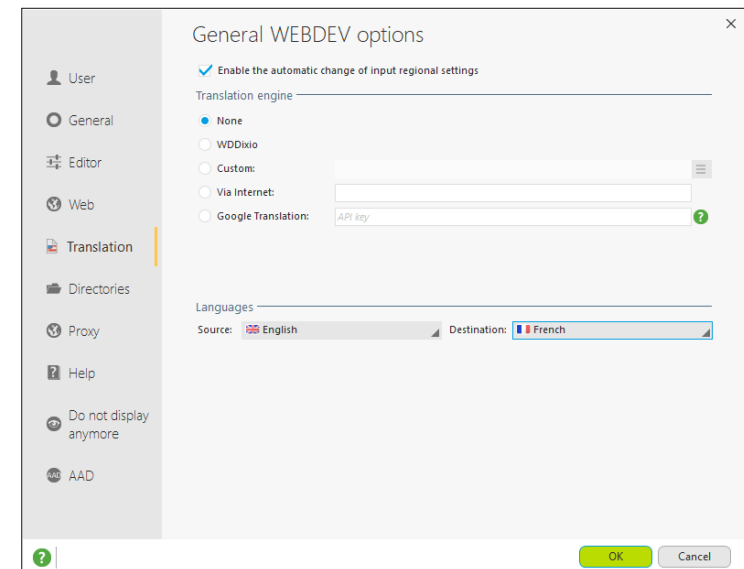
- a direct translation of messages performed in the different editors: this is the feature that was just used to translate some elements of our site. This translation can possibly use a translation tool, Google Translate (providing that you own a license), ...
- an "industrialized" translation performed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

Direct input of translations

The translations are typed in the product interface directly: this is the method that was used until now.

If you want to use a translation software or a translation site, WEBDEV can be configured to use this software:

1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV".
2. Display the "Translation" tab.




3. Specify:

- Whether the regional settings must be automatically enabled according to the language used for the input. In this case, if the language requires a specific character set, this character set will be automatically selected.

- The software or the site that will be used for translation. You have the ability to use WDDixio, translation dictionary supplied with WDMMSG (see next paragraph), a specific translation software or site, or Google Translate. See the online help for more details.

- The supported languages.

4. When the translation parameters are defined, you have the ability to click the  button found in the different description windows of project elements: this button allows you to use the software defined for translation.

Translation with WDMMSG and WDTRAD

An optional tool named **WDMMSG** is available, allowing you to:

- check out all 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 translation tools
- in HFSQL format.

WDMMSG is also supplied with WDTRAD, a computer-assisted translation tool. WDTRAD is used to easily type all translations for the multilingual information of a project.

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

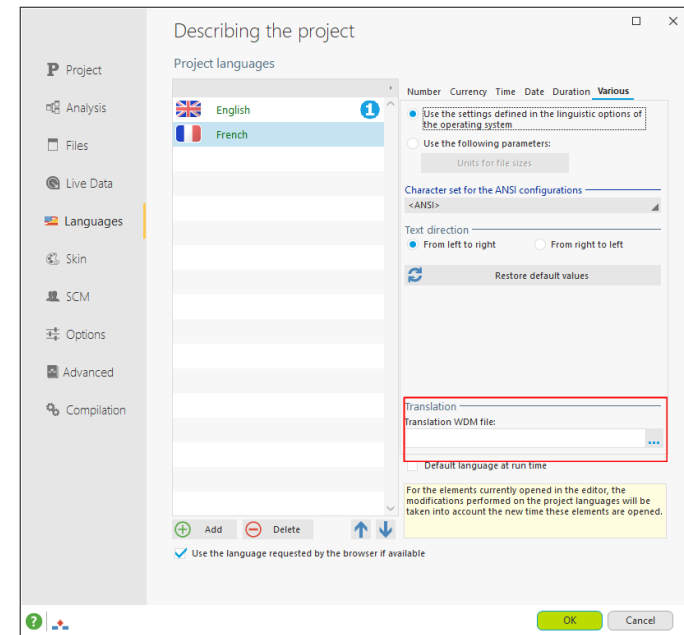
Other elements to translate: the framework messages

Various information and messages are found in the WEBDEV framework. For example, the names of days and months used by the functions for date management come from the WEBDEV framework. To translate one or more libraries of this framework, you must use WDINT (optional tool supplied with WDMMSG).

This software is used to get a WDM extension file containing all the translations of libraries. To use this file in your application:

- you have the ability to use **LoadError**.

- you have the ability to include the file to the project description in the "Languages" tab. Simply select the desired language and click on the "Various" tab.



Contact PC SOFT Sales Department for more details about WDINT.

Programming the change of language

By default, the project is run in the runtime language defined for the project, in the "Languages" tab of project description ("Description" in the "Project" pane).

In a site, the language can be chosen via a menu option. You can change the language of the application being run using **Nation** in the code associated with the menu option.

Adding a menu option

► To add a menu option:

1. If necessary, open "Product_List_PAGE" in the editor (double-click on its name in the "Project explorer" pane).
2. Click the menu and select "Open the template" from the popup menu.
3. In the page template, select and click "Pages".
4. The menu becomes editable and a yellow border is displayed.
5. Press the Space key on the keyboard: the caption becomes editable.
6. Type the caption ("Languages") and validate.
7. Select the "LANGUAGES" option you have just created.
8. Display the popup menu (right click) and select "Insert a sub-menu".
9. Select the menu option that was just created.
10. Press Space to edit the caption.
11. Enter the caption of the first sub-option: "English". Press Enter to validate the input.
12. Select the first sub-option again.
13. Press Enter: a second sub-option is created.
14. Press the Space key and type the caption of second sub-option: "French".

► As we are positioned in the page template, let's translate the last menu option:

1. Select "CONTACT".
2. Display the description window of option.
3. Type "Contact us" for the English language.

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

Programming

► To type the code for managing languages:

1. Select "LANGUAGES .. FRENCH" in the page template displayed in the editor.
2. Display the popup menu (right click). Select "Code".
3. Enter the following code in the server code of the menu option:



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

4. Close the code window.
5. Select "LANGUAGES .. ENGLISH" in the page template displayed in the editor.
6. Display the popup menu (right click). Select "Code".
7. Write the following code:

```
Nation (nationEnglish)
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 reload a page (the current page in our case) in order to take the change of language into account.

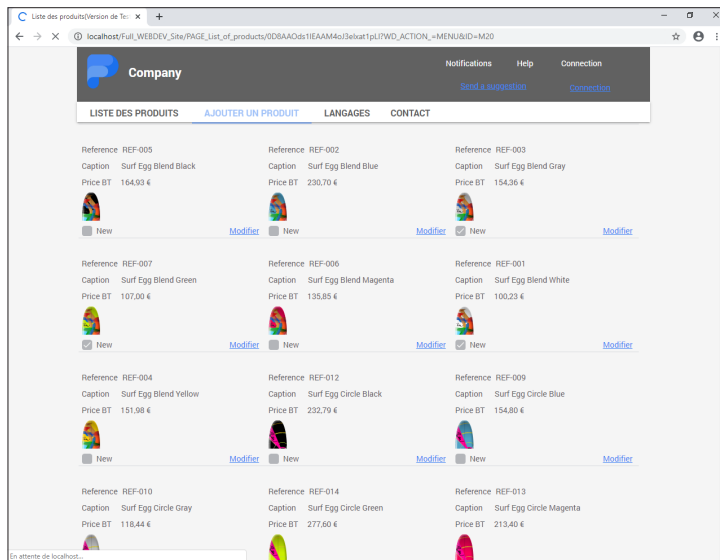
8. Close the code window.
9. Press Esc to exit from the edit mode.
10. Save the page template ( or Ctrl + S).
11. Update the pages that use the page template by clicking the  icon in the orange bar. Validate the update window.
12. Close the page template displayed in the editor.

Project test

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 .. ENGLISH".
3. The elements that have been translated are displayed in French:



4. Close the browser.

LESSON 5.6. SEO

This lesson will teach you the following concepts...

- SEO wizard.
- Properties of pages.
- Navigation map.
- Physical naming of pages.



Estimated time: 20 mn

Overview

This lesson presents the best practices to adopt in order for your Internet site to be properly optimized in search engines. Indeed, having an Internet site is good, but optimizing it for search engines is even better!



Remark

This section concerns the developers of Internet sites only. SEO is not essential if you are developing an Intranet site.



Answer

If you did not perform the operations in the previous lessons, you can follow this lesson by opening a corrected project: on the "Home" pane, in the WEBDEV home page (Ctrl + <), click "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 Session 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 optimized, an Internet site must necessarily be developed in AWP mode.

The available methods

WEBDEV proposes several tools to improve the search engine optimization of your AWP pages:

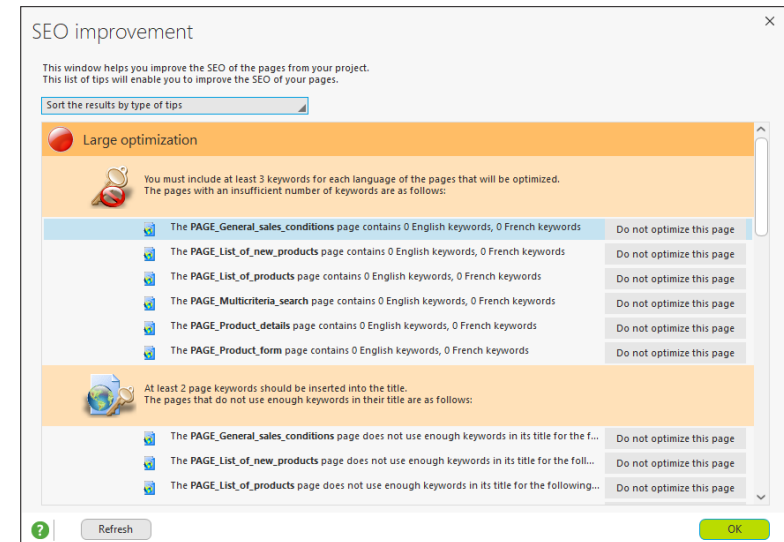
- An seo wizard.
- The properties of pages.
- The navigation map.

We are going to present these different methods.

SEO wizard

WEBDEV is supplied with an SEO wizard.

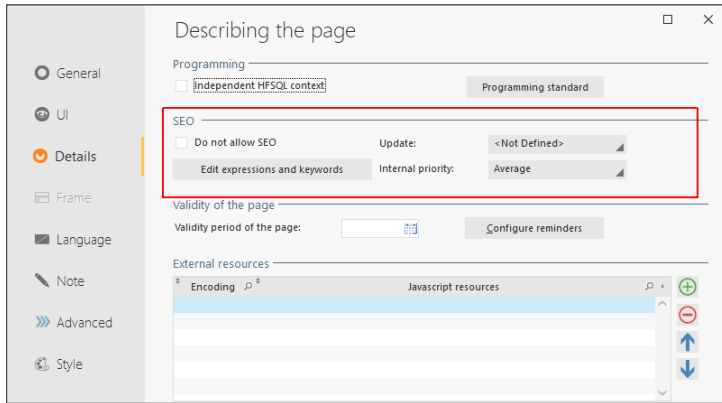
- To start this wizard, in the ribbon, on the "Project" pane, in the "Web" group, expand "SEO" and select "SEO wizard".



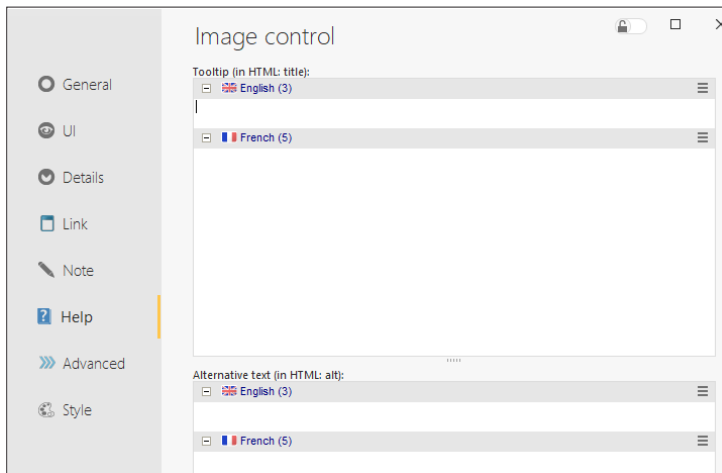
The wizard proposes tips to improve search engine optimization. The most important tips are as follows:

- **Title of pages:** The title of pages must reflect the content and contain the important keywords. The page title is defined in the description window of the page, for each language.
- **Keywords of pages:** We advise you to associate keywords with a page. To define these keywords:
 - Display the description window of "PAGE_List_of_new_products" page: display the popup menu and select "Description".
 - Display the "Details" tab.

- In the "SEO" section, click "Edit expressions and keywords".



- **Alternative text:** The alternative texts are used to replace images when these ones are not displayed (because the Web user explicitly asked not to display the images, because they are not loaded yet, because the Web user is using a browser in text mode or because a screen reader for partially sighted person is used). 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 Image control, in the "Help" tab.



Properties of pages

As already seen, the SEO wizard indicates several modifications that can be performed in the page description window:

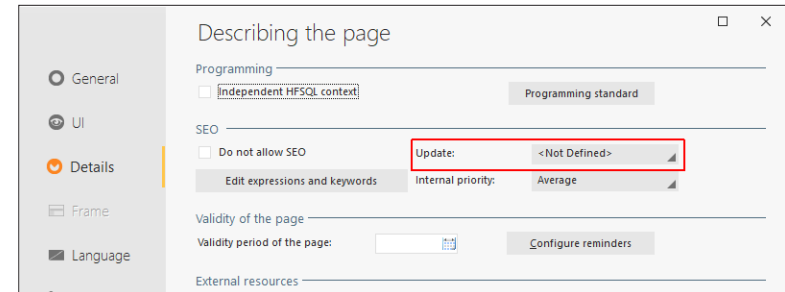
- Page title,
- Keywords of pages, ...

Other properties that influence SEO can be defined in the page description window:

- Update frequency,
- Importance in the site.

Update frequency

The update frequency tells the search engine how often the page should be reindexed.



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

- If the page changes more often that 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 will be proposed first to the Web user when several site pages correspond to a search.

Navigation map


The navigation map defines the organization of your site. In SEO, it has two uses:

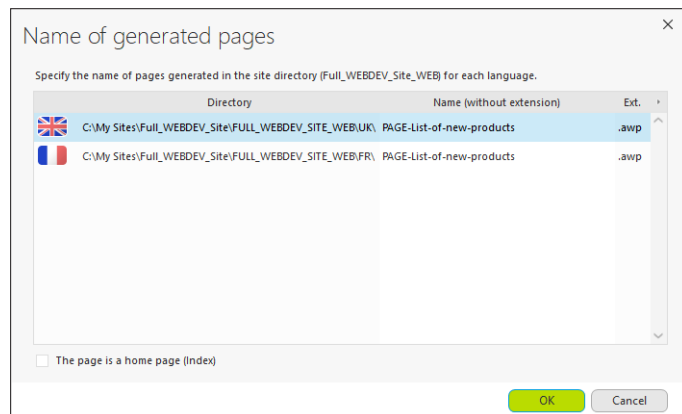
- allow you to display the "Site map" control to guide the Web users within the site.
 - generate the sitemap.xml file.
- To define the navigation map:
1. In the ribbon, on the "Project" pane, in the "Web" group, click "Navigation map".
 2. A window is opened, allowing you to build the tree structure of your site. An automatic build mode can help you create a tree structure by automatically detecting the links between pages. The navigation map can be configured later.
- The sitemap file is automatically built by WEBDEV from the navigation map. 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 webmasters of search engines so that they can index the content of your site.

Physical naming of pages

To improve SEO even more, you can define the physical name of 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) is PAGE_SalesDASH can be generated in French in the 'tableau-de-bord-ventes.awp' file and in English in the 'sales-dashboard.awp' file.

- To configure the generation name of a page:
1. Display the description window of "PAGE_List_of_new_products" page.
 2. In the "General" tab, click the "Language" button () on the right of "Generated file" control.
 3. A window used to configure the name of generated pages is displayed:



4. You can change the page name and validate.

LESSON 6.1. DEPLOYING A SITE

This lesson will teach you the following concepts...

- How to deploy?
- Required configuration.



Estimated time: 30 mn

Overview

When the WEBDEV site is developed, it must be deployed on a server in order to make it accessible to the Web users.

We are going to present the different steps required to deploy a WEBDEV site.



Remark

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 WEBDEV site (in Session mode or in AWP mode):

- Remote deployment from the development computer (by HTTP or FTP).
- Deployment in PCSCloud (PC SOFT's Cloud).
- Remote deployment from a management computer (by HTTP or FTP) via a "deployment package".
- Deployment via PC SOFT test hosting service.
- 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.
- Deployment via a Docker image of the site.

Required configuration

For the deployment on the server, the following elements must have been installed and configured:

- A Web server,
- A WEBDEV application server (a 10-connection version is supplied with WEBDEV).

Remark: If a remote setup is performed by FTP, an FTP server must also be installed on the server.

When installing WEBDEV, you have the ability to install the test version of WEBDEV application server. This allows you to test the deployment of your dynamic WEBDEV 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 HTTP).



Remark

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 HTTP.

Installing and configuring the "WEBDEV Application Server - 10 Connections"

The setup program of "WEBDEV Application Server - 10 Connections" is available:

- for download on the www.windev.com site, in the "Download" section, "Upgrades", "Download WEBDEV 25", "Application server (deployment)".
- in the WEBDEV setup package: start "Menu.exe" and select "WEBDEV Server - 10 connections Setup".

Installing the "WEBDEV Application Server - 10 Connections"

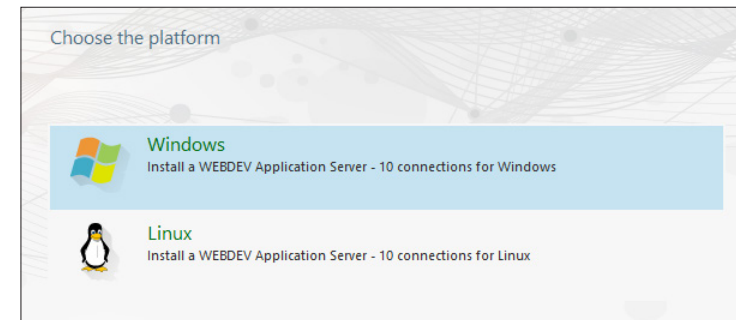
The setup steps are as follows:

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

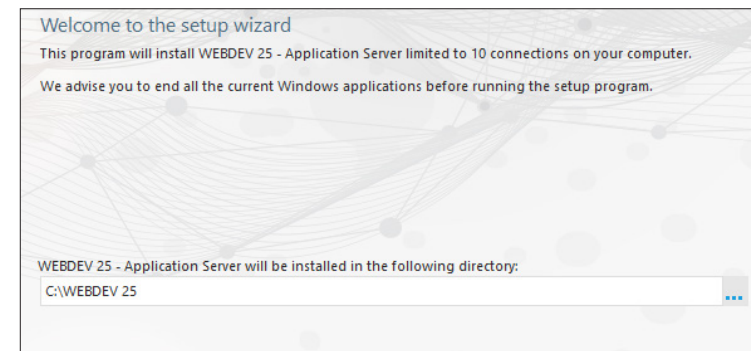


Remark

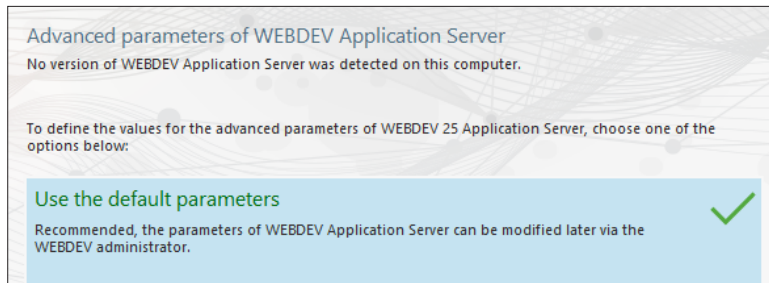
This section presents the deployment on a Windows server. If you want to use a Linux server, see the documentation about the WEBDEV application server for Linux or the online help.



3. Select the setup path of application server ("C:\WEBDEV 25" by default).



4. The next step is used to specify the advanced parameters of WEBDEV application server. Keep the option "Use the default parameters".



5. If the WEBDEV Application Server has never been installed on the computer:
- Specify the settings of a Windows account that will be used as WEBDEV administrator. This account will also be called "Hosting account". You can create a new Windows account or use an existing Windows account. Select an option and go to the next step.
 - The next step proposes the hosting of sites from previous versions. Select "Allow the deployment of sites in version 25 only" and go to the next step.
6. If a previous version of the WEBDEV Application Server has been installed on the computer, the wizard allows:
- defining the management of sites and webservices of earlier versions. Keep the option "Keep the sites in earlier versions" and go to the next step.
 - importing WEBDEV accounts. Keep the option "Do not import existing accounts" and go to the next step.
7. The next step is used to manage the hosting of SaaS sites. The SaaS activation automatically installs a site and a Webservice to manage your SaaS sites. See the online help for more details. Validate this step.
8. The next step allows defining the need for a secure connection to access all administration sites.
9. Validate the different setup steps.
10. Select the virtual Web servers on which the WEBDEV application server must be installed. The default Web site is sufficient. Validate.
11. Keep the selected options and check "Start the WEBDEV administrator". Validate.
12. The administrator is automatically started.

Remark

The PDF file named "WebDevDeployment.pdf" is installed with the WEBDEV application server 10 connections. The option "Display the user guide (pdf)" is used to automatically display this file at the end of setup. This file may help you solve the configuration problems linked to the access rights on the server.

Remark: If the option "Place the icons in the Start menu" was checked at the end of setup, this file will also be accessible via the "Start" menu.

13. "WEBDEV Application Server - 10 connections" is automatically configured. WEBDEV sites can be deployed immediately.

Configuration via the Hosting Control Center

By default, a WEBDEV Application Server "administrator" is created when installing WEBDEV Application Server - 10 connections. This user has all the rights (administration, deployment, statistics).

You can use the Hosting Control Center to configure the different characteristics of this user or to use a specific user to deploy your sites.

Reminder: The Hosting Control Center is a tool designed to automatically configure the following elements:

- WEBDEV Application Server,
- Web server (IIS),
- FTP server (IIS). The configuration of FTP server is required to a setup via FTP only.

► If not already open, start the Hosting Control Center from Windows Start menu.

First of all, we will see the different parameters available in the Hosting Control Center.

► To view the different parameters:

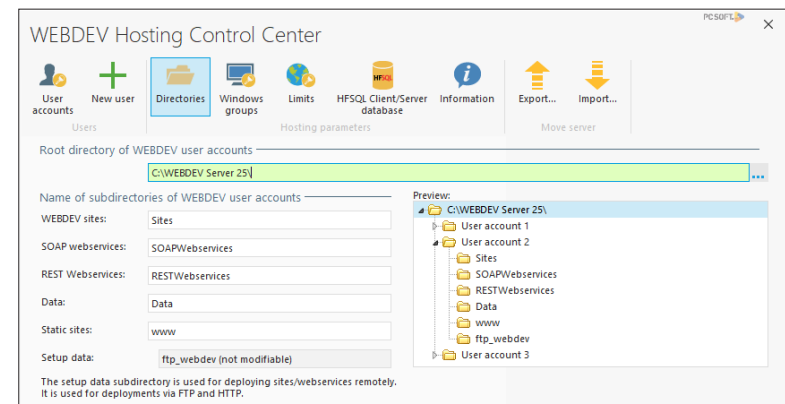
- To configure the directories used by your server, click "Directories" in the "Hosting parameters" group.
Remark: The different options of this group allow you to define the parameters that will be used by default for your hosting server.
- Indicate where the sites will be installed. Choose the root directory where the sub-directories of WEBDEV accounts will be created.



Remark

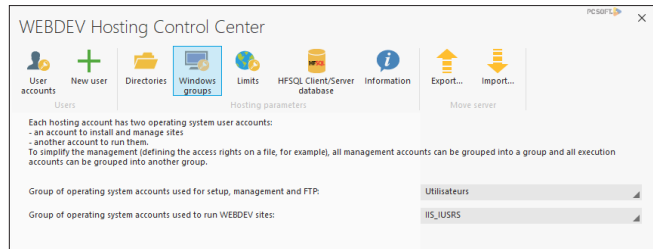
Use a directory local to the computer. If you want to use a network directory, a UNC path must necessarily be specified. The Internet guest of computer must have access to this path without having to authenticate.

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



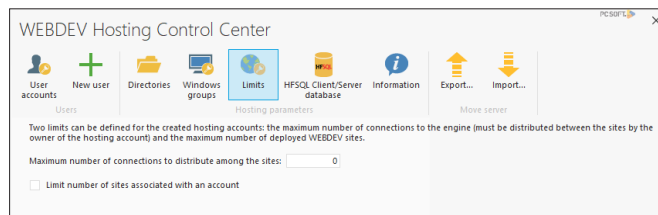
3. Click "Windows groups" and define the groups where the Windows users created for the deployment will be assigned.

For the deployment, you can create a group or use the standard "Power Users" group. A good practice for running sites consists in using the "IIS_IUSRS" group (on Windows versions where it is available).



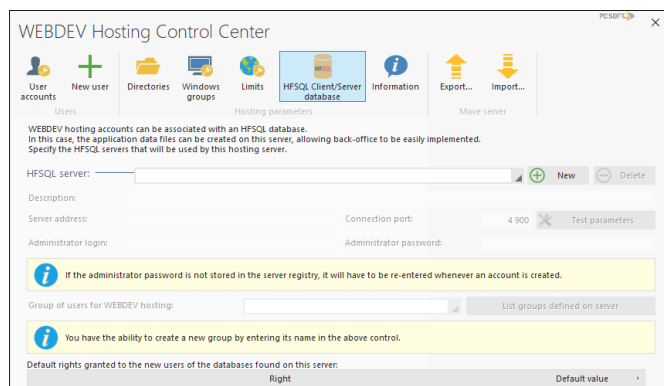
Click "Apply" to validate.

4. Click "Limits" to specify the resource limitations to apply to WEBDEV accounts:



- Maximum number of connections that will be shared between sites (0 corresponds to an unlimited number),
- Limitation regarding the number of sites to associate with an account, ...

5. Click "HFSQL Client/Server database" to configure the creation of an HFSQL database whenever a user is created.



6. Once the hosting is configured, click "Apply" then the "Refresh" button (if this button is proposed).

Creating a deployment account

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

This account can be used to deploy WEBDEV sites from WEBDEV.

To create a deployment account:

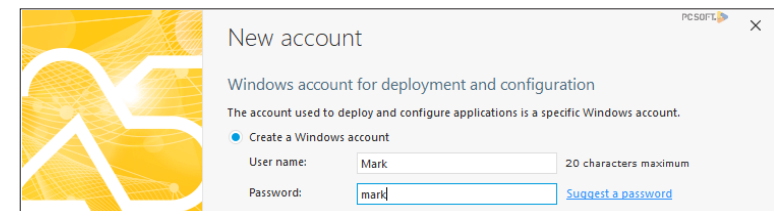
1. Click the "New user" button. The wizard for creating a new user starts. All you have to do is follow the different steps.

Remark: If not already done, the Hosting Control Center proposes to define a management group. In this example, answer "Yes" to the question "Do you want to continue without this group?".

2. You can:

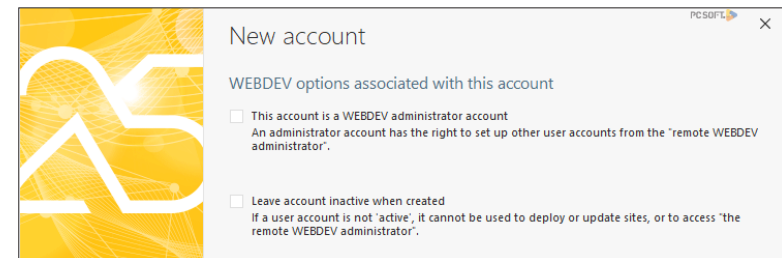
- Create a Windows account. This account will be used for deploying and configuring your sites.
- Use an existing Windows account.

In this example we will create a Windows account. Type the user name and password (you also have the ability to generate the password. In this case, don't forget to write it down!).



Go to the next wizard step.

3. The wizard proposes the WEBDEV options associated with the account:

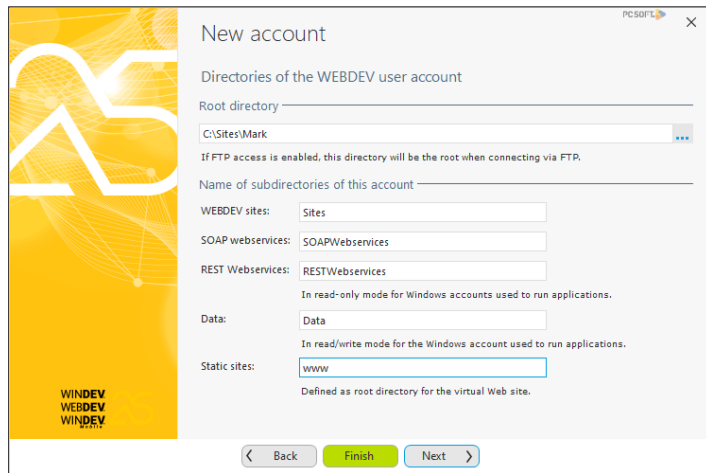


Keep the default options and go to the next step.

4. The wizard proposes to use a Windows account to run the applications. Keep the default options and go to the next step of the wizard.

5. Type the information regarding 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.

8. Go to the next step.

9. This step specifies whether FTP access and deployment should be allowed for the user. In our case, this option is not necessary because we will perform an HTTP deployment. Go to the next step.

10. The wizard is ended. Check all 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. We will use the user we created to test the deployment of a site by HTTP.

The different deployment modes

Now that our Web server was configured, we are ready to deploy our site. Several methods are available. The choice of a method mainly depends on the technical constraints (ability to use an FTP server or not, access to the server, ...).

A dynamic WEBDEV site can be deployed according to one of these methods:

- **Remote deployment from the development computer** (by HTTP or FTP). We are going to deploy our site according to this method by HTTP.
Remark: the setup via FTP requires an FTP server.
- **Deployment in PC SOFT Cloud.**

- **Remote deployment from a management computer** (by HTTP or FTP) via a "deployment package". This deployment mode operates like the direct remote deployment. 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 via WDDeploy. WDDeploy is a freely distributable tool that is supplied with WEBDEV.
- **Deployment by physical media** (DVD-ROM, etc.). 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 operates in Windows only.
- **Deployment via PC SOFT test hosting service.**
- **Deployment via a Docker image of the site.**

Let's now study the deployment by HTTP.

A detailed example: deployment via HTTP

We are going to deploy the "Full_WEBDEV_Site" project that was used in part 3 of this tutorial. A corrected version is available if you did not use this project yet.

► To open this project in WEBDEV:

1. Display the WEBDEV home page if necessary (Ctrl + <).
2. In the home page, click "Tutorial" and select "Full WEBDEV Site (Exercise)".



Answer

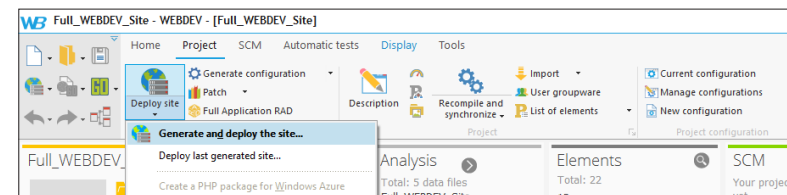
If you did not perform the operations in the previous parts, open the corrected project. This project contains the different pages created in the previous parts. To open the corrected project, in WEBDEV's home page, click "Tutorial" and select "Full WEBDEV Site (Answer)".

Preparing the setup

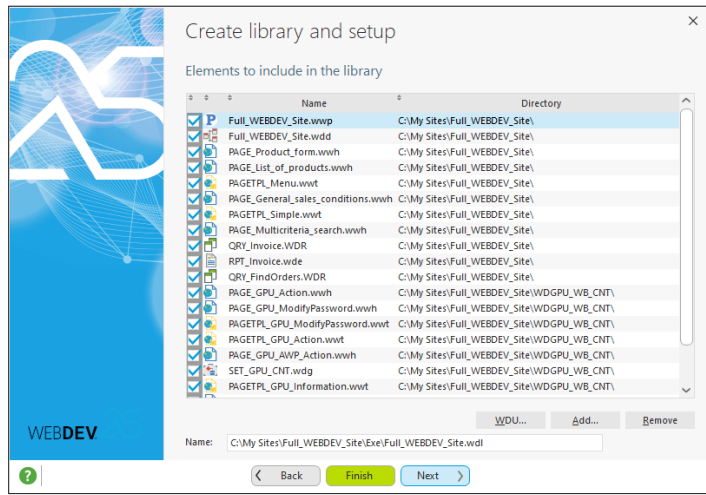
A setup wizard is supplied with WEBDEV ; this wizard allows you to easily install your site at the hosting company (Internet or Extranet site for example) or on one of your servers dedicated to WEBDEV hosting (Intranet site for example).

► We are going to use this wizard:

1. In the ribbon, on the "Project" pane, in the "Generation" group, expand "Deploy site" and select "Generate and deploy the site".



2. The wizard starts. Before performing the setup, all elements found in your site must be included in a library. A library is a file that groups all elements created during the development steps (description of database, pages, reports, queries, ...). The HTML pages and the images are not included in the library.



Go to the next step.

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

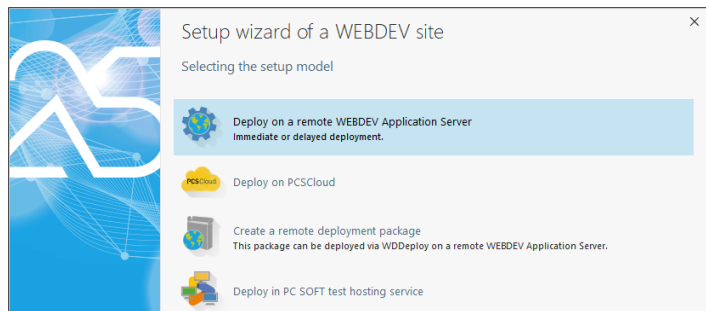
4. The information about the library version is used to enter the elements that will be displayed in the file properties in the Windows explorer. Go to the next step.

5. 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 HTTP:

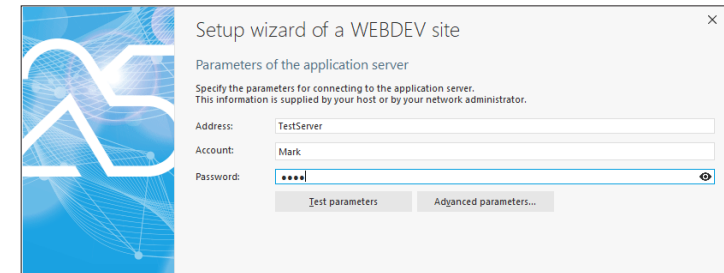
1. Select the first option "Deploy on a remote WEBDEV Application Server".



Go to the next step.

2. To define the server parameters, the following information must be supplied by your hosting company. We are going to type the information corresponding to the setup that was performed beforehand:

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



- User account that runs the setup. We created it in the previous step: this account is "Mark".
- Password associated with the account (in our example, also "mark").



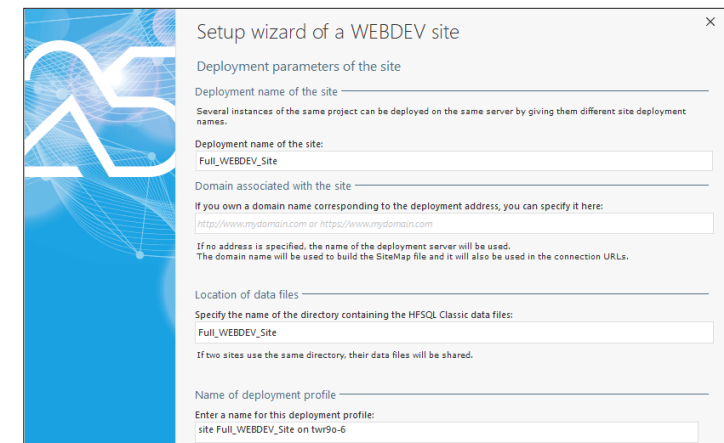
Remark

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

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

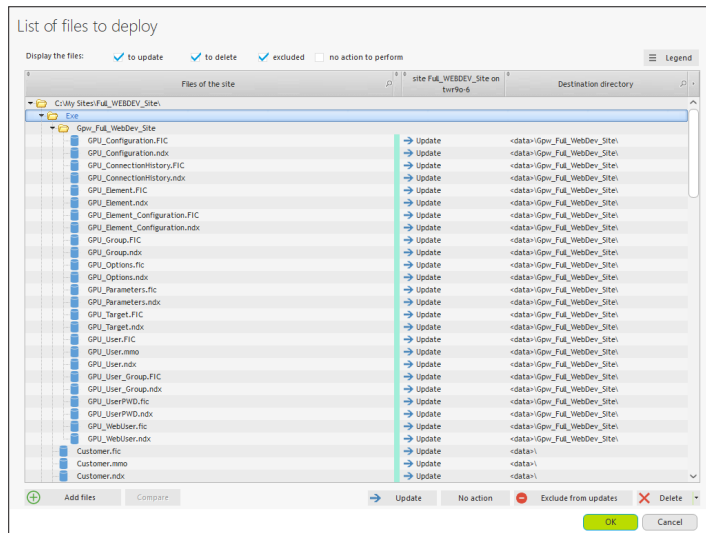
3. Once you have entered your account details, go to the next step.

4. Type the parameters for site deployment. We will keep the default options.



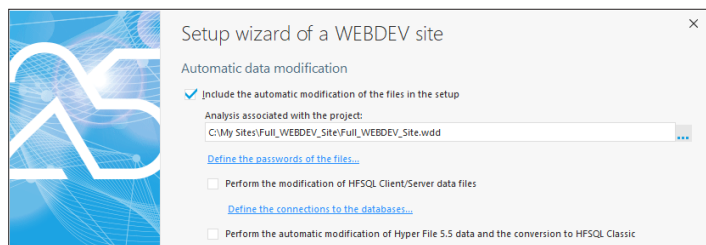
Go to the next step.

5. WEBDEV establishes the connection and it summarizes the operations to perform (number of files to update, number of files to delete, etc.). To get the details of the operations and modify them if necessary, click "Edit the list of files".



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.



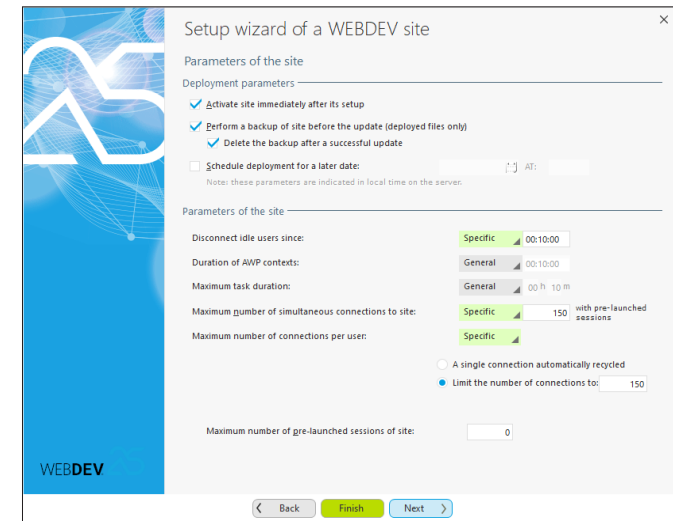
Keep the default options and go to the next step.

8. The wizard proposes to select the deployment mode:

- Deploy without disconnecting users. This option allows you to update the site without disconnecting the users.
- Deploy by disconnecting users. If this option is selected, connected users will be automatically disconnected to update the site.

Select "Deploy by disconnecting users" and go to the next step.

9. Specify the site parameter:



You can modify:

- The maximum number of connections to the site: if this value is set to "5" for example, only 5 Web users will be able to connect to your site at the same time.
- The maximum number of connections per Web user: if this value is set to "5" for example, a Web user will be able to start your site up to 5 times.
- The amount of idle time before user disconnection: this option is used to free all the resources occupied by the session of Web user if this one has performed no action since the specified duration.

10. By default, your site is immediately enabled after setup. The users will have no access to your site if "Activate the site immediately after its setup" is unchecked.

11. 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 ("Schedule deployment for a later date").

12. Go to the next step.

13. The wizard allows you to define the parameters for site security: Change of IP, ... Go to the next step.

14. The wizard proposes to automatically generate the statistical files for the site. These statistics affect for example the actions performed on the site, the origin of Web users, ... Keep the options proposed by default and go to the next step.

15. Validate the setup ("Finish"). The setup wizard transfers the files.

During the file transfer, the wizard compresses and encrypts the transferred data. Your data is transferred with a high-security level.

At the end of setup, a link allows you to immediately start the site.

Correspondence between the directories of the development computer and the deployment

The distribution of site files is slightly different between the development and the deployment server. Let's see a summary of the distribution automatically proposed by WEBDEV.

Directory on the development computer	Directory on the deployment server
<Project name>\<Project name>_WEB	<site>\<site name>\<site name in uppercase>_WEB
< Project Name>\Exe	For a site: <ul style="list-style-type: none"> • data files (.fic, .ndx, .mmo, .ftx): <data>\site name • other files: <site>\<site name> For a Webservice: <ul style="list-style-type: none"> • data files (.fic, .ndx, .mmo, .ftx): <data>\site name • other files: <webservice>\<site name>

where:

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

The distribution of files can be configured in the deployment wizard.

LESSON 6.2. MANAGING A SITE

This lesson will teach you the following concepts...

- Local WEBDEV administrator.
- Remote WEBDEV administrator.



Estimated time: 10 mn

Overview

Two tools can be used to manage the WEBDEV sites (and the Webservices):

- the WEBDEV administrator.
- the remote WEBDEV administrator.

Local WEBDEV administrator

The WEBDEV administrator is an application, installed with the Application Server.

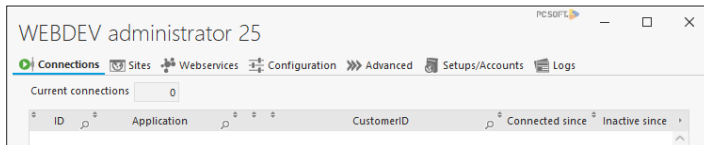


Remark

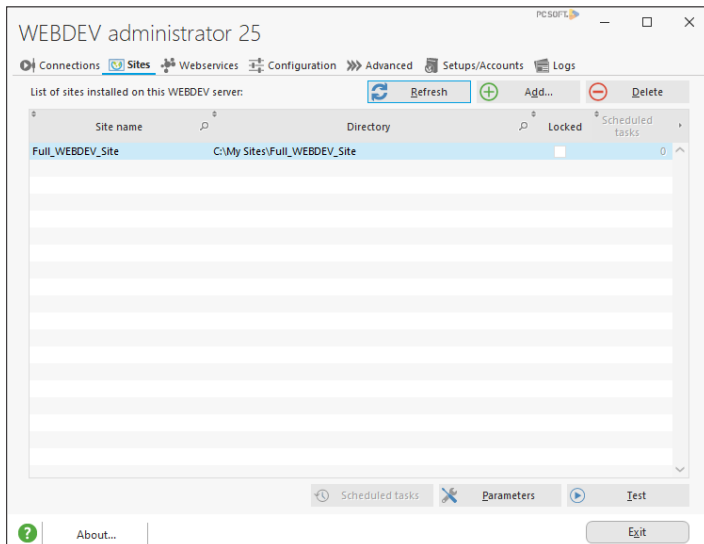
The WEBDEV administrator can be used only if you can directly access the Web server on which your site is deployed.

This application includes seven parts, represented by seven tabs:

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

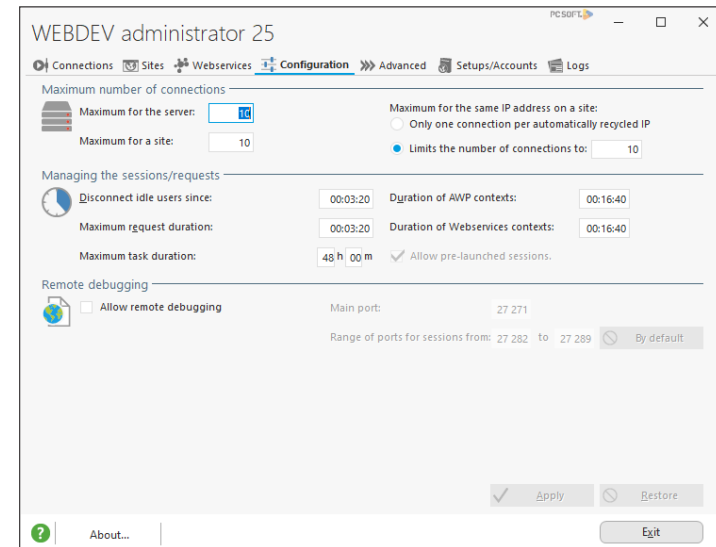


- Sites: Lists the sites deployed on the server as well as their parameters:

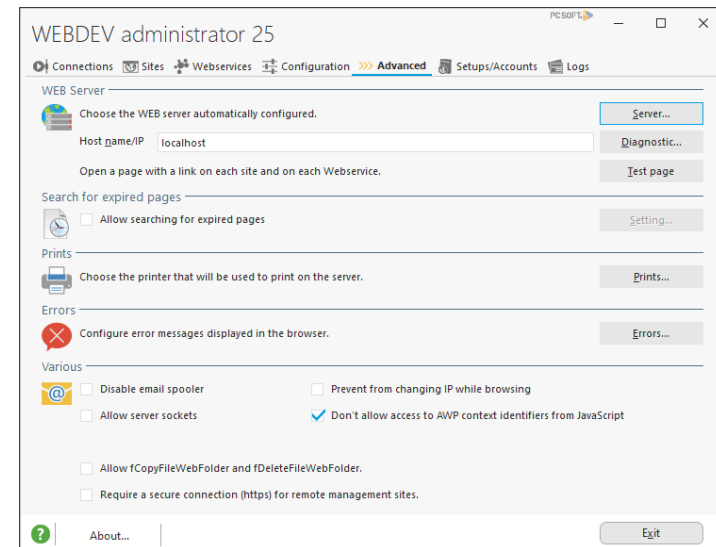


- Webservices: Lists the webservices deployed on the server as well as their parameters.

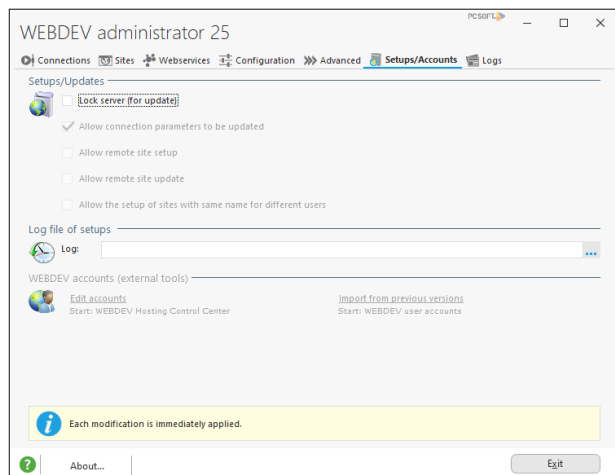
- Configuration: Used to manage the server configuration.



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



- **Setups/Accounts:** Used to lock the server for maintenance, to allow or forbid the remote setups and to configure the log of setups. This log is used to trace the elements installed in a site. It is especially useful before contacting our Technical Support if you encounter errors during the deployment.



- **Logs:** Used to see the logs for the sites or webservices for a specific period. The elements displayed are the elements for which errors occurred during the specified period and for which the logs are enabled.

Remote WEBDEV administrator

The remote administrator is a WEBDEV site that provides features similar to the WEBDEV administrator but that can be used to manage the WEBDEV sites remotely.



Caution!

If you are using the remote administrator, the security configuration is important, we advise you to access it via an HTTPS connection and to choose strong passwords.

Remark: This tool is not available with the WEBDEV application server 10 connections. When deploying your dynamic site at a hosting company (and if this hosting company allows the remote management of your site), this tool can be used to update the parameters of your site.

► To start WAdminWeb250:

1. Open your favorite browser (Internet Explorer for example) on your computer.
2. Type the following URL in the address bar of browser (while respecting the case):

```
http://computer/WAdminWeb250
```

where "computer" is the name of deployment server.

3. Type the login and password used to connect to the remote WEBDEV administrator.
4. The administrator starts.

Specific Web
features

LESSON 7.1. THE STYLES

This lesson will teach you the following concepts...

- The CSS styles.
- The WEBDEV styles.



Estimated time: 1 h

Overview

WEBDEV allows you to "design" your sites. To help you with this task, WEBDEV proposes to use two types of styles:

- The CSS styles,
- The WEBDEV styles.

Using styles for your controls presents a huge benefit: a modification performed in a WEBDEV or CSS style is automatically applied to all controls that use this style.

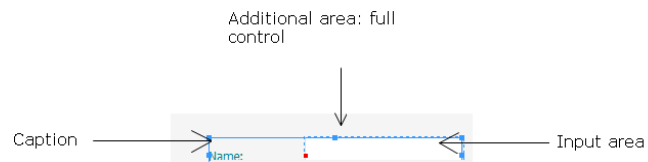
Discovering styles

Before using the styles, let's see their application support: the controls.

A control: several elements

Each WEBDEV control includes several elements.

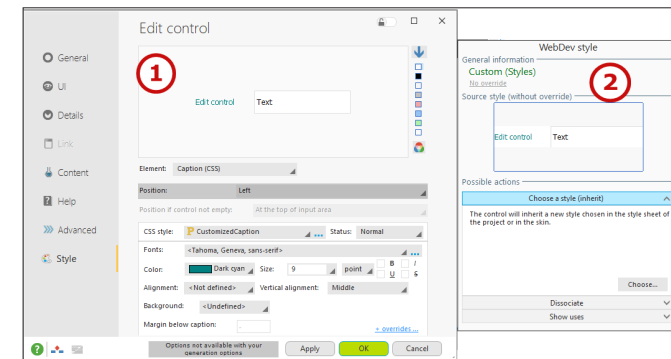
- ▶ Let's consider the edit control for example. This control includes three elements:
 - the caption.
 - the input area.
 - an additional area containing the caption and the input area.



- ▶ How are the styles applied to the edit control? The mode for applying styles depends on the type of style used (WEBDEV or CSS).
 - **The CSS styles in WEBDEV will be applied to the control elements.**
For the edit control, a specific CSS style can be applied to the caption of edit control or to the input area.
 - **The WEBDEV styles will be applied to the entire control.**
A WEBDEV style includes several CSS styles.
For example, on the edit control, the WEBDEV style contains:
 - The CSS style for the caption,
 - The CSS style for the input area,
 - The different style options for the global area of control.

Discovering the styles via an example

- ▶ To discover the styles in WEBDEV, a simple example was prepared for you:
 1. Start WEBDEV 25 (if not already done).
 2. Display the WEBDEV home page if necessary (Ctrl + <+>).
 3. In the home window, click "Tutorial" and select "Styles (Answer)".
 4. Open the "PAGE_Styles" page in the editor.
 5. Display the description window of "Name" edit control (display the popup menu of control and select "Description").
 6. Select the "Style" tab. This tab is used to define the style characteristics of control.
 7. The following window is displayed:

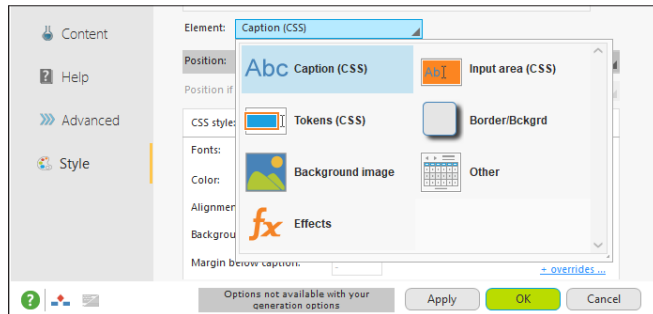


This window allows you to manage the CSS styles (section 1 of window) and the WEBDEV styles (section 2 of window) for the control. Let's take a look at the features of this window.

Selecting the CSS style of a control

► To select the CSS styles of a control:

1. In the "Style" tab of the description window of control, in the "Element" combo box, choose the element with the requested style.



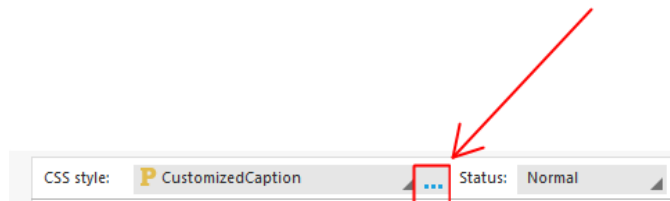
Remark: All elements flagged "(CSS)" can use a CSS style directly: all you have to do is select its name.

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

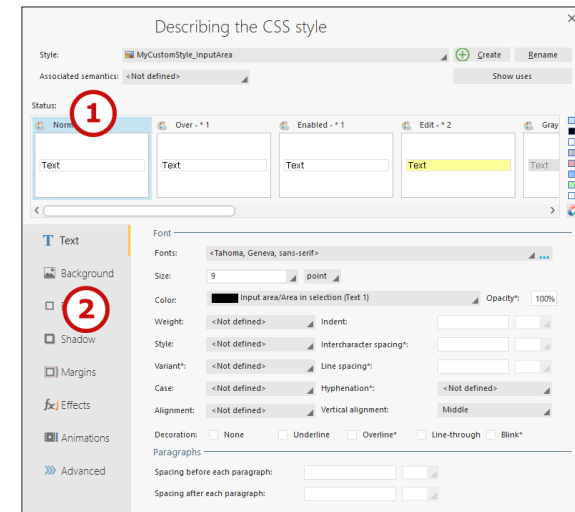
Editing the CSS style of a control

► To create or edit a CSS style:

1. In the "Style" tab of the description window of control, click "... " 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 CSS styles of project.

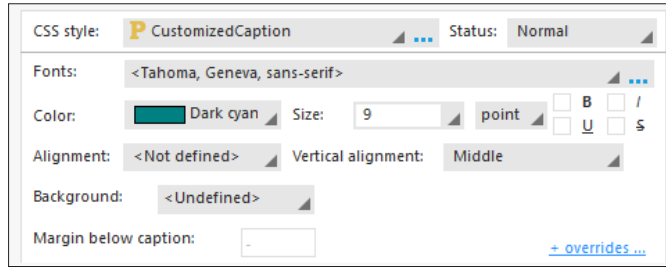
► Let's study this window. This window includes 2 sections:

- Section 1. The status of control onto which the style will be applied: normal, rollover, active, ...
- Section 2. The CSS properties associated with the selected status. Each CSS property can be modified for each status.

► Let's check the characteristics of the CSS style named "MyCustomStyle_InputArea":

1. Select the "MyCustomStyle_InputArea" style 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 "Edit" status (the 4th status of section 1).
5. In the "Background" tab, the background color of input area is pastel yellow. **The CSS style contains the style for each one of the states.**
6. Close the description window of CSS style.

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



- Font,
- Size,
- Bold, Italic, Underlined, Strikeout,
- Vertical and horizontal alignment,
- Background color.



Caution

These options are used to quickly override 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 override all the other CSS style properties.

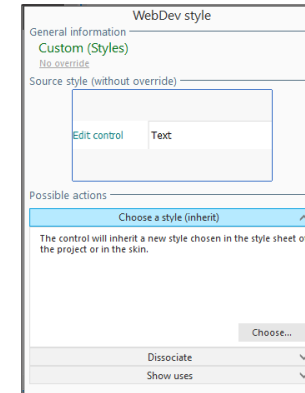
Let's now discover the WEBDEV styles.

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 control.

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



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

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

The project skin defines a WEBDEV style for each type of control. This WEBDEV style can be used "as it is" or it can be modified on some control elements.

For example, to specify that the caption must be bold on an edit control (and on this one only), you must:

- Select the "Caption (CSS)" element in the bottom area of "Style" tab.
- Select "Bold" in the bottom area.

This modification will not dissociate the WEBDEV style from the control: the Bold property will be overridden 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 override.



Remark


To see the overridden elements, simply click the link that indicates the number of overrides performed for the control, at the top of the WEBDEV style management panel.

Implementing styles: a practical example

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

Remark: We will be using an edit control but the principle for handling styles can be applied to all controls available in WEBDEV.

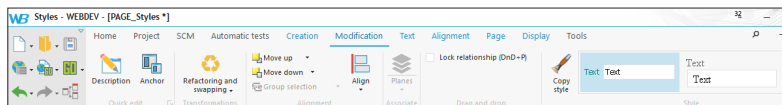
► To create the edit control:

1. On the "Creation" pane, in the "Usual controls" group, click .
2. Click the position where the control will be created in the page.

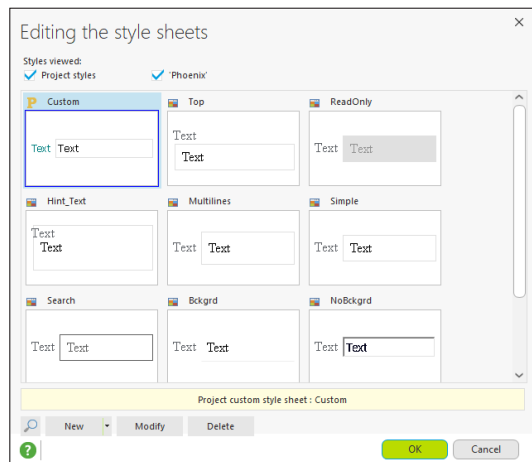
This control is associated with a WEBDEV style (it is the default WEBDEV style defined in the skin chosen when creating the project).

► Three methods can be used to select the WEBDEV style of a control:

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



- By displaying the popup menu of control and by selecting "Choose a WEBDEV style".



► This last method will be used to associate the control that was just created with the "EDT_Custom" style. This style is used by all other page controls.

1. Select the control that was just created.
2. Display the context menu of control and select "Choose a WEBDEV style".
3. In the window that appears, select the "Custom" style.

4. Validate the window for style selection. The color of control caption changes: the style is applied.

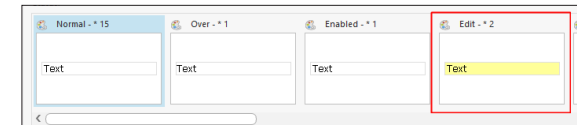
► Save the page (Ctrl + S) and run the page test ( among the quick access buttons).

1. The page is displayed in the browser.
2. Enter in input in the new edit control: the input area is colored in yellow.
3. Close the browser.

Overloading a CSS style for a control

► We are now going to modify the style of the input area found in the control:

1. Display the description window of created control.
2. In the "Style" tab, choose the "Input area (CSS)" element.
3. Click the link "More overrides" to override the CSS of this element.
4. Select the "Edit" status.



5. In the "Background" tab, change the background color. Choose a purple color, for example.
6. Validate. In the WEBDEV styles pane, the link (top right) indicates "1 override".
7. By clicking it, you can see that the background color of the editable inside area was overridden.
8. Validate the control description window.


► Save the page (Ctrl + S) and run the page test ( among the quick access buttons).

1. The page is displayed in the browser.
2. Move the cursor from a control to another one: the background color is yellow for all controls except for the control that was just created.
3. Close the browser.

Modifying a CSS style for all controls

► We are now going to modify the CSS style of input area in order to change the text color. This modification will be performed for all edit controls of the page.

1. Display the description window of created control (Alt + Enter).
2. In the "Style" tab, choose the "Input area (CSS)" element.
3. Click the [...] button on the right of the name of CSS style. The window for editing the CSS style is displayed.
4. Select the "Normal" status.
5. In the "Text" tab, change the color. Choose a green color for example.
6. Validate then validate the description window of control.

- ▶ Save the page (Ctrl + S) and run the page test ( among the quick access buttons).
 1. The page is displayed in the browser.
 2. Type text in the different edit controls of the page: the text is displayed with the new selected color. This modification was taken into account even on the control that was created beforehand and whose style was overridden.
 3. Close the browser.

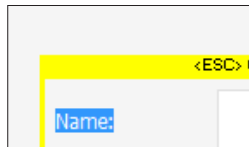
Styles and rich text

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

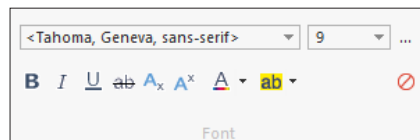
The rich text allows you to use different styles for the different words found in the control caption. Therefore, for the caption of an edit control, you have the ability to define that part of the caption will be using the default color while another part will be using a different color.

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" edit control.
 2. Press the SPACE or ENTER key on the keyboard.
 3. The caption text becomes editable.



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



- Click the **A^x** button to switch the star in exponent.
 - Change the text color to red with the **A** button.
7. Exit from the edit mode (Esc key).
 8. The red asterisk appears in the control caption.

LESSON 7.2. THE COOKIES

This lesson will teach you the following concepts...

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



Estimated time: 10 mn

What is a cookie?

A cookie is an easy way to temporarily store an information on the computer of Web user. This information can be read again later by the site that created it.

This allows you to avoid asking for details already supplied during a previous visit and to propose custom pages



Internet

A cookie has an expiration date (30 days after its creation by default). It is automatically destroyed by the browser of Web user when its lifetime is exceeded.

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

A cookie is used to store on the computer of Web user various information such as the user name, the pages displayed by the user, the date of his last connection, the backup of his options, ...

This information, saved in the format of cookies, will be read by the site during the next connection of Web user. Therefore, the site will be able to propose custom information to the user:

- advertising banner related to subjects looked up during the last connection,
- custom home page with the user name and the date of last connection,
- special offers corresponding to the searches performed during the last visit, ...



Remark

Cookies are not encrypted when they are stored: we advise you not to use them to store sensitive information.

What is a cookie made of?

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

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

- Name of cookie, used by the site to identify the cookie.
- Text of cookie, corresponding to the information written by the site: pages displayed, details supplied by the Web user, ...
- Expiration date after which the cookie is not valid anymore (it will be automatically deleted).
- Name of Internet domain that created the cookie.

Practical example

- ▶ To check the management of cookies, we are going to import the unit example named "The cookies" into the "Full_WEBDEV_Site" project.
 1. Display the WEBDEV home page (Ctrl + <).
 2. In the home page, click "Tutorial" and select "Full WEBDEV Site (Answer)".
- ▶ To open a unit example:
 1. Display the WEBDEV home page (Ctrl + <).
 2. Click "Open an example".
 3. In the search control, type "Cookies". The unit example named "The Cookies" appears in the window.
 4. Click the name of unit example.
 5. The page corresponding to the unit example is displayed in the editor.
 6. Save the page.

How to use the cookies?

Cookies can be used according to two different modes:

- Cookies used in Server code.
- Cookies used in Browser code.

To use cookies, WEBDEV includes 2 functions that can be used both in server code and in browser code:

- **CookieWrite**: is used to send a cookie when displaying the HTML page in the browser of Web user.
- **CookieRead**: is used to retrieve the value of a cookie saved on the computer of Web user.

The "PAGE_Cookies" 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 TRANSACTIONS AND PAYMENT

This lesson will teach you the following concepts...

- Securing the information and the pages via TLS/SSL.
- Secured payment.

 Estimated time: 15 mn

Securing the information and the pages via TLS/SSL

Overview

By default, the data exchanged between the computer of 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 is not longer exchanged via the HTTP protocol but via the HTTPS protocol (for example : <https://customers.mywebdevsite.com/customers>).

Implementing secure transactions via the TLS/SSL protocol

To implement secure transactions via TLS/SSL, you must install a certificate on the Web server and configure the Web server.

Two different methods can be used to get a certificate:

1. A purchase beside a certified organism.
2. The generation of a self-signed certificate.

See the online help for more details, keyword: "SSL".

Transactions secured by TLS/SSL in a WEBDEV site

The secure mode is implemented when displaying the page that requires to be secured (page for 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 actions will be performed in secure mode (which means encrypted), regardless of the objects used (link, table, loop, clickable image, ...).

See the online help for more details, keyword: "SSLActive".

Secure payment

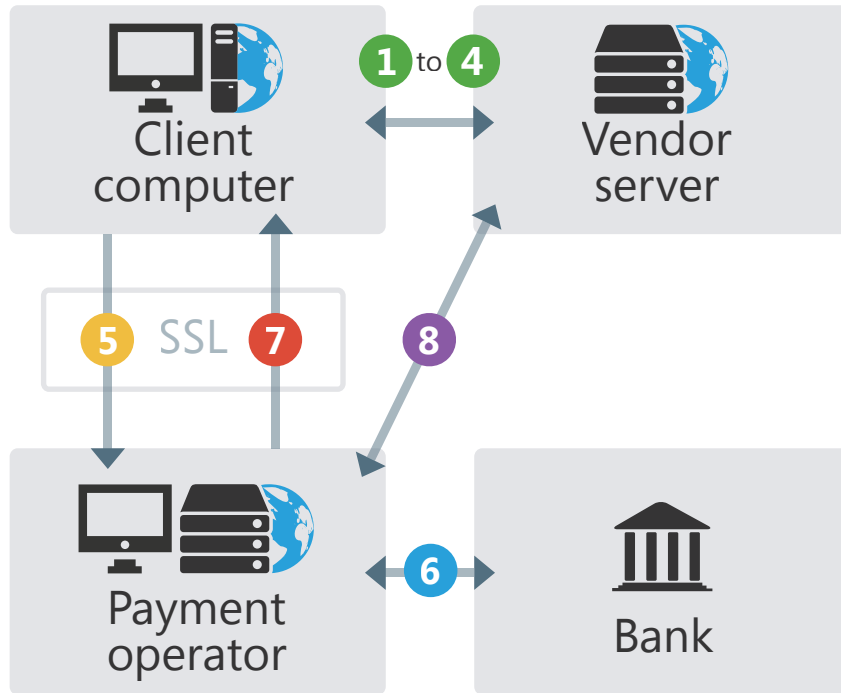
Overview

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

The secure payment is an essential feature for an e-commerce site. The payment solution must reassure the Web user (the "customer") and must guarantee the payment to the 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:



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

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

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

Caution! Important: this domain evolves very quickly. Before implementing a payment solution, always check the latest solutions proposed by the different providers!

System for secure payment in a WEBDEV site

General steps

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 WEBDEV home page (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" pane, select the "Example" configuration.
 6. You can test this example.

LESSON 7.4. ANCHORING AND ZONING

This lesson will teach you the following concepts...

- What is an anchor?
- Creating a page in Zoning mode.
- Implementing the anchors.



Estimated time: 30 mn

Overview

WEBDEV allows you to manage two types of anchors:

- The anchor to content: An anchor to content is used to define the behavior of control or area when modifying its content. Will the control keep the same size? Will it adapt to its content?
- The browser anchor: A browser anchor is used to define the behavior of a control or area when resizing the browser. Will the control be enlarged? Will it move?
The browser anchor is used to define the behavior of pages during the display on specific resolutions (tablets, phones, ...).

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

Let's play with anchors right now with an example. In this lesson, we will use browser anchors only. We are going to create a project containing a page used to implement the browser anchors and to present their benefits.


- ▶ To create a project in WEBDEV:
 1. Start WEBDEV (if not already done).
 2. Display the WEBDEV home page if necessary (Ctrl + <).
 3. In the home page, click the "Create a project" button and select "Site". The project creation wizard starts.
 4. Keep the default options of the different steps of the project creation wizard, until you reach "Description - General information".
 5. Type the project name ("Anchors") and validate the different steps of the wizard with the default options.



Answer

A corrected example can be accessed at any time to check the validity of operations performed.
To open this corrected project, in the home page, click "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 new element window, click "Page" then "Page".
Remark: To display the window for creating a new element, click  among the quick access buttons.
 - The page creation wizard starts.
 - Click "Blank" and validate the wizard.
 2. The window for saving the page is displayed.
 3. Enter "Anchors" as title. The name ("PAGE_Anchors") is automatically proposed.
 4. Validate.

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

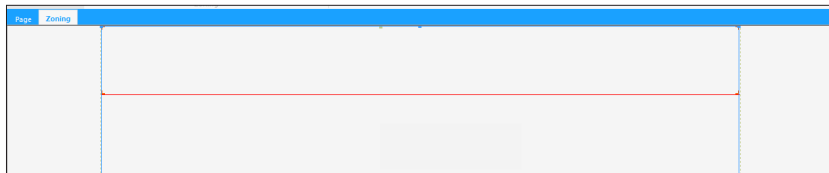
- A header (at the top on the entire width).
- A menu on the left (on the entire height).
- A content area in the middle.

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

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

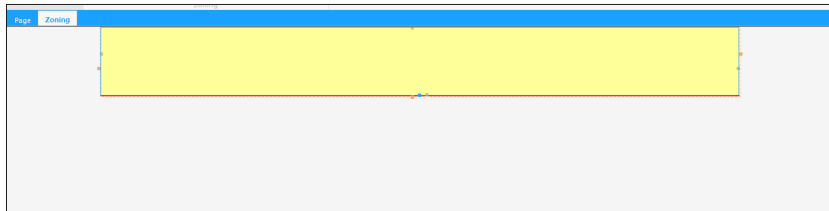
► To create a first title area at the top of the page:

1. On the "Page" pane, in the "Edit" area, expand "Zoning" and select "Divide horizontally".
2. The mouse cursor turns into a pen.
3. Click at the top of the page and draw an horizontal line. The area is created when releasing the mouse button.



► To view this area, we are going to associate it with a background color:

1. Display the description window of area: select "Description of 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 area.



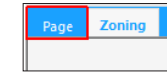
► We are going to perform the same operations to split the bottom area of our page:

1. On the "Page" pane, in the "Edit" area, expand "Zoning" and select "Divide vertically".
2. The mouse cursor turns into a pen.
3. Click in the middle of bottom area and draw a vertical line.

► To see 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:

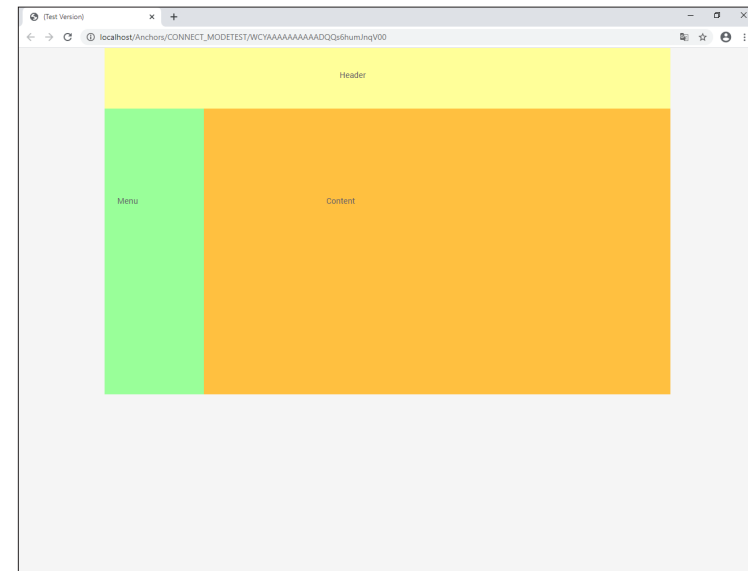
1. Exit from the "Zoning" edit mode by clicking the "Page" tab at the top of the page (below the ribbon).



2. On the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Simple static".
3. Click inside the yellow area. The Static control is automatically created.
4. Modify the caption (Space key on the keyboard): the control caption is "Header".
5. Repeat the operations 2 to 4 to create:
 - a Static control displaying "Menu" in the green area.
 - a Static control displaying "Content" in the orange area.

► Save the page (Ctrl + S) and run the page test ( among the quick access buttons).

1. The page is displayed in the browser.
2. Resize the browser: the page remains centered but the areas are not resized.



We are going to change this operating mode in order for the areas to adapt to the browser size. We will be using the anchors.

► Close the browser.

Implementing the anchoring

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

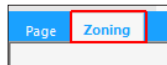
We want to get the following behavior when enlarging the browser:

- The header area (yellow) must be enlarged in width.
- The menu area (green) must be enlarged in height.
- The content area (orange) must be enlarged in width and in height.
- For the controls, we want them to remain centered in relation to their area.

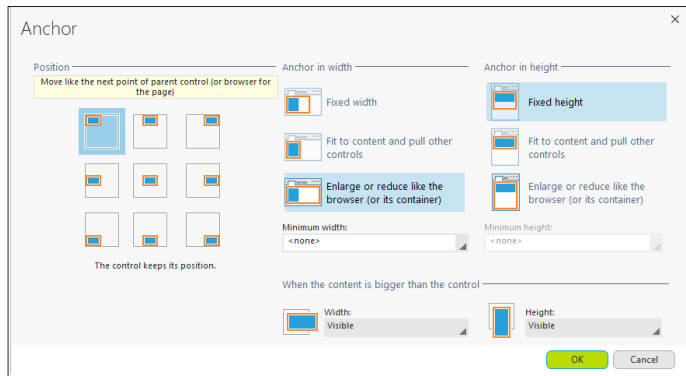
We are now going to perform all necessary operations.

► To implement the anchoring of areas:

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

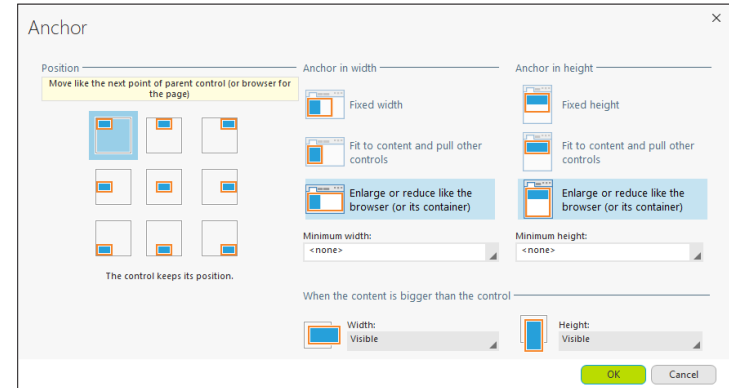


2. Select the yellow area.
3. Display the popup menu and select "Anchor". The window for anchor management is displayed.
4. The area must be enlarged in width. Select:
 - Position: The control keeps its position.
 - Anchor in width: Enlarge or reduce like the browser.
 - Anchor in height: Fixed height.



5. Validate.
6. Select the green area, display the popup menu and select "Anchor".
7. The area must be enlarged in height. Select:
 - Position: No anchor.
 - Anchor in width: Fixed width.
 - Anchor in height: Enlarge or reduce like the browser.


8. Validate.
9. Select the orange area and display the window for anchor management. The area must be enlarged in width and height. Select:
 - Position: No anchor.
 - Anchor in width: Enlarge or reduce like the browser.
 - Anchor in height: Enlarge or reduce like the browser.

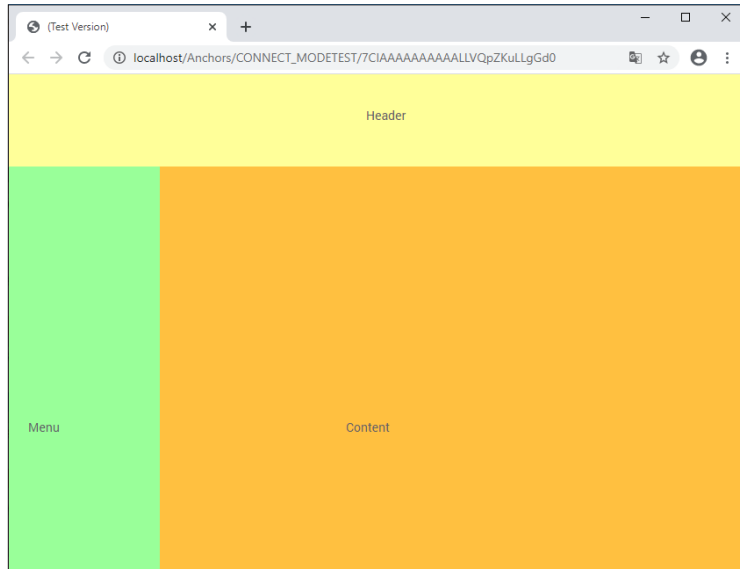


10. Validate.
- To implement the anchoring of controls:
 1. Display the page in Page mode: click the "Page" tab at the top of page.

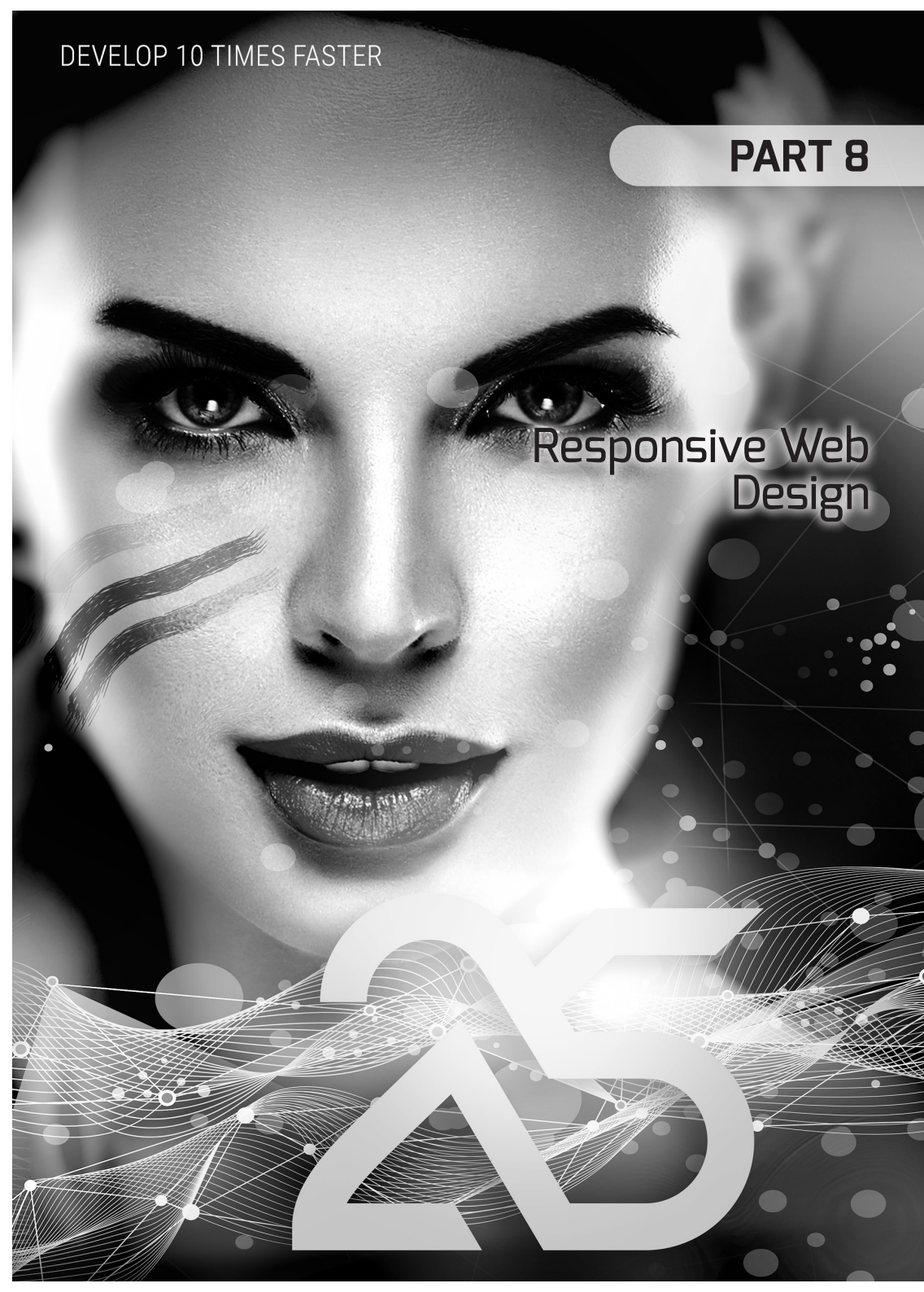


2. Select "Header".
3. Display the popup menu and select "Anchor".
4. The control must be centered in width:
 - Position: Horizontally centered.
 - Anchor in width: Fixed width.
 - Anchor in height: Fixed height.
5. Validate.
6. Repeat the same operations for the "Menu" control and for the "Content" control:
 - The "Menu" caption must be vertically centered.
 - The "Content" caption must be centered.

- ▶ Save the page (Ctrl + S) and run the page test ( among the quick access buttons).
- 1. The page is displayed in the browser.



- 2. Resize the browser: the areas adapt to the available space in the browser.
- ▶ Close the browser.



LESSON 8.1. RESPONSIVE WEB DESIGN

This lesson will teach you the following concepts...

- What is Responsive Web Design?
- Benefits.
- Concepts.



Estimated time: 5 mn

What is Responsive Web Design?

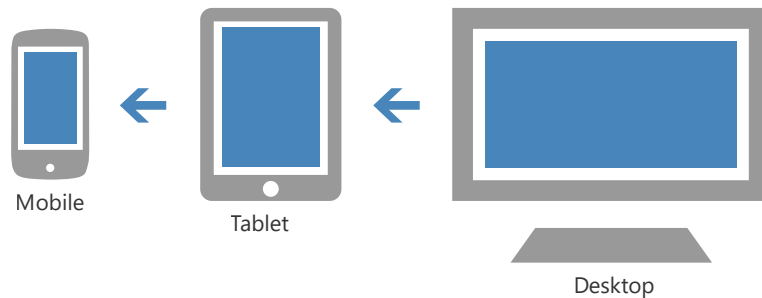
Responsive Web Design is a method for developing Web sites allowing the user to get the best possible reading quality regardless of the platform used (mobile devices, tablets, PC, ...).

Benefits

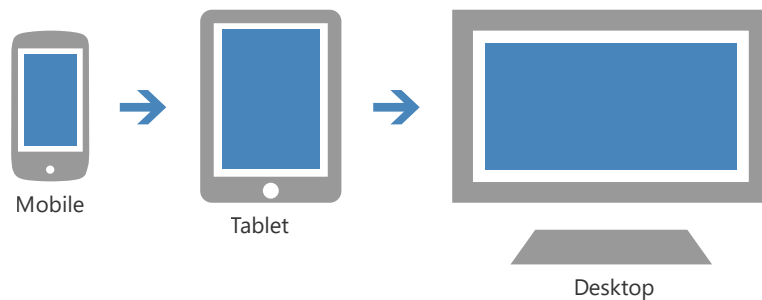
- The site can be displayed with a high-quality level on any platform (from mobile device to desktop via the tablet).
- The site is better referenced. Indeed, the search engines give priority to the sites that display properly on all platforms.
- The maintenance is simplified: there is a single page to maintain for all devices.

The concepts in WEBDEV

The principle of Responsive Web Design development with WEBDEV consists in defining the page for the Desktop and adapting it to smaller screen sizes. It is the principle of "Desktop First".



However, you can also do the opposite: design the Mobile site and modify it by adding features to get the Desktop site. It is the principle of "Mobile First".



Several WEBDEV tools are available to help you. We will discover them in the next lesson.

LESSON 8.2. RESPONSIVE WEB DESIGN AND WEBDEV

This lesson will teach you the following concepts...

- Available WEBDEV tools.
- Positioning grid.
- Resolution slices.
- Overrides.

 Estimated time: 30 mn

Overview

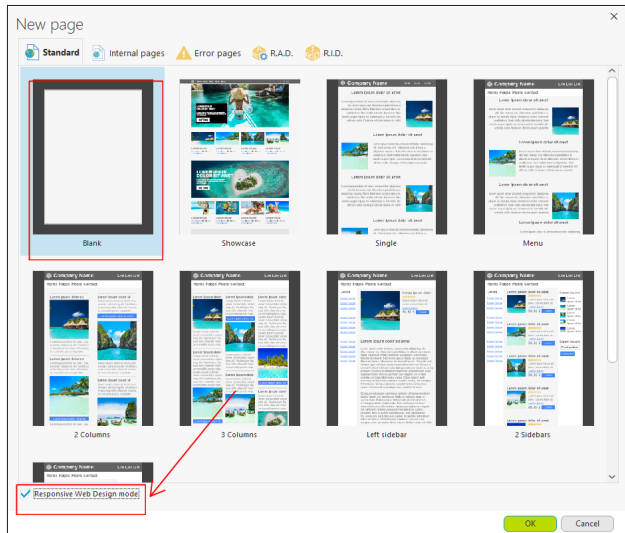
An example project will allow you to discover the methods that can be used in WEBDEV to create sites in Responsive Web Design mode. This allows you to perform the operations and to better understand the available tools.

- ▶ A simple example was prepared for you:
 1. Start WEBDEV 25 (if not already done).
 2. Display the WEBDEV home page if necessary (Ctrl + <).
 3. Open the "WW_Responsive" project. To do so, in the home page, click "Tutorial" and select "Responsive Web Design (Exercise)".

My first page in Responsive Web Design mode

Creating a page

- ▶ In WEBDEV, the method for creating a page in Responsive mode is identical to the method for creating the other pages:
 1. Create a new blank page.
 - In the window for creating a new element, click "Page" then "Page".
 - Remark: To display the window for creating a new element, click  among the quick access buttons.
 - The page creation wizard starts.
 - Click "Blank" and check "Responsive Web Design mode".



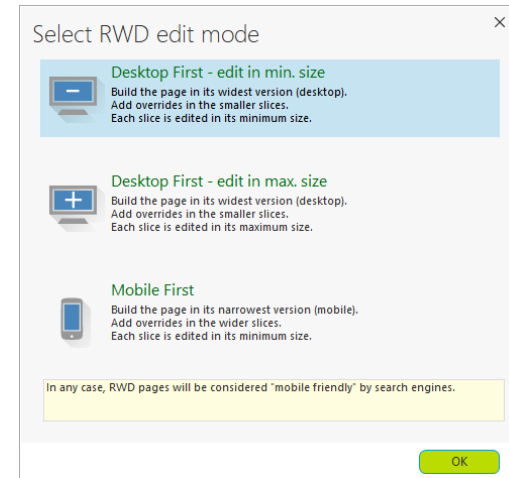
- Validate the wizard.



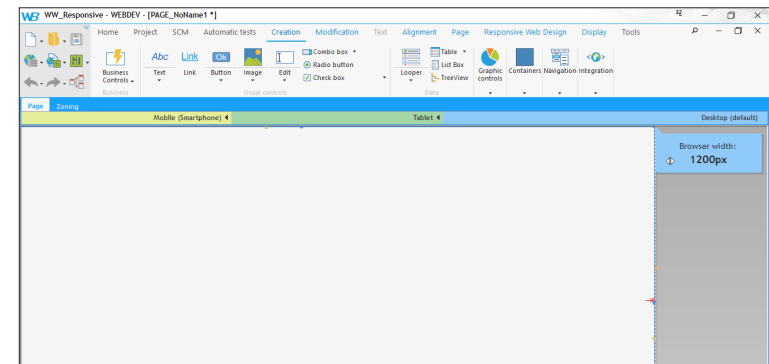
Remark

All page templates included with WEBDEV are available in Responsive mode. Simply check "Responsive Web Design mode". We are now going to create a blank page to understand the principle of Responsive Web Design.

2. WEBDEV displays a window allowing to choose the main width used for Responsive Web Design. In this example, we are going to create a page for the "Desktop" mode.



3. Select "Desktop First - edit in max. size" and validate.
4. The page save window appears. Type the page title: "Responsive". The name ("PAGE_Responsive") is automatically proposed.
5. Validate.
6. The page is created in the editor. This page is displayed in "Desktop" mode.



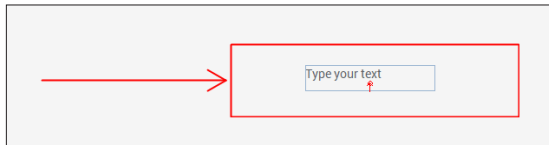
Fluidity and Responsive Web Design

Let's see how the page behaves in the WEBDEV editor by creating a Static control.

- ▶ To create the Static control:
 1. On the "Creation" pane, in the "Usual controls" group, click "ABC".
 2. Then, click inside the page to create the control.

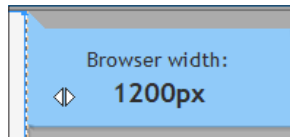


3. Press F7 three times to display the border of controls.



The first principle of responsive is fluidity: when the browser is reduced, the page controls are reduced accordingly.

- ▶ To check this fluidity in the editor:
 1. Use the cursor found at the top of the page to reduce the browser size.



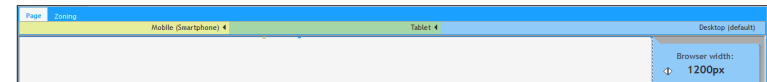
2. The Static control that was just created is reduced accordingly.

Resolution slices

The cursor that was just used belongs to the resolution slices, second principle of WEBDEV Responsive.

The resolution slices are used to materialize the resolutions of different devices.

- ▶ Let's study the operating mode of these resolution slices.
 1. In the "WW_Responsive" project, open the PAGE_Adaptations page.
 2. Three slices appear above the page:



- In blue, the Desktop slice, for displaying on a desktop screen,
- In green, the Tablet slice corresponding to the display on tablets,
- In yellow, the Mobile slice for the display on Smartphones.

The resolution slices allow you to define the layouts that will be used for each type of device.



Remark

You can add custom resolution slices to manage the different devices more efficiently. To change resolution slice, you can use the cursor or click the resolution slice directly.



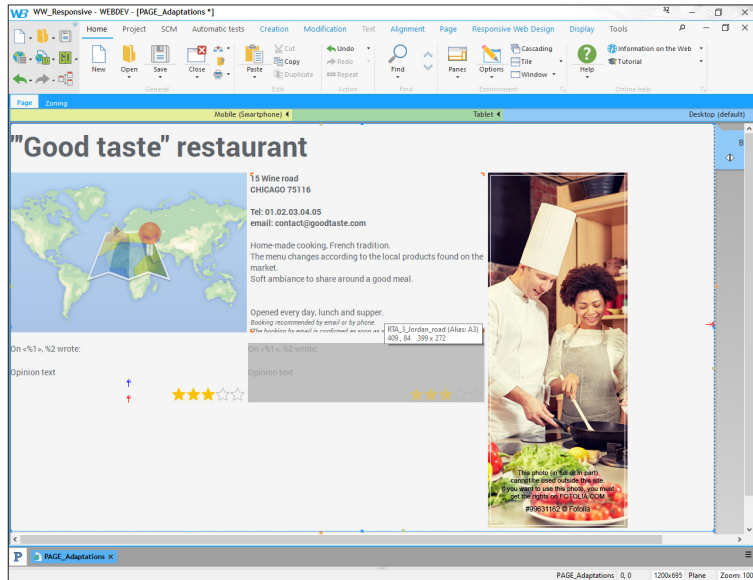
Caution

Responsive WEB Design with WEBDEV is using the concept of "Desktop First" by default. In this mode, the controls are always created in the Desktop slice. Then, the interface must be adapted to smaller slices.

The Desktop slice corresponds to the base of the page, without adaptation.

To perform adaptations in the other resolution slices: all you have to do is perform the modifications in the editor directly.

► Let's study the PAGE_Adaptations page.



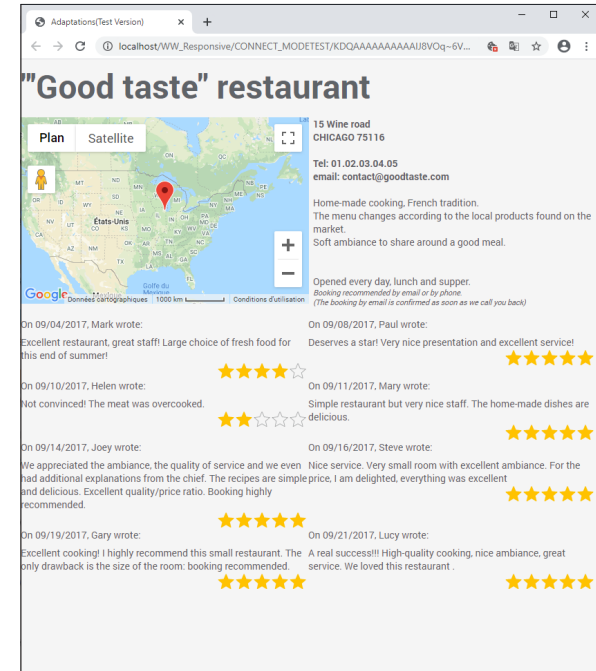
► This page contains different controls:

- a title at the top,
- a Map control to indicate the restaurant location,
- a Static control containing the restaurant address,
- an Image control on the right to display an advertising banner,
- a Looper control at the bottom to display the opinions.

► Run the page test (among the quick access buttons).

- The page appears in test as it was in the editor in Desktop mode.
Remark: This page is using a Map control. In test as in deployment, it is necessary to use a Google key. Otherwise, an error occurs in the Map control. See the online help to get this key. If you own a key, specify it in the "Project" tab of the description window of project.
- Reduce the size of your browser to see the change of slice.

► Some page controls move or disappear according to the resolution slices displayed. For example, the advertising image is shown in the Desktop slice but it becomes invisible in the smaller slices.



How to adapt the controls to the different slices?

Performing adaptations regarding the positioning and/or the size is child's play! The modifications are performed in the editor directly (like any other modification). In fact, these adaptations are overrides.



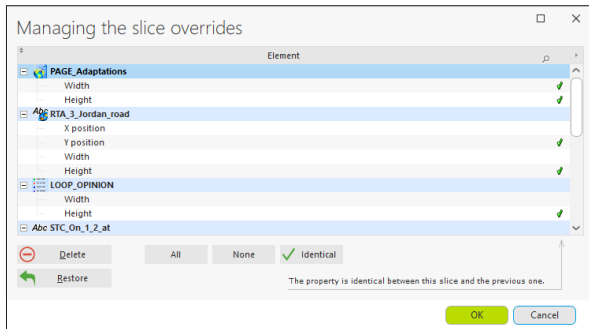
Caution

In order for the sizes to be respected when reducing the browser, we advise you to click the resolution slice to modify **before** any modification. The default size of the selected slice will be taken as reference and no error will occur at runtime.

It is possible to override several properties for each resolution slice. They can be identified in the control description window via the picto.

For example, for the Image control, a different image file can be used according to the resolution slice.

- ▶ To override the size and position of controls, you must:
 1. Select the requested slice.
 2. Perform the modification in the editor.
- ▶ To see the list of overrides for a resolution slice:
 1. Select the requested resolution slice (for example, for the tablet slice, click inside the tablet slice).
 2. Display the popup menu of the slice bar (right click) and select "List of slice overrides".
 3. The list of overrides performed is displayed:



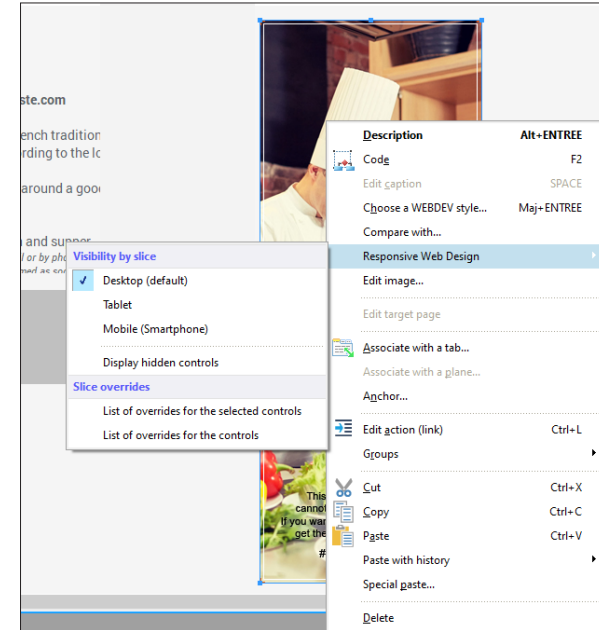
Remark: This window allows you to delete the overrides in order to restore the control's default operating mode.

Special case: Control visibility

The visibility of a control is defined in the "UI" tab of the control description window. In Responsive mode, this option cannot be overridden by slice.

- ▶ To define the visibility of a control by slice, you must:
 1. Select the control.
 2. Display the popup menu of control (right mouse click).

- 3. Select "Responsive Web Design .. Visibility by slice".

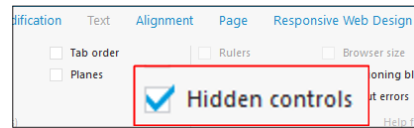


- 4. Keep checked only the slices where the control must remain visible.

Remark: Only the visibility defined in the description window of control can be modified by programming. The visibility overrides performed by resolution slice cannot be modified by programming.

Remark

To see the hidden controls (to modify their position if overlapped for example), on the "Display" pane, in the "Help for edit" group, check "Hidden controls".



LESSON 8.3. ADVANCED CONCEPTS

This lesson will teach you the following concepts...

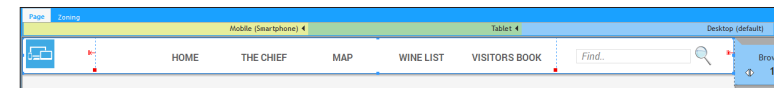
- Specific control: Navigation Bar control with "Hamburger" menu.
- The anchors.
- Managing the order of controls.

 Estimated time: 30 mn

Controls specific to Responsive Web Design

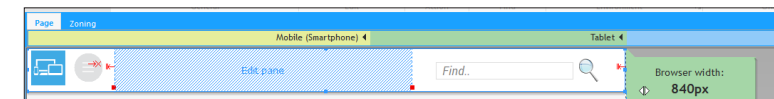
WEBDEV includes a specific control for Responsive Web Design: the Navigation Bar control. The Navigation Bar control is used to create a menu area that can easily adapt to the browser size. This area can contain all types of controls.

- ▶ Let's study the operating mode of Navigation Bar control.
 1. Open the "WW_Responsive" project if necessary. To do so, in WEBDEV's home page, click "Tutorial" and select "Responsive Web Design (Exercise)".
 2. Open the PAGE_NavigationBar page.
- ▶ The Navigation Bar control is the control found at the top that contains the menu.

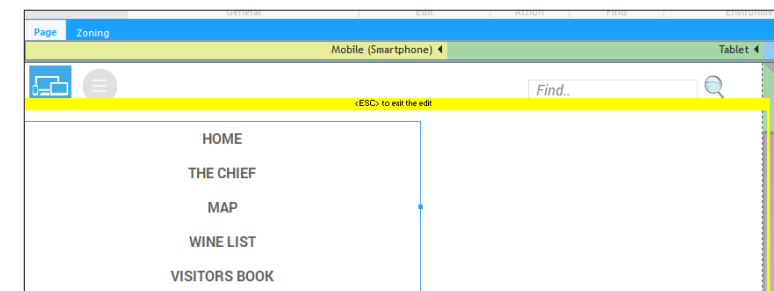


This Navigation Bar control is divided into 3 sections:

- On the left, the site logo.
 - In the middle, the site menu.
 - On the right, a search control with an Image control to start the search.
- ▶ Let's switch to Tablet mode:
 1. Click the header of the Tablet slice.
 2. Click the Navigation Bar control to select it: the middle section is hatched.



- ▶ From the Tablet slice, the Navigation Bar control makes the middle section invisible and adds a Hamburger menu in the left section. The "hamburger" menu expands a popup containing the controls of middle section. This popup can be customized, all you have to do is click the central pane. Then, this popup can be modified according to your own needs.
- ▶ Let's study the content of middle section:
 1. Click inside the hatched middle section. The popup is displayed:

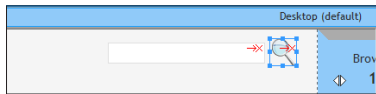


- You can see the site menu. The menu is vertical now. Indeed, the Navigation Bar control automatically overrides the Menu control so that it is displayed vertically.
- To exit from the edit mode of popup, press the Esc key.

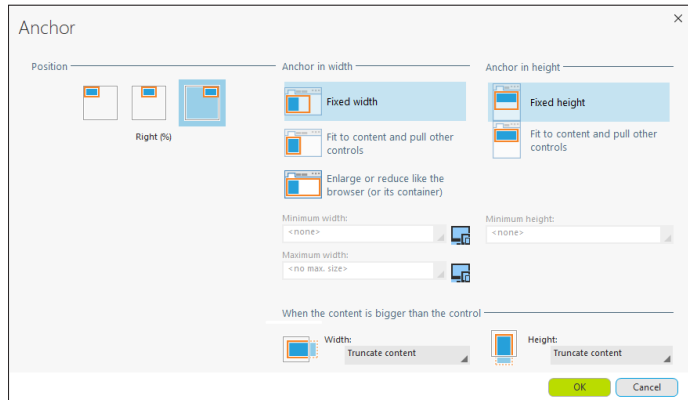
The anchors

The management of anchors was already presented in the previous chapters. However, the anchors operate differently in a Responsive Web Design page.

- Let's study the operating mode of anchors in Responsive Web Design mode.
 - Open the "WW_Responsive" project if necessary. To do so, in WEBDEV's home page, click "Tutorial" and select "Responsive Web Design (Exercise)".
 - Open the PAGE_Anchor page.
 - Select the Image control used to start the search at the top right:



- Display the popup menu (right mouse click) and select "Anchor".
- The window for anchor management is displayed:



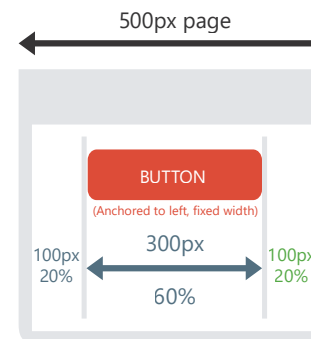
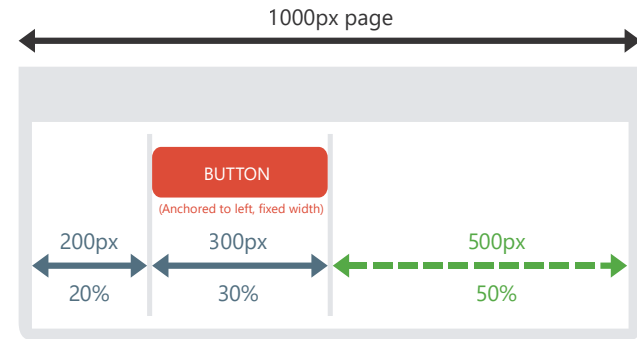
WEBDEV proposes several types of anchors:

- the position anchors,
- the anchors in width.
- the anchors in height.

Positioning anchors

The position anchor proposes the following choices:

- An anchor to left.
 - A centered anchor.
 - An anchor to right.
- Let's study these different possibilities:
 - The centered anchor must be selected in order for the control to remain centered when resizing the browser.
 - The anchor to left is used to specify that the size between the control and the left page border will always be the same proportionally to the page size. The section found on the right of control can be truncated if the browser reduction is too important.

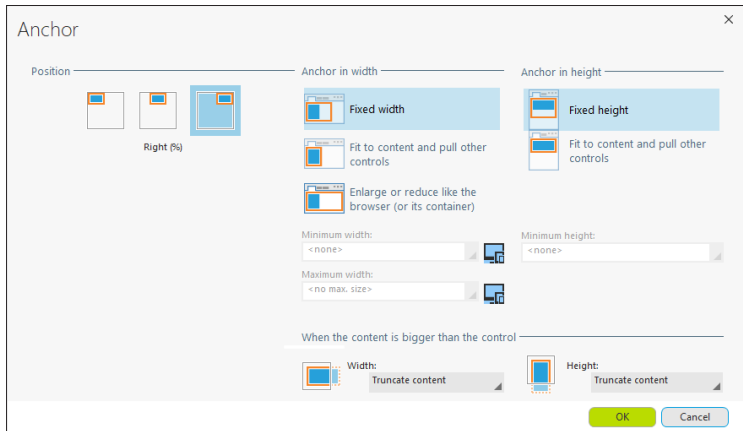


- The anchor to right performs the same action as the anchor to left, but on the right.
- In our example, the search button found in the top right corner of page is anchored to right. Meaning that the spacing between the right border of control and the right page border will always be the same proportionally to the page width.

- ▶ Run the page test (among the quick access buttons).
 - The page appears in test as it was in the editor in Desktop mode.
 - Reduce the size of your browser.
 - The area on the left of search button is reduced while the area on the right remains the same.
- ▶ Close the browser.

Size anchors

- ▶ Let's now study the size anchors.
 1. In the PAGE_Anchor page, select the edit control named "Search" beside the search button.
 2. Display the popup menu (right mouse click) and select "Anchor".
 3. The window for anchor management is displayed.
- ▶ Let's study the different possibilities:



- For the anchors in width, the options are as follows:
 - Fixed width: the control keeps its dimensions defined in the editor.
 - Fit to content and pull other controls: the control width adapts to content. In this case, you have the ability to specify the minimum width and the maximum width.
 - Enlarge or reduce like the browser: the control width is modified proportionally to the browser width. In this case, you have the ability to specify the minimum width and the maximum width.
- For the anchors in height, the options are as follows:
 - Fixed height: the control keeps its dimensions.
 - Fit to content and pull other controls: the control height is adapted to content. In this case, you have the ability to specify the minimum height.

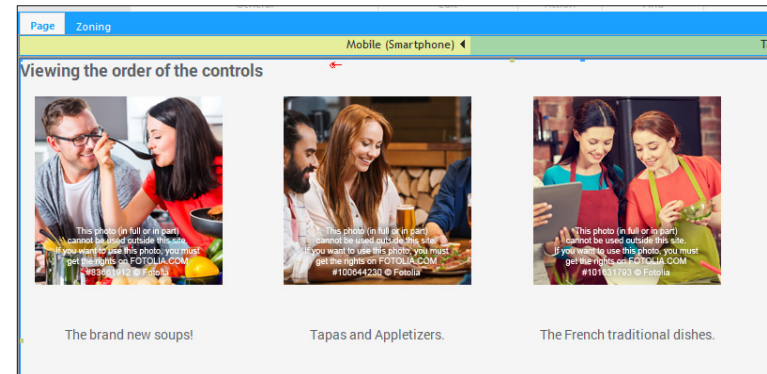
- It is also possible to define the behavior of the control when its content is larger than it is. You can:
 - Enlarge the control and push the other controls.
 - Truncate the content.
 - Enable a scrollbar that will always be visible.
 - Display a scrollbar.
- ▶ In our example, the edit control and the search buttons are defined as keeping their edit size ("Fixed width" and Fixed height" options).
- ▶ Run the page test (among the quick access buttons).
 - The page appears in test as it was in the editor in Desktop mode.
 - Reduce the size of your browser.
 - The edit control and the search button are not resized.
- ▶ Close the browser and go back to the editor.

Order of controls in Responsive Web Design mode

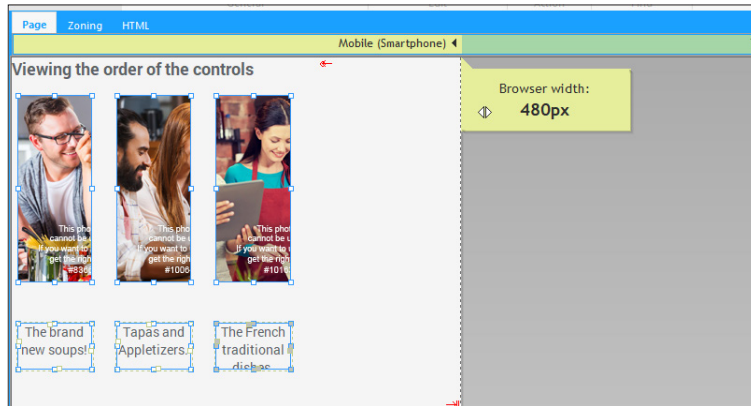
We have seen that the pages in Responsive Web Design mode adapt according to the browser size. The override mechanism by slice is used to modify the interface according to the requested slice, especially by modifying the position of the controls.

You must pay great attention to this modification of position. Indeed, in a responsive page, you cannot change the order of controls in the page.

- ▶ To understand the management of order of controls:
 1. Open the "WW_Responsive" project if necessary. To do so, in WEBDEV's home page, click "Tutorial" and select "Responsive Web Design (Exercise)".
 2. Open the PAGE_ControlOrder page.
- ▶ Let's study this page:
 1. In the Desktop slice, 3 products are presented horizontally.

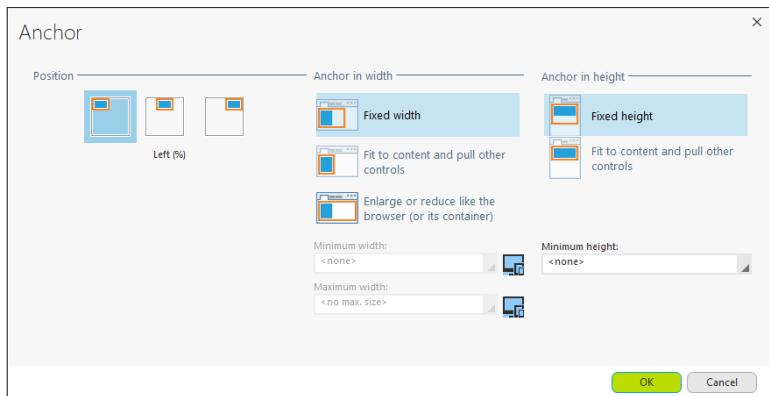


2. Click the header of the Mobile slice.



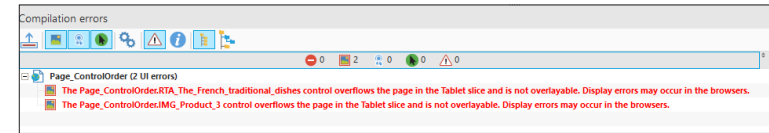
3. In the Mobile slice, the product images are very small.

- ▶ On Mobile, the controls should be the same size as in the Desktop slice and an image should be moved.
- ▶ Go back to the Desktop slice to modify the control anchors.
 1. Select the Image controls and the associated Static controls.
 2. Display the popup menu and select "Anchor".
 3. In the window that is displayed, select "Fixed width".



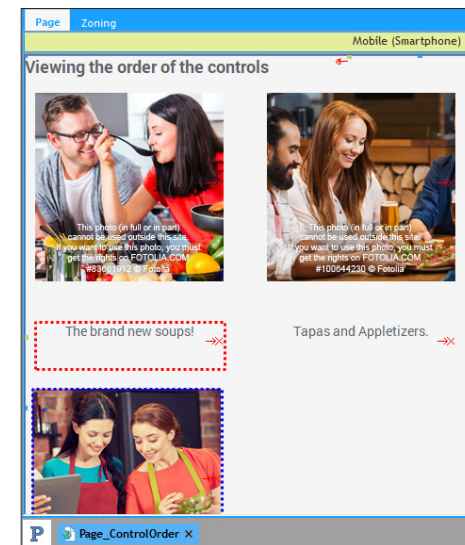
4. Validate the anchoring window.

5. Save the page (Ctrl + S). Two compilation errors occur.



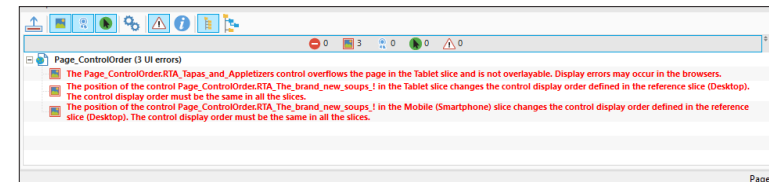
These errors indicate that the controls IMG_Product_3 and RTA_The_traditional_French_dishes overflow the page at the end of the Tablet slice.

6. Click the Tablet slice. Move the "Browser width" cursor to the left. Indeed, from a particular browser dimension, the image and caption of 3rd product do not fit in the page anymore.
7. Select these two controls in the Tablet slice and move them below the first two.



8. Save the page (Ctrl + S).

9. New UI errors occur in the pane of compilation errors.



10. These errors are errors regarding the display order.

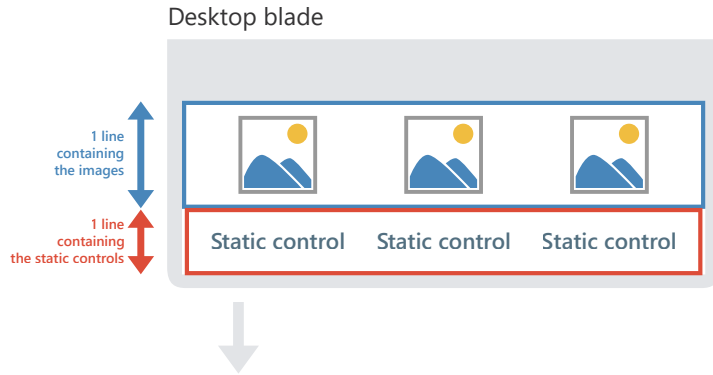
Why do these errors occur?

This use case is quite common and it is important to understand the principle used.

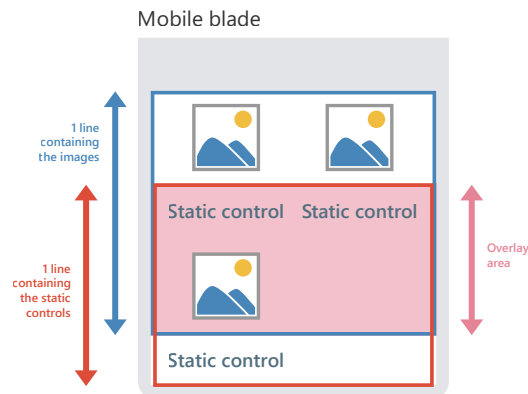
Each control is positioned in relation to its previous control.

In our example, each image is positioned in relation to the previous image. It is the same for the Static controls.

The first static on the left (number 4 in the example) is positioned in relation to the last image (number 3 in our example).

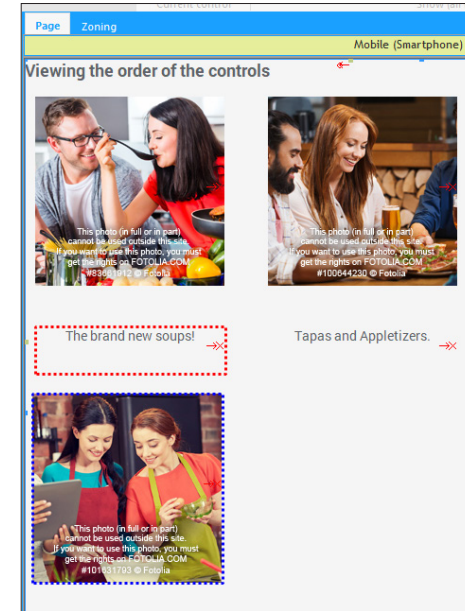


When the image and its associated static were moved into the Tablet slice (image 3 and static 6 in our example), the static 4 was positioned "above" its "reference" control (image 3, beside which it should be).



This positioning is not possible in Responsive Web Design. Therefore, new UI errors have been displayed.

To guide you, the area where the display order was changed is bordered by red dots.



- To fix this problem, simply specify the text-level semantics. In our case, each Image control must be linked to its static: simply create cells around each "product" (image + static). The cell must have a "Fixed width" anchor.



Remark

In most cases, in Responsive Web Design mode, to group controls and to get the same operating mode for each control of the group, we advise you to include them in a cell.

- ▶ To apply these corrections in our page, we are going to delete all the overrides performed in the Tablet slice:
 1. Display the Tablet slice popup menu (right-click the section header).
 2. Select "List of slice overrides".
 3. In the window that appears, select all the overrides and click "Delete".
 4. Validate the window.

- ▶ To create a cell containing one or more existing controls:
 1. Display the Desktop slice if necessary.
 2. Select the controls (the first image on the left and its static for example).
 3. On the "Modification" pane, in the "Transformations" group, expand "Refactoring and swapping" and select "Create a Cell control from the selection".
 4. Select the cell. Select "Anchor" from its popup menu.
 5. In the anchoring window, select "Fixed width" to anchor in width.
 6. Validate.
 7. Perform these operations again for the other two groups of controls.

- ▶ We are now going to reposition the controls in Tablet slice:
 1. Switch to Tablet slice.
 2. Select the cell containing the last two controls and move this cell below the first two controls.
 3. Save the page (Ctrl + S). No compilation error is displayed.



Useful features
of WEBDEV

LESSON 9.1. THE INTERNAL COMPONENTS

This lesson will teach you the following concepts...

- What is an internal component?
- Creating an internal component, step by step.
- Distributing and using an internal component.



Estimated time: 30 mn

Overview

An internal component is a group of project elements. This group is used to:

- organize a project,
- share elements between different projects (mainly via the Source Code Manager).

When an internal component is included in a project, all component elements are included in the project. The public elements can be handled in the editor directly. Furthermore, the internal component can be debugged from the project that is using it.

The projects that use an internal component have access in the WEBDEV editor to the name of objects, procedures or methods made visible by the component creator.

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

How to proceed? Don't change anything, create your pages, procedures, classes. Then, when it's done, choose the option for creating a component and that's it!

An internal component can contain code, pages, an analysis, data files, etc.

Step by step

Step 1: Creating an internal component

We are going to create an internal component used to subscribe and unsubscribe to a newsletter in a Web site.

This component includes:


- a page, used to see the list of persons who subscribed to the newsletter,
- a Web control template that contains the different controls (edit of email address, validation button, ...),
- an analysis describing the data file of subscribers.

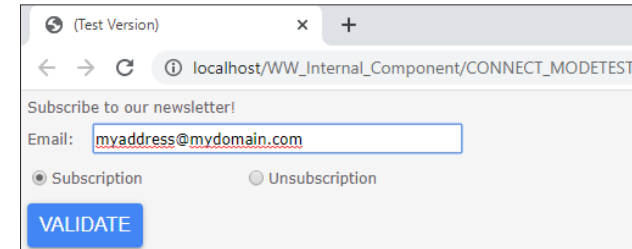
To avoid having to develop the code required for the component to operate, all necessary elements have been grouped in a project named "WW_Internal_Component". We will be using this project to create our internal component. A new project will be created later to use this internal component.

► To open the example project:

1. Display the home page if necessary (Ctrl + <).
2. In the home page, click "Tutorial" and select "Internal Component (Exercise)".
3. The project is loaded.

► We are going to test the control template of this project:

1. Open the Web control template named "TPLC_NewsletterSubscription" (double-click its name in the "Project explorer" pane).
2. Run the test of Web control template ( among the quick access buttons).
3. Type an email address.

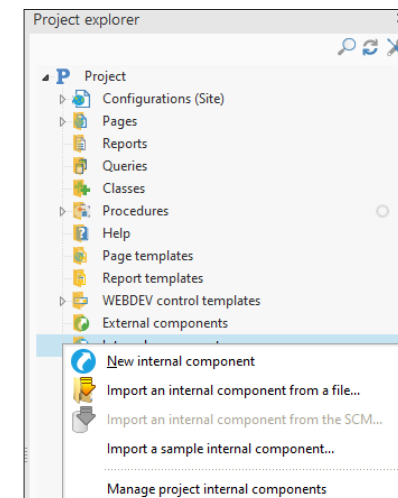


4. Click "VALIDATE" to add the address into the data file.
5. Close the browser and go back to the editor.
6. You can check the content of the data file with WDMMap (accessible in the ribbon "Tools" pane).

We are now going to create our internal component.

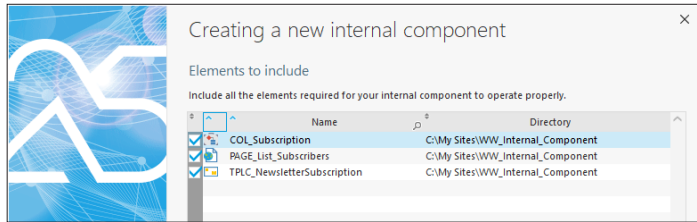
► To create an internal component:

1. In the "Project explorer" pane:
 - Select the "Internal components" folder.
 - Display the popup menu (right click).

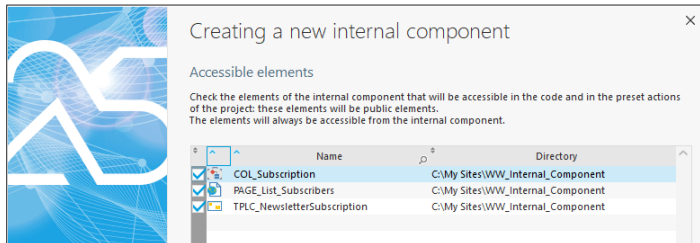


- Select "New internal component".
- The wizard for creating an internal component starts.

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

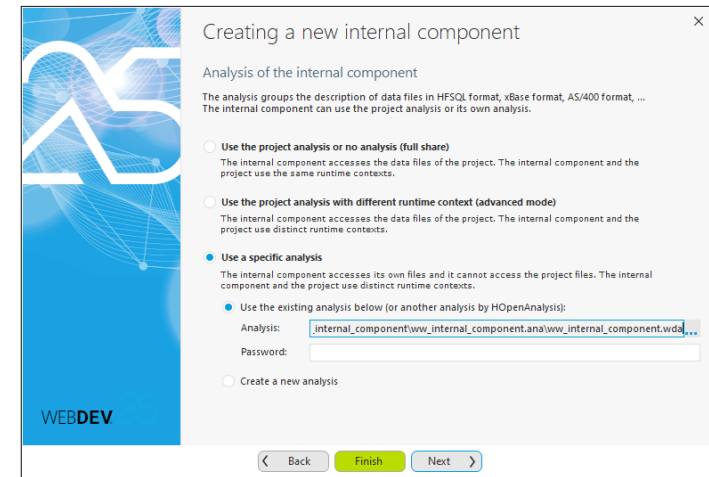


6. Go to the next step.
7. The wizard asks you to select the component elements that will be accessible from the client project. In our case, all elements must be selected.



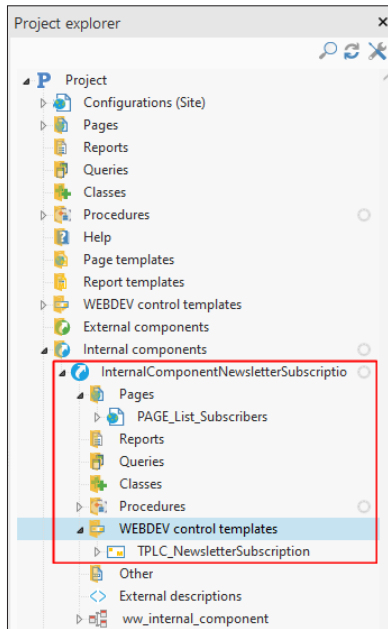
8. Go to the next step.

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" in the project subdirectory "WW_Internal_component.ana"). For example: "My Sites\My Examples\WW_Internal_Component (Exercises)\WW_Internal_Component.ana".
11. End the wizard. WEBDEV will create the internal component in a specific directory of your project.

In the "Project explorer" pane, the "TPLC_NewsletterSubscription" Web control template is no longer in the project's list of Web control templates. On the contrary, the internal component that was just created is listed in the "Internal components" 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.



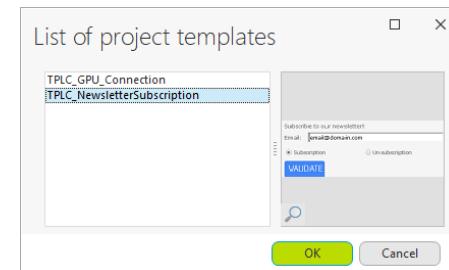
Remark

In this example, we will present a "direct" use of internal component. To share resources, we recommend that you use the internal components via the Source Code Manager (SCM). See the online help for more details (keyword: "Internal component").

In our example, we are going to import the internal component into the "Full_WEBDEV_Site" project.

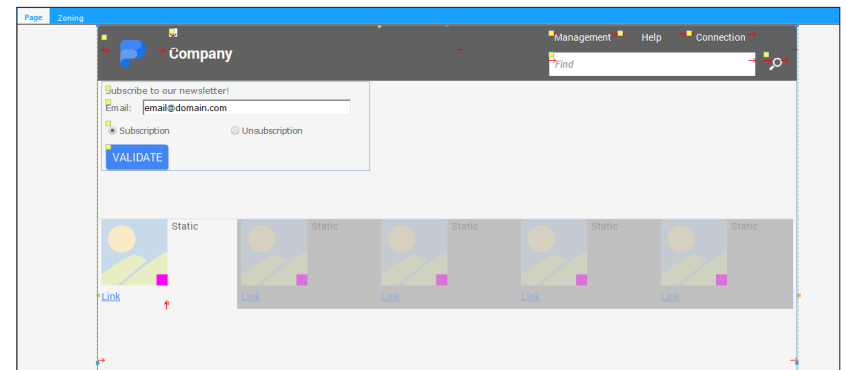
- ▶ Open the "Full_WEBDEV_Site" project.
 1. Display the WEBDEV home page (Ctrl + <).
 2. In the home page, click "Tutorial" and select "Full WEBDEV Site (With pages)". The project is loaded.


- ▶ Include the internal component in the project.
 1. In the ribbon, on the "Project" pane, in the "Project" group, expand "Import" and select "An internal component .. From a file".
 2. Select the file "My Sites\My Examples\WW_Internal_Component (Exercices)\InternalComponentNewsletterSubscription.wci". This file corresponds to the component that was created previously.
 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 site home page. To do so:
 1. In the editor, open the "PAGE_List_of_new_products" page (double-click its name in the "Project explorer" pane).
 2. Move the looper found in the page to the bottom in order to insert the control template of internal component.
 3. Select the "Creation" pane of ribbon and in the "Containers" group, click "Control template".
 4. The window for selecting the template to instantiate is displayed.

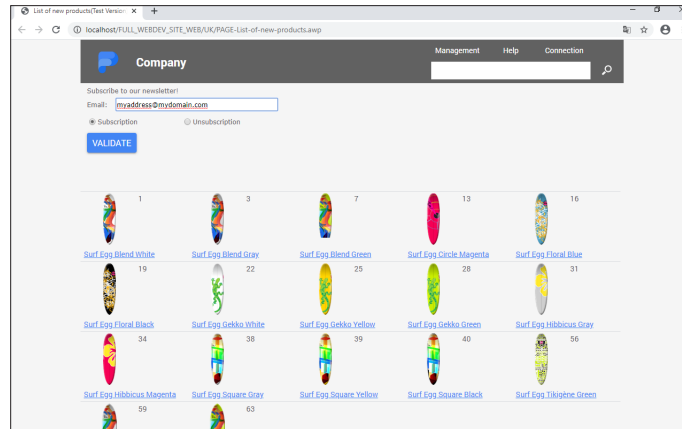


Remark: The "TPLC_GPU_Connection" template corresponds to the control template used by the user groupware that we saw in a previous lesson.

5. Select the template coming from the internal component and validate.
6. Click the location where the Web control template must be positioned in the page.



- We are going to check the operating mode of control template:
1. Run the page test ( among the quick access buttons).
 2. Type an email address.
 3. Click "VALIDATE".
 4. A message confirms the subscription.



LESSON 9.2. AUTOMATIC MANAGEMENT OF ERRORS

This lesson will teach you the following concepts...

- What is the automatic management of errors?
- Using the automatic management of errors.



Estimated time: 10 mn

Overview

WEBDEV can manage the errors automatically. This feature helps you reduce the number of code lines while centralizing the management of errors.

The use of this feature also makes the code easier to read.

Operating mode

Two operations are performed when an error is detected by a WLanguage function:

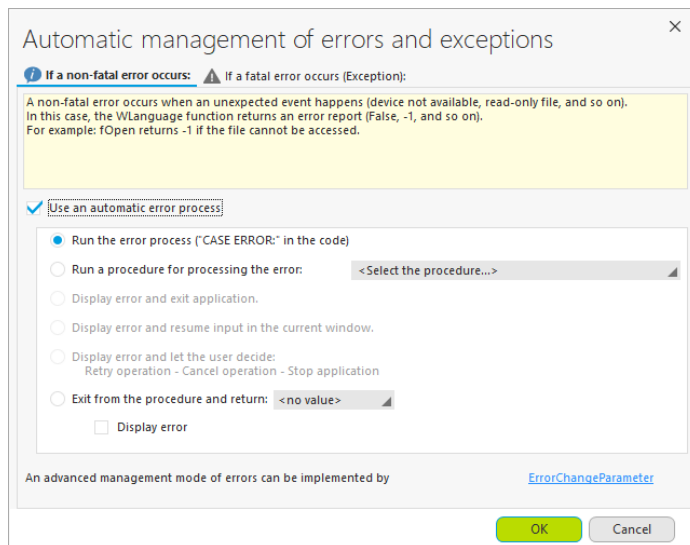
- an error value is returned by the function (for example, `fOpen` returns "-1" if the specified file was not opened).
- the error is detected by WLanguage (the `ErrorOccurred` variable is set to `True`) and the error details are returned by `ErrorInfo`.

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 link "If error: By program" in the code editor:



- by programming with `ErrorChangeParameter`.

Types of affected errors

Two types of errors can occur in WLanguage:

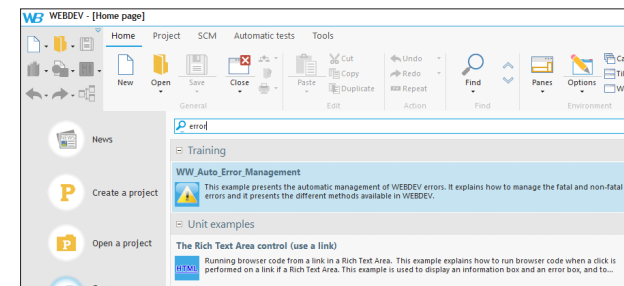
- the "non-fatal" errors (also called runtime errors): in most cases, these errors are managed in the code and they do not stop the application. For example, opening a file that cannot be accessed or an archive that does not exist.
- the "fatal" errors (also called programming errors): in most cases, these errors are linked to a development problem (access to a non-declared file, use of non-existing controls ...). A "fatal" error can also occur after a "non-fatal" error that was not processed properly. In this case, the application will be stopped.

The mechanism for managing 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

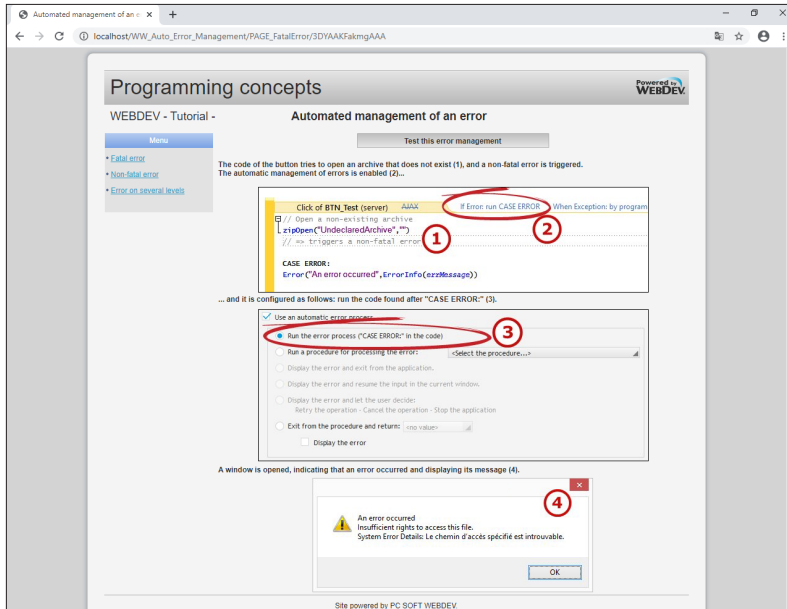
- To understand the different error cases as well as their management, we will be using a training example supplied with WEBDEV.

1. Display the WEBDEV home page (Ctrl + <).
2. On the home page, click "Open an example". The list of complete examples and training examples supplied with WEBDEV is displayed.
3. Type "Error" in the search area. Only the examples containing this word are listed.



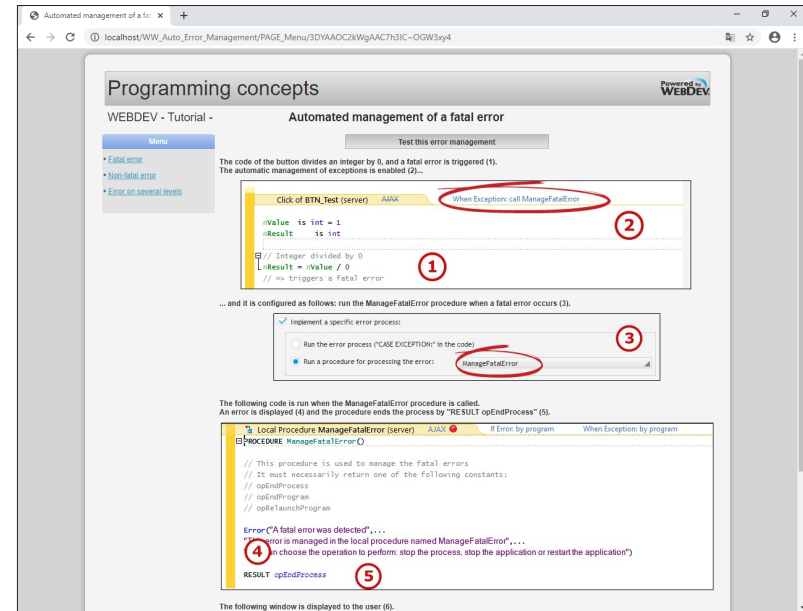
4. Select the "WW_Auto_Error_Management" project. The project is loaded.

- ▶ This project presents:
 - the management of a non-fatal error (opening a non-existent archive).
 - the management of a fatal error (dividing an integer by 0).
 - the management of an error on various levels.
- ▶ Run the project test by clicking (among the quick access buttons).
 1. Click "Non-fatal error".
 2. The following page is displayed.



3. Click the "Test this error management" button.
4. When running the code line that triggers the error, an error message is displayed.
5. Validate the error message.

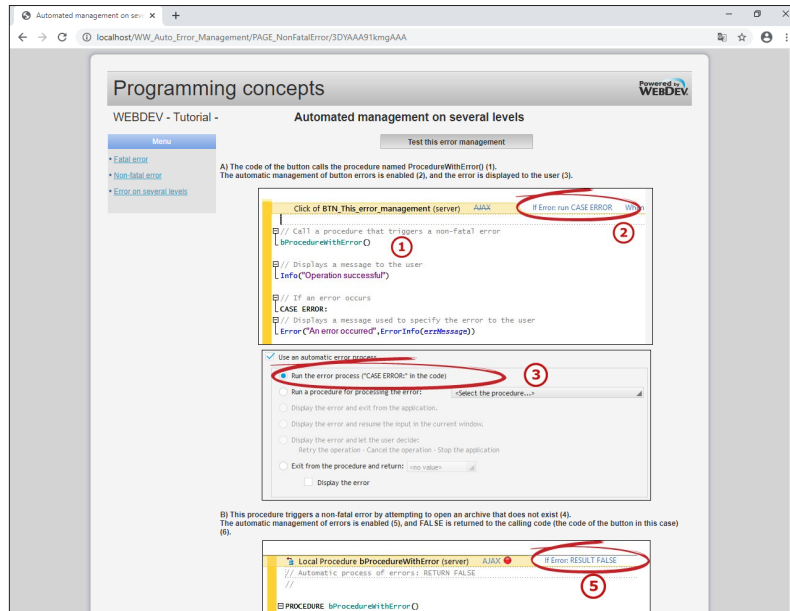
- ▶ Click "Fatal error".
 1. The following page is displayed. This page is used to test the error when an integer is divided by 0.



2. Click the "Test this error management" button.
3. When running the code line that triggers the error, a procedure is automatically called. This procedure is used to display the error message and to stop the current process.
4. Click "OK".

► Click "Error on several levels".

1. The following page is displayed. This page is used to test an error on several levels (process calling a procedure that opens an archive that does not exist).



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

3. When running the code line that triggers the error:

- the procedure returns "False" to the calling process.
- the calling process displays an error message and stops the process.

4. Validate the error message.

► Close the browser to end the test.



Example

WEBDEV also proposes an automatic management of HFSQL errors. See the online help for more details (keyword: HFSQL, Managing errors).

LESSON 9.3. IMPORT/EXPORT

This lesson will teach you the following concepts...

- Importing elements from a project to another one.
- Exporting the elements of your project.
- Specific import operations (WINDEV projects, HTML pages).



Estimated time: 10 mn

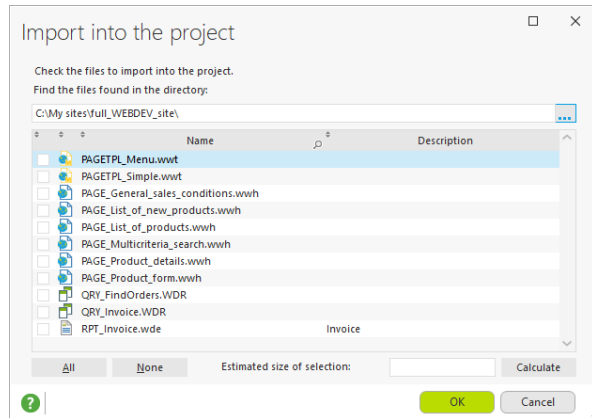
Importing elements

You have the ability to import existing WEBDEV elements into your project. All types of WEBDEV elements can be imported:

- pages, reports,
- classes, components,
- procedures, ...

► To import existing elements into the current project:

1. On the "Project" pane, in the "Project" group, expand "Import" and select "WEBDEV elements and their dependencies...".
2. Click "...", and select the directory containing the elements to import (the directory must include WEBDEV elements).
3. Validate. WEBDEV returns the list of elements that can be imported (subdirectories are ignored).



4. Select the elements to import and validate. The elements (and all files used by these elements : images, etc.) are now included in the project.



Remark

The "Calculate" button (found in the import window) is used to calculate the size of selected elements along with their dependencies.

Exporting elements

You also have the ability to export elements from your project to another directory for example. These elements can be re-used in other projects.

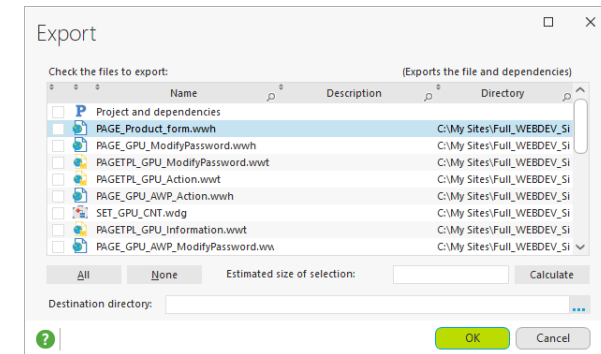


Caution!

Exporting elements is a lot more than a simple copy of elements. It is actually the only safe method for transferring a project element along with its dependencies (images, icons, ...). A practical solution for transmitting pages along with their dependencies by email for example.

► To export elements from your project:

1. On the "Home" pane, in the "General" group, expand "Save" and select "Export .. To a directory...".
2. In the window that is displayed, select the project elements that will be exported.



3. Specify the name of destination directory (or select it with the "..." button).
4. Validate. Then, the elements are exported to the specified directory. These elements are still available in your project.



Remark

The "Calculate" button (found in the import window) is used to calculate the size of selected elements along with their dependencies.

Specific import operations

Importing a WINDEV project

A widespread need today is to transform Windows applications (or parts of applications) into dynamic websites.

Thanks to the Webification wizard, WEBDEV allows you to:

- to transform your windows into pages,
- to be assisted while adapting the WLanguage code.

To test the webification of a WINDEV application, we have prepared a WINDEV project.



Remark

A WINDEV application can only be webified:

- using WINDEV 64 bits with WEBDEV 64 bits included.
Caution: You must have both WINDEV and WEBDEV licenses to perform this type of setup. To use the WINDEV-integrated version of WEBDEV, both WEBDEV and WINDEV dongles must be connected to the development computer.
- using WEBDEV 64 bits.

In this course, we will webify a WINDEV project directly from WEBDEV. **WEBDEV 64 bits is required to perform this operation.**

► To transform an existing WINDEV project into a website:

1. Open the WEBDEV home page (Ctrl + <).
2. In the home page, click "Tutorial" and select "Webify a WINDEV project (Exercise)".
3. The webification wizard starts and indicates that a WEBDEV project configuration will be created. Go to the next step of the wizard.
4. The wizard shows the different elements to transform. In our case, select all the elements.

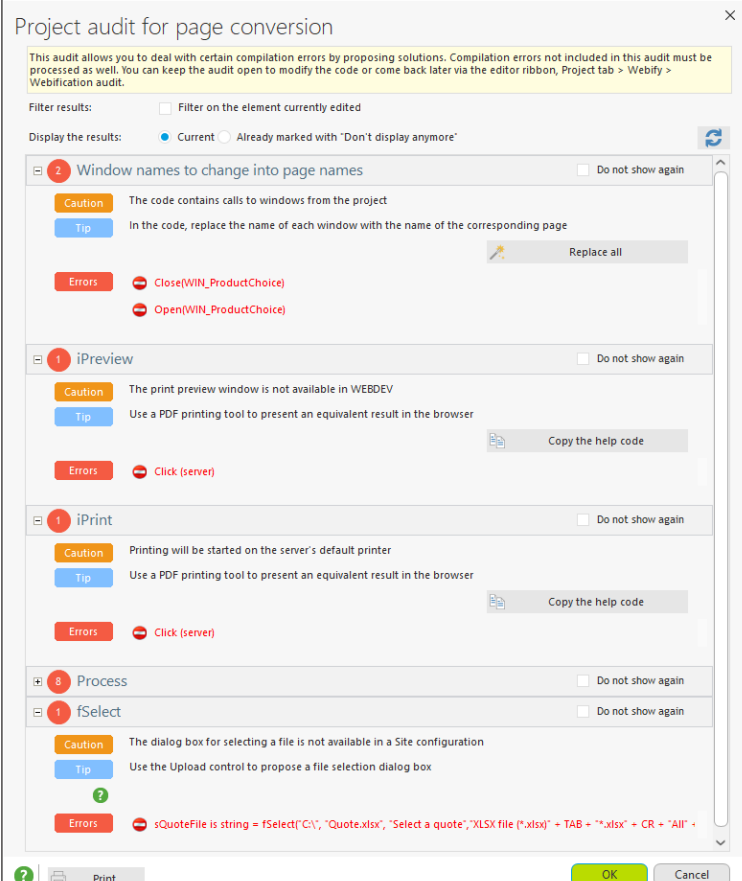


Remark

When webifying a WINDEV project, it is advisable to first transform one or two windows of your application. You can transform the other windows as you go along.

5. Go to the next step.
6. The wizard shows the elements that can be integrated into the site. In our example, we keep the report. Keep the default options and go to the next step.
7. The wizard is ended. Click "Finish".

8. The project audit for page conversion starts automatically. This audit makes it possible to deal with certain compilation errors by proposing solutions. It is necessary to correct all these errors for the application to be fully webified. Some errors may be related to processes or features that are not available on Web, unavailable WLanguage functions, etc, ... In our example, the audit shows the following problems:



Project audit for page conversion

This audit allows you to deal with certain compilation errors by proposing solutions. Compilation errors not included in this audit must be processed as well. You can keep the audit open to modify the code or come back later via the editor ribbon, Project tab > Webify > Webification audit.

Filter results: Filter on the element currently edited

Display the results: Current Already marked with "Don't display anymore"

2 Window names to change into page names Do not show again

Caution The code contains calls to windows from the project

Tip In the code, replace the name of each window with the name of the corresponding page

Replace all

Errors

- Close(WIN_ProductChoice)
- Open(WIN_ProductChoice)

1 iPreview Do not show again

Caution The print preview window is not available in WEBDEV

Tip Use a PDF printing tool to present an equivalent result in the browser

Copy the help code

Errors

- Click (server)

1 iPrint Do not show again

Caution Printing will be started on the server's default printer

Tip Use a PDF printing tool to present an equivalent result in the browser

Copy the help code

Errors

- Click (server)

6 Process Do not show again

1 fSelect Do not show again

Caution The dialog box for selecting a file is not available in a Site configuration

Tip Use the Upload control to propose a file selection dialog box

Errors

- sQuoteFile is string = fSelect("C:\", "Quote.xlsx", "Select a quote");XLSX file (*.xlsx) + TAB + ".xlsx" + CR + "All"

Print **OK** **Cancel**

► We will examine the different problems one by one:

1. The names of the windows: Of course, windows have been transformed into pages. The name of the elements has been changed in the project. This must also be done in the code. Click "Replace all" to automatically perform this operation.
2. Our WINDEV application printed a report in "Report viewer" mode. This option is not available on a WEBDEV site. The code must be modified to create a PDF file.
 - Double-click "Click (server)" next to "Errors". The click code with the error is displayed.
 - In the event banner, click on "AJAX enabled" to disable Ajax mode.
 - Replace the code with:

```
iDestination (iGenericPDF)
iPrintReport (RPT_OrderForm)
FileDisplay (iLastFile, typeMimePDF, iLastFile)
fDelete (iLastFile)
```

- Refresh the display in the audit window: all print errors have disappeared.

3. The last problem detected is related to the use of **fSelect**. Indeed, it is not possible to use a file picker in a website. It is preferable to use an Upload control. We have already seen this type of control in Part 3 (paragraph "Managing the image of a product").

This small example allowed us to see the simplicity and assistance provided by WEBDEV to transform a WINDEV project into a site.

Remark: In some cases, new developments may be necessary to adapt the application's behavior to the Web. For example:

- If the application was using an Image Editor control, it may be necessary to use an Upload control, which will allow downloading the modified image.
- If the application was using a Word Processing control, it may be necessary to use an HTML control with adaptations of the analysis to manage the HTML format.

For more details on the webification of a WINDEV project, see the online help.

Importing an HTML page

A page found in a non-WEBDEV site is useful? You want to retrieve its interface? Nothing's easier. WEBDEV gives you the ability to import your existing HTML pages into your WEBDEV project. Each imported page becomes an element of your WEBDEV site. This page can be modified!

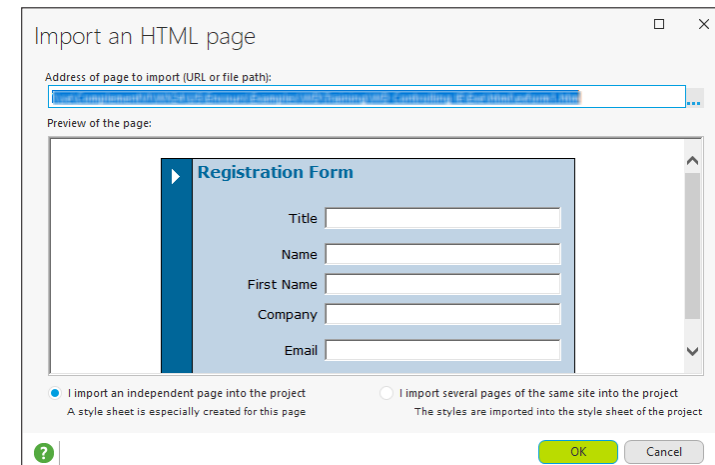


Remark

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 (for example: an existing project or a new project).
2. On the "Project" pane, in the "Project" group, expand "Import" and select "An HTML page...". The wizard for importing HTML pages starts.



3. Specify the location of HTML page. This page can be a file found on your computer ("..." button) or an Internet address (for example: <http://www.mywebdevsite.com/mapping.html>). The HTML page is displayed in the "Page preview" frame.
4. Validate. A new WEBDEV page was just created.
5. Save the page. The page is automatically added to the list of project elements.

PART 10

Optimizing and
debugging a
project



LESSON 10.1. OVERVIEW

This lesson will teach you the following concepts...

- Why optimize a site?
- Example project.



Estimated time: 5 mn

Overview

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

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

This part presents these tools and their use.

A project to optimize was prepared, allowing you to handle these features!

Opening project

► Open the "WW_Optimization" project.

1. Start WEBDEV 25 (if not already done).
2. Display the WEBDEV home page if necessary (Ctrl + <).
3. Click "Tutorial" and select "Optimize a project".

LESSON 10.2. PROJECT AUDITS

This lesson will teach you the following concepts...

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



Estimated time: 20 mn

What is an audit?

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

Two types of audits are available:

- **The static audit.** The static audit 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.

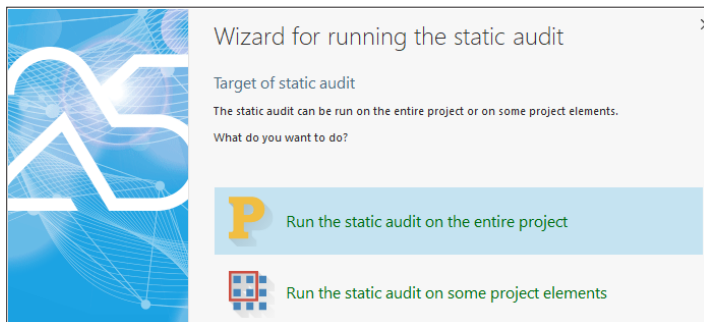
We are going to test these audits on the "WW_Optimization" project.

Static audit

The static audit is an environment feature used to study the source codes 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 performance" group, expand "Edition audit" and select "Trigger the edition audit".
2. The wizard starts. We are going to define the target of static audit.



3. Select "Run the static audit on the entire project".
4. Validate the wizard.

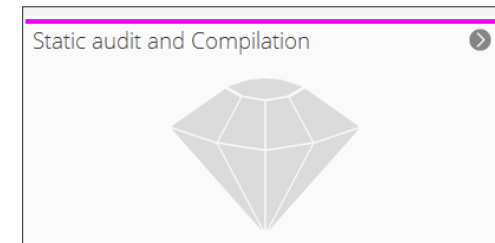


Remark

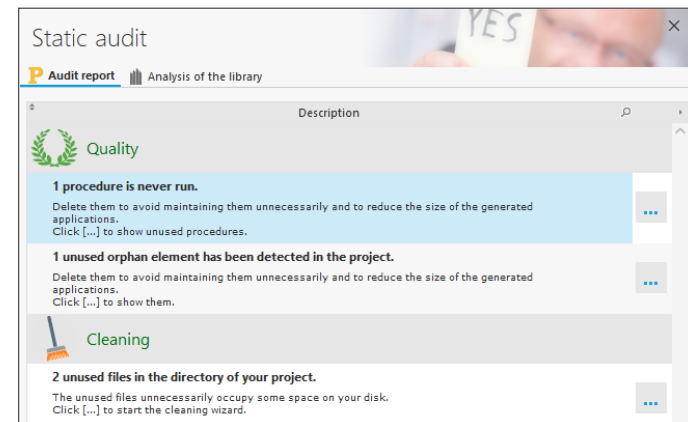
The static audit of project can also be started from the project dashboard, via the "Static audit and Compilation" widget.

Simply:

- display the Widget if necessary:
 - Switch the dashboard to modification mode (right mouse click, "Modification mode" option).
 - Add the Widget (right mouse click, "Add .. Static audit" option).
 - Close the modification mode (right mouse click, "Modification mode" option).
- click the arrow.



5. The audit report is displayed:



The static audit includes:

- The static audit of project.
- The audit for the content of application library.

► Let's study the topics presented by this report.

Procedure not run

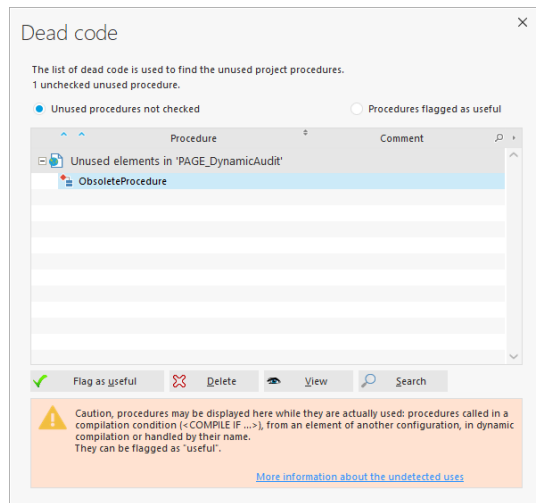
In our project, the audit indicates that a procedure is never run.

In huge projects, you may want to create one or more procedures to perform a process then, further to a code reorganization, the procedure is no longer used but it remains in the project.

The presence of unused procedures unnecessarily consumes the resources distributed to the end users.

► To fix this problem:

1. Click the [...] button to get more details. The window that lists the dead codes is displayed.



2. The "ObsoleteProcedure" procedure is never called. This window is used to:
 - specify that the procedure is still used ("Flag as useful"). In this case, the procedure will not be taken into account anymore during the audit.
 - delete the procedure if it is actually unused ("Delete").
 - see the procedure ("View").
 - find the use cases in the strings for example ("Search").
 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.

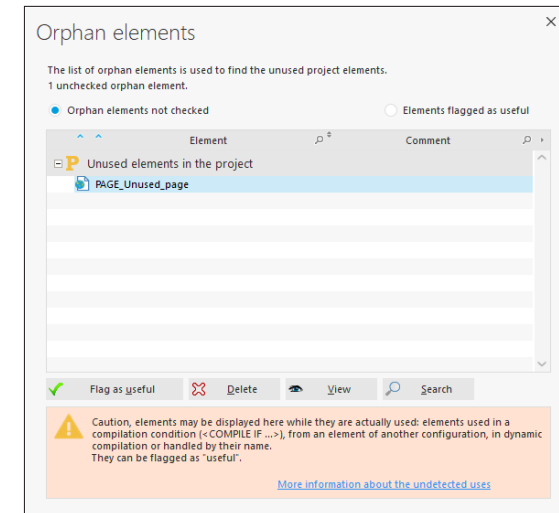
Orphan element

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 ("Flag as useful"). 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").
 - see the page ("View").
 - find the use cases in the strings for example ("Search").
 3. In our case, this "PAGE_Unused_Page" page is actually unused, click "Delete".
 4. Close the window of orphan elements (click the cross in the top right corner).
- In the window of static audit, click the "Refresh" button to update the audit report.

Cleaning the project

Our project contains several unused files. You have the ability to clean the project in order to keep the necessary elements only. The client setup is not weighted with images, external files, ... not used.

► To fix this problem:

1. Click the [...] button to get more details.
2. The project cleanup wizard starts. This wizard indicates the unused files that can be deleted.
3. Go to the next step.
4. Select the type of cleanup 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. Finish the wizard.

► Our project was optimized according to the tips given by the static audit. The static audit is used to get an overall status on the source code of your project. Our advice: run it on a regular basis!

Let's see what happens at run time by starting the dynamic audit.

Dynamic audit

The dynamic audit is used to study the application execution. The audit is used to detect problems such as:

- Excessive memory consumption,
- Slowness of algorithms used,
- Errors "hidden" at run time,
- ...

A dynamic audit can be performed in a test environment or on a live application.

The "WW_Optimization" project contains a specific page triggering errors that can be detected by the dynamic audit.

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

► To start the dynamic audit on the "WW_Optimization" project:

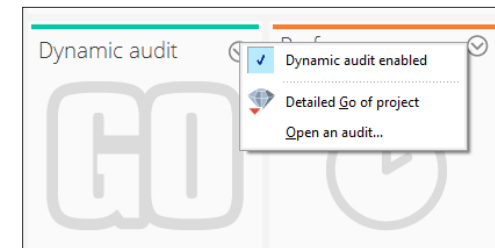
1. In the ribbon, on the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Debug the project while the audit is enabled". The project test is run.



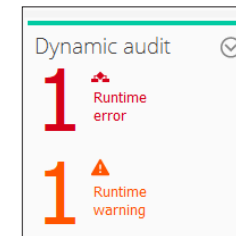
Remark

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

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



Remark: The dynamic audit is automatically run at each test of the project. A status report is automatically displayed in the dashboard:



2. Click "TEST PAGE OF DYNAMIC AUDIT".

3. Click the different buttons found in the page.

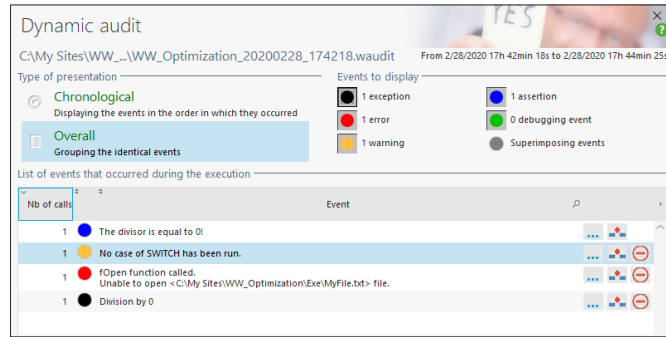
- At the end of each process, a toast message is displayed to specify that the process is over.

Remark: Wait for the toast to be displayed before clicking another button.

- For the "Assertion and Exception" option, an assertion is displayed: click "Continue" in order for the message to appear in toast format.

4. Stop the site test.

5. The report window of dynamic audit is displayed.




► Let's study this window:

- The top section of this window is used to choose the display mode of data. You can:
 - choose a chronological display (respecting the order in which the events occurred) or an overall display, used to group the different types of problems. During a chronological display, a chronological border allows you to see the position and importance of problems.
 - choose the type of problem to display (error, assertion, ...). This allows you for example to concentrate on the major errors, ...
- The bottom section of this window 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 a nonexistent image file to an Image control.

For each problem, a "..." button is used to access the event details. If the event is linked to a specific line of code, the  button allows you to open the code editor directly at the corresponding location in order to fix the problem.

► Close the window of dynamic audit.



Remark

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 study the result. See the online help for more details (keyword: "Dynamic audit").

LESSON 10.3. PERFORMANCE PROFILER

This lesson will teach you the following concepts...

- Overview.
- Starting the performance profiler.
- Studying the result.



Estimated time: 15 mn

Overview

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

The principle is straightforward:

- You run the test of your site.
- During this test, the performance profiler keeps track of all actions performed and saves the execution times of each one of the processes run.

At the end of test, the performance profiler displays:

- the 10 most time consuming operations,
- the duration and the number of calls for all processes run.

The "WW_Optimization" project contains a specific page used to view the interesting results with the performance profiler.

Starting the performance profiler

The performance profiler can be started:

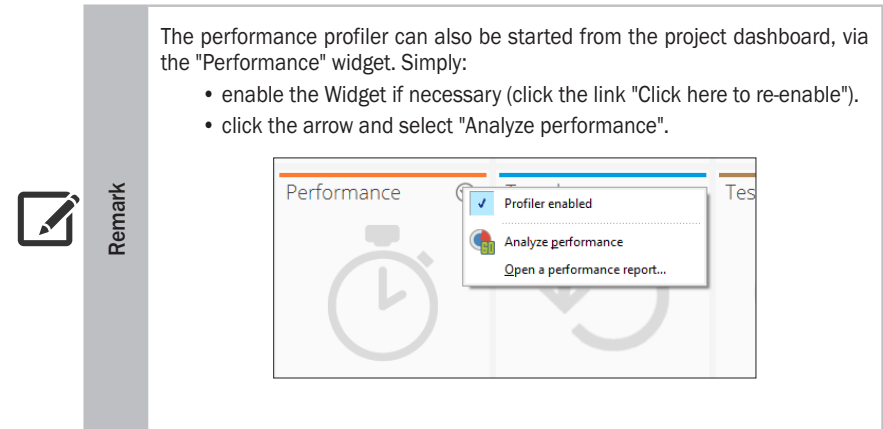
- **from the WEBDEV editor:**
In this case, the project is automatically run in test mode. You can use your application and start the processes of your choice.
To go back to the WEBDEV editor, all you have to do is exit from your site.
Then, the performance profiler displays the monitoring result. This result is saved in a WPF file.
- **from one of your WLanguage processes**, with the following functions:

ProfilerStart	Starts "collecting data" for the performance profiler.
ProfilerEnd	Stops "collecting data" for the performance profiler.

In this case, only the code found between **ProfilerStart** and **ProfilerEnd** is studied. 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:

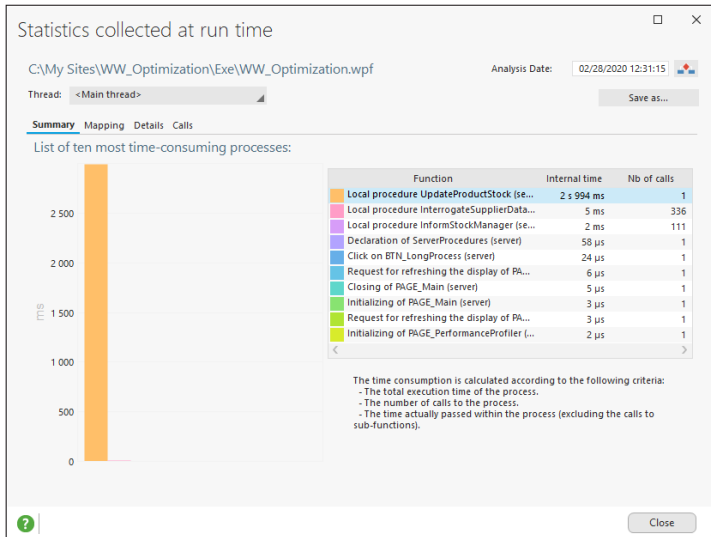
1. In the ribbon, on the "Project" page, "Audit and performance" group, expand "Analyze performance" and select "Analyze performance".



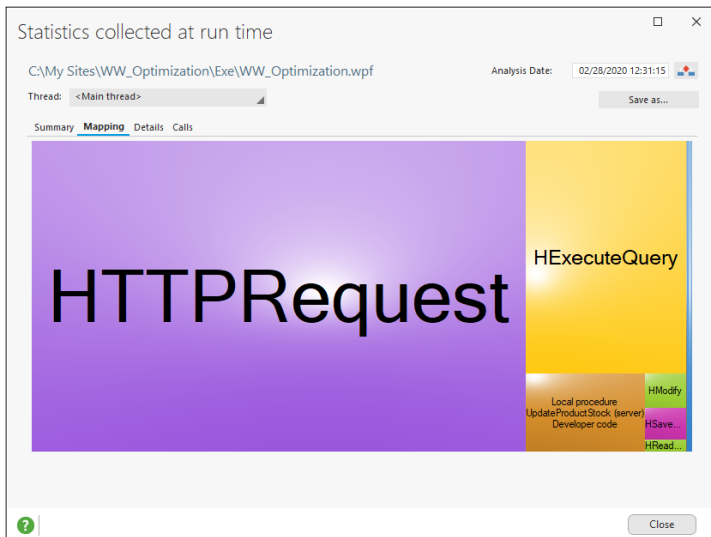
2. The project test is run.
3. Click "TEST PAGE OF PERFORMANCE PROFILER".
4. Click "PROCESS TO ANALYZE".
5. Validate the information window and stop the project test. The performance profiler report window appears.

Studying the result

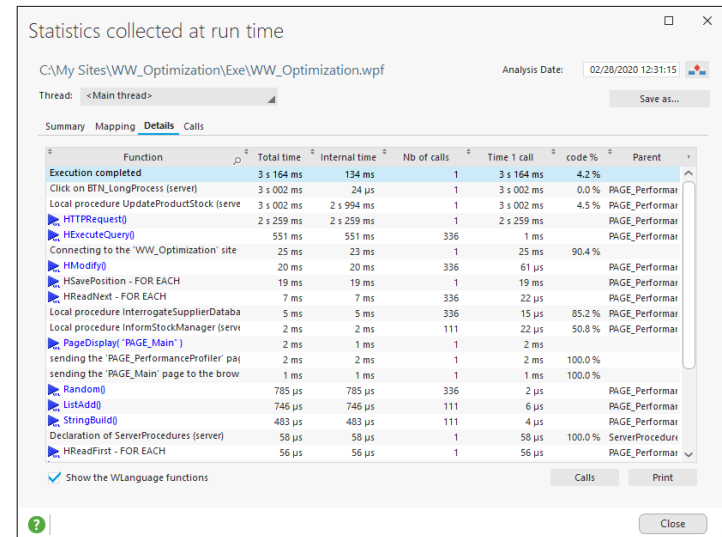
- Let's study the report window of the performance profiler. Results are displayed in several tabs:
 - the "Summary" tab presents the ten longest processes.
 - the "Mapping" tab presents a graphical view of main processes.
 - the "Details" tab presents all processes run during the application test (from the slowest one to the fastest one).
 - the "Calls" tab is used to view the details of 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's a call to **HTTPRequest**:



- The "Details" tab presents all processes or events run, from the slowest one to the fastest one.



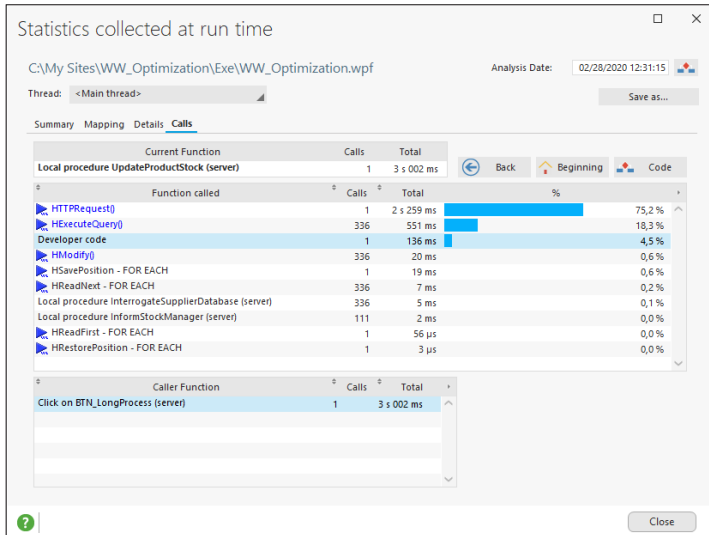
The following information is displayed for each process:

- Function: Function, event or procedure run.
- Total Time: Execution time of function.
- Internal time: Execution time due to the engine.
- Nb of calls: Number of calls made to the function (procedure or event).
- Time 1 call: Execution time of a call to the function (procedure or event).
- Code %: Percentage of time spent in the process of function or procedure (developer code that can be optimized).
- Parent: Element that contains the process or event.

- In our case, the "Details" tab indicates that the call to **HTTPRequest** is one of the elements taking the longest time.

1. Select this line. We are going to check whether this slowdown is caused by a specific problem.

2. Click the "Calls" button to display the details of calls to the **UpdateProductStock** procedure. Select "HTTPRequest" and click "Code": the corresponding line of WLanguage code is displayed in the code editor.



3. Close the performance profiler.
4. The following line of WLanguage code is run:

```
HTTPRequest ("supplier-addr")
```

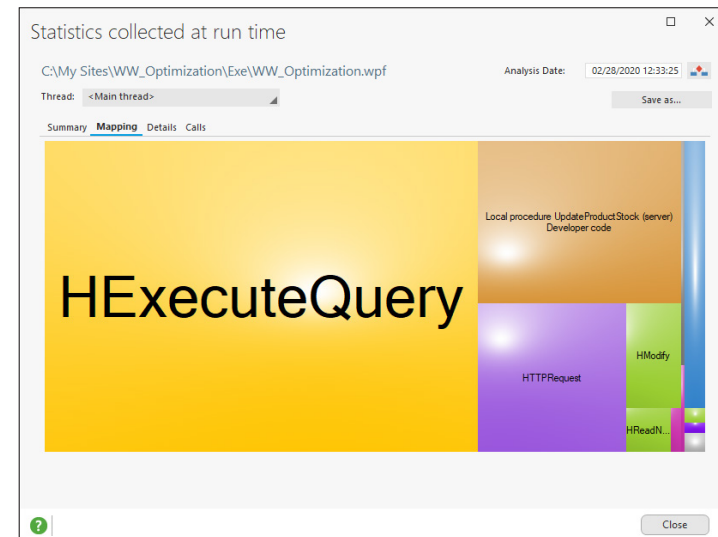
The slowdown is caused by the fact that the address specified for **HTTPRequest** is not accessible.

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

```
// Checks the accessibility of supplier server
HTTPRequest ("www.google.com")
```

2. Save the code (Ctrl + S).

- We are now going to restart the performance profiler:
 1. On the "Project" pane, "Audit and performance" group, expand "Analyze performance" and select "Analyze performance".
 2. The project test is run.
 3. Click "TEST PAGE OF PERFORMANCE PROFILER".
 4. Click "PROCESS TO ANALYZE".
 5. Validate the information window and stop the project test. The performance profiler report window appears.
 6. In the "Mapping" tab, **HTTPRequest** does not appear with the same importance.




- Close the performance profiler report window.

LESSON 10.4. DEBUGGING A PROJECT

This lesson will teach you the following concepts...

- Overview.
- Using the debugger.

 Estimated time: 15 mn

Overview

Let's take a look at the debugger that is supplied with WEBDEV.


What is the debugger?

The debugger is a powerful tool used to follow the code or application progress, 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:

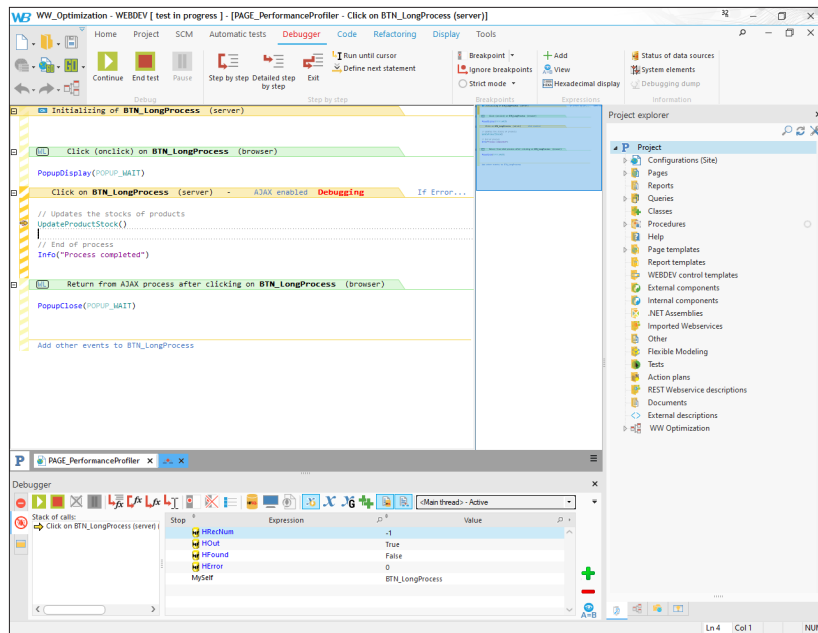
1. Open "PAGE_PerformanceProfiler" in the page editor (double-click its name in the "Project explorer" pane).
2. In the WEBDEV editor, set a breakpoint in the code of the Button control "PROCESS TO ANALYZE": click next to the first line of code in the "Click (server)" event or press Ctrl + B. A red dot will appear. The debugger will be automatically started when the line preceded by the breakpoint is run.
3. Run the test of "PAGE_PerformanceProfiler" page ( among the quick access buttons).



Remark

See the online help for more details (keyword: "Debugger, Run a test").

4. Click "PROCESS TO ANALYZE". The debugger starts: the code editor is displayed in "Debugger" mode. The current line is preceded by a little arrow.



The "Debugger" pane appears in the lower section of the screen. This pane displays two distinct areas:

- the call stack: this area is used to find out the hierarchy of processes displayed in the debugger. In our example, we are debugging the "Click" event of BTN_LongProcess.
- the list of expressions to evaluate. By default, the main variables used in the code are displayed in this section. You can 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.

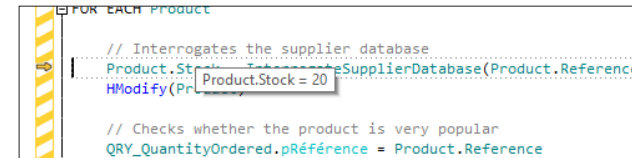
► First of all, we are going to run the different statements step by step and see the content of variables:

1. Press F7 (or click "Step by step"). The current line is automatically run by going into the sub-procedures if necessary.

Remark: To avoid going into the sub-procedures, press the F8 key or click the "Step by Step" button.

The values of variables are modified (if necessary) in the "Debugger" pane (displayed at the bottom of the screen).

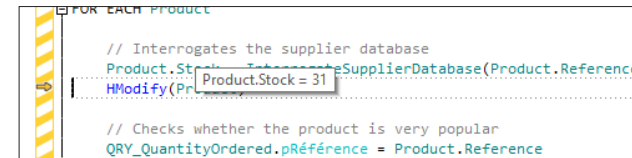
2. Keep pressing F7 until you reach "Product.Stock = InterrogateSupplierDatabase ...", hover "Product.Stock" with the cursor. A tooltip is displayed with the expression value:



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 F8 to run the line.

5. Hover "Product.Stock" again. The value of "Product.Stock" displayed in the tooltip corresponds to the result of InterrogateSupplierDatabase procedure.

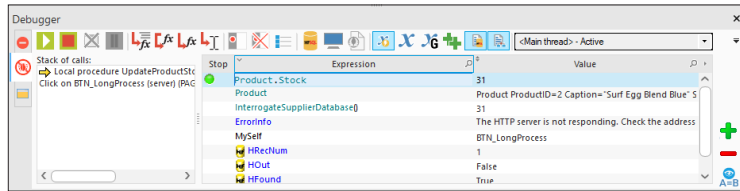


► We are now going to add an expression to monitor the evolution of its value in the "Debugger" pane. This expression can have any type: variable, function, operation on variables, ... The result of the expression is calculated and displayed.

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 expression to 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:

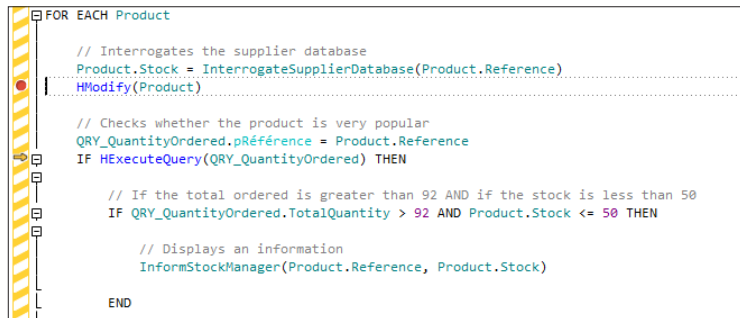
1. Position the cursor on the following line:

```
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:

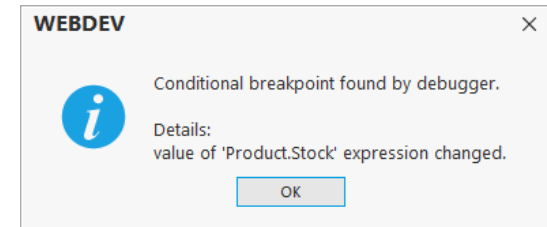
1. Click in the hatched area with the mouse, in front of **HModify**. A breakpoint (red bullet) appears.



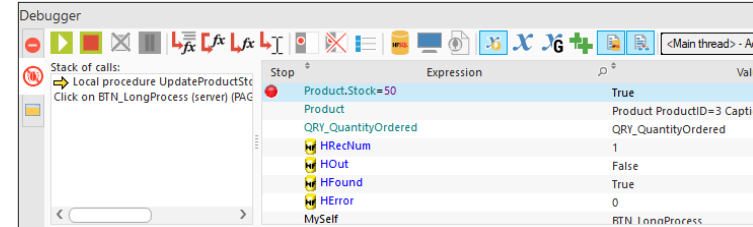
2. Press F5 (or click "Continue" in the ribbon). The code is run until it reaches the breakpoint. The arrow used to identify the current line moves until it reaches the breakpoint.
3. Click the breakpoint to remove it.

► To end this test, we will be using an "Auto-stop" expression. An "Auto-stop" expression is used to start the debugger as soon as a condition is checked or when the value of a variable is modified. In our example, the debugger will be started as soon as the stock value is equal to 50:

1. In the "Debugger" pane, select the "Product.Stock" expression that was added beforehand.
2. Click the green circle.
3. Press F5 to continue the test.
4. A message is displayed, indicating that the value of "Product.Stock" expression changed.



5. Validate.
6. In the "Debugger" pane, select the "Product.Stock" expression. Click the expression again: the "Expression" column becomes editable. In the "Expression" area, add "=50". You will get "Product.Stock = 50".



7. Press F5. The program continues to run. The debugger is started again when the value of Product.Stock variable is equal to 50.
- That's it, the main features of debugger have been presented here. To stop the test in the debugger, click "End the test" found in the ribbon.

PART 11

Managing a
HFSQL Client/
Server database



LESSON 11.1. INTRODUCTION

This lesson will teach you the following concepts...

- Principle of Client/Server.
- Why switch an application to HFSQL Client/Server?



Estimated time: 5 mn

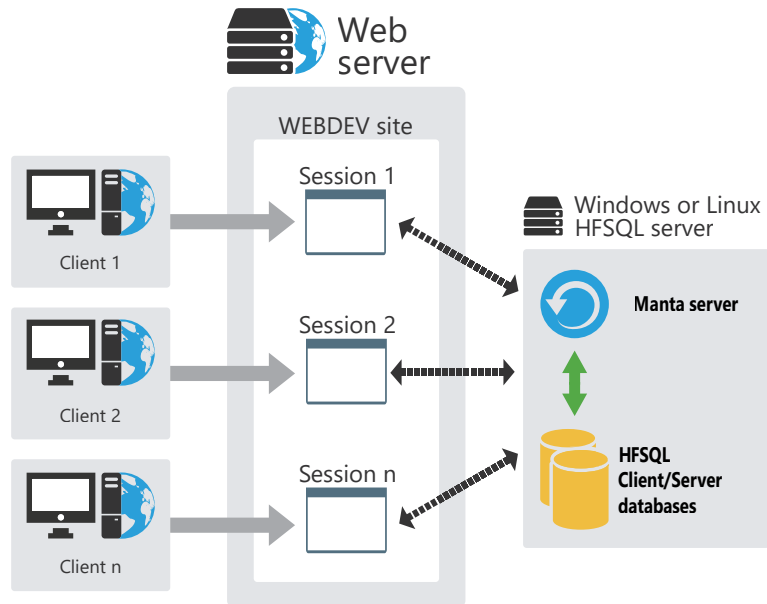
Overview

WEBDEV allows you to create applications/sites that access HFSQL Client/Server databases.

A database in Client/Server mode is used to host the databases on a server (this server can differ from the one where the site is deployed).

This operating mode:

- increases 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 secure (use of a login and password and definition of rights granted to the users).
- No management of directories: all the database files are grouped at the same location.
- The databases in Client/Server mode can be used by an Internet connection.
- Management of native multi-user mode: the performances are optimized in multi-user mode.

LESSON 11.2. IMPLEMENTING A CLIENT/ SERVER DATABASE

This lesson will teach you the following concepts...

- Installing a local HFSQL server.
- Creating a site that uses an HFSQL Client/Server database.
- Adapting a site to manage an HFSQL Client/Server database.
- Features available in Client/Server mode.

 Estimated time: 10 mn

Overview

In this lesson, we are going to perform all the operations required to develop and to deploy a site that is using an HFSQL Client/Server database.

Installing a local HFSQL server

The first operation to perform consists in installing an 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 HFSQL server setup program is available on the WEBDEV setup media. If you do not have this media, the HFSQL server setup is also available on our site (www.windev.com).

To install the HFSQL server locally:

1. Start the setup program of WEBDEV.
2. Select "HFSQL Server Setup".
3. Then, select "Install or update an HFSQL server".
4. Accept the license agreement.
5. Choose the platform ("For Windows on this computer").
6. If HFSQL servers are already installed on the current computer, select "Install a new server".
7. Select the setup directory and specify the server name and port. The port 4900 will be used by default.



Remark

Don't forget to open this port on the firewall in order to connect to the HFSQL server from another computer.

8. Specify the name and password of the user with management that will be created by default on the HFSQL server. By default, this user is "Admin" and has no password.



Remark

For security reasons, don't forget to change the administrator password.

9. If the HFSQL Control Center is not installed or is not accessible from your computer, check the option "Install the HFSQL Control Center".



Caution!

The HFSQL Control Center is required to manage the HFSQL Client/Server database.

10. The wizard proposes to configure the sending of notifications in order to identify the server dysfunctions in real time. In our example, go to the next step and indicate that the setting will be performed "Later".

11. The wizard proposes to enable the authentication via Active Directory. Enable this option if necessary.

12. Validate (or not) the participation in product improvement by allowing us to collect information regarding the use of product. This optional and anonymous collect allows PC SOFT to improve the product features.

13. The setup is performed. By default (if you have not changed the administrator settings), to connect to the server in administrator mode, you must use the "Admin" user without a password.

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

Creating a WEBDEV site that is using an HFSQL Client/Server database is child's play. You must:

1. Create the project by asking 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 data file in the analysis, indicate that this data file is in Client/Server mode and specify the connection used.



Remark

You can also describe the connection to the HFSQL server by programming. See the online help for more details: "[HDescribeConnection](#)".

Adapting a site to use a HFSQL Client/Server database

Overview

Switching a database from HFSQL Classic mode to 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

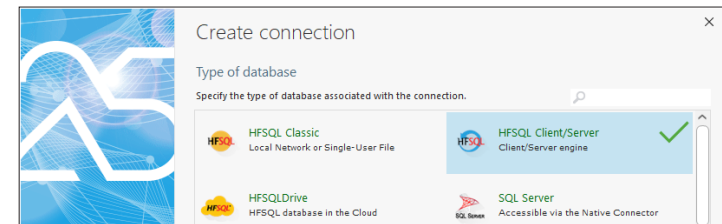


Answer

If you did not perform the operations in the previous lessons, you can follow this lesson by opening a corrected project: in the WEBDEV home page (Ctrl + <), click "Tutorial" and select "Full WEBDEV Site (Answer)".

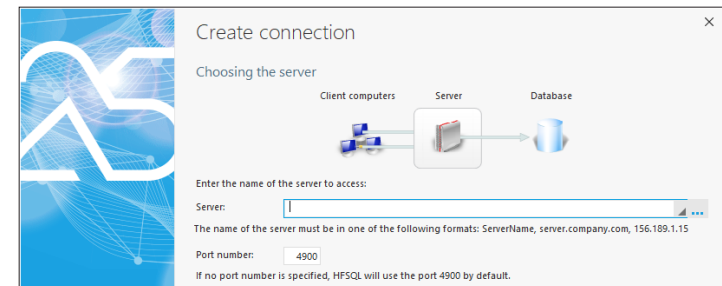
► To adapt the project:

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 WEBDEV menu. The data model editor is displayed.
3. In the ribbon, on the "Analysis" pane, in the "Connection" group, click "New connection". A wizard is opened, allowing you to create a connection.
4. Select the type of connection to create: "HFSQL Client/Server".



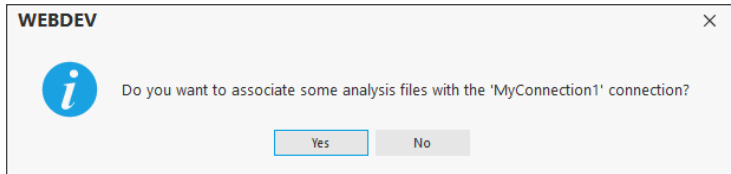
Go to the next step.

5. In the following steps, specify:



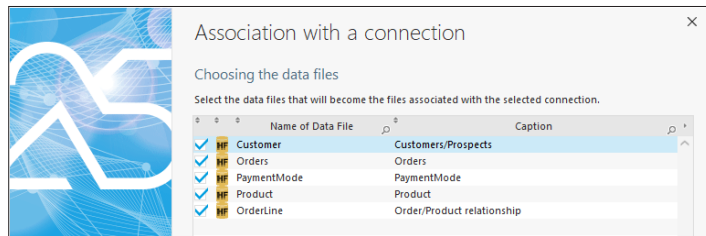
- the name of the server on which the HFSQL Client/Server server was installed (localhost if it was installed on your computer for example) and the port number. Go to the next step.
 - the user name and password (to use the administrator, specify the custom login and password when installing the HFSQL server). Go to the next step.
 - the database name ("Full_WEBDEV_Site" in our example). Go to the next step.
6. Type the connection name (keep the proposed name).

7. Go to the next step and validate. The connection to the database is automatically created. The wizard proposes to associate the different data files found in the analysis with the connection that was just created.



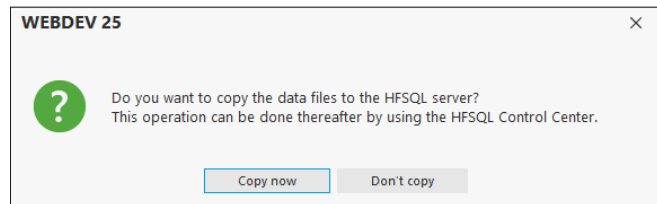
8. Click "Yes".

9. In the next step, select all the data files proposed:

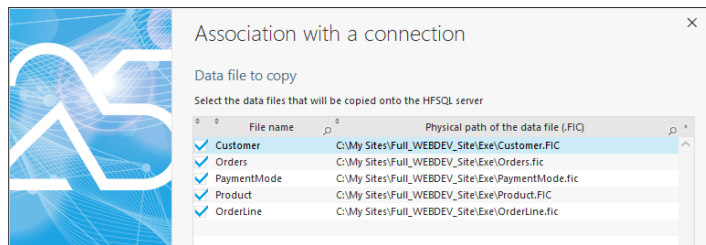


Go to the next step.

10. Then, the wizard proposes to copy the data files onto the server. Validate ("Copy now").

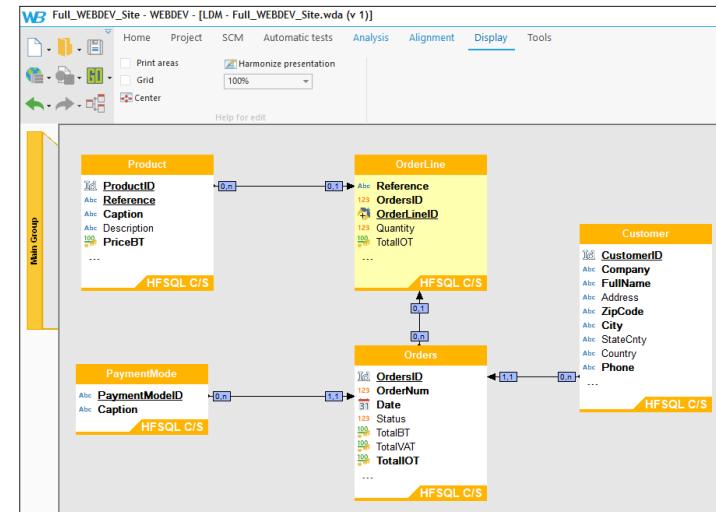


11. Select the analysis files that will be copied onto the server: in our case, all data files found in the EXE directory.



12. Go to the next step and validate.

13. The data files in the analysis are automatically changed into HFSQL Client/Server data files and associated with the selected connection.



14. Generate the analysis: In the ribbon, go to "Analysis", in the "Analysis" group, click "Generation". An automatic modification of data files is automatically proposed. If all the data files are up to date, the automatic modification of data files can be canceled.



Remark

Switching to Client/Server mode: tips

- Check the code of your project : in HFSQL Client/Server mode, the **HSubstDir** statements, ... are useless.
- Depending on the parameters specified when creating the connection, the connection defined in the analysis can be modified by **HOpenConnection** and **HChangeConnection**.
- **HOpenConnection** is used to go back to HFSQL Classic mode: all you have to do is specify the path of directory containing the HFSQL Classic data files.

15. The development project was successfully adapted. You may also have to adapt the deployed site (if the deployed site is using HFSQL Classic data files for example). This operation is configured when creating the setup program of the site.

Features available in HFSQL Client/Server mode

HFSQL Client/Server proposes several features:

- Transactions,
- Logs,
- Stored procedures,
- Triggers,
- Hot automatic data modification,
- Hot reindexing,
- Scheduled backups,
- Incremental backups,
- Universal replication.

These features will not be described here (some of them have been presented in this tutorial in HFSQL Classic mode).

See the online help for more details.

LESSON 11.3. MANAGING A CLIENT/SERVER DATABASE

This lesson will teach you the following concepts...

- The HFSQL Control Center.
- Creating a user in the HFSQL Control Center.
- Saving the database.



Estimated time: 20 mn

Overview

Now that we know how to create and/or adapt a site so that it operates in HFSQL Client/Server mode, let's see how to manage the associated database.

Indeed, a Client/Server database requires:

- a specific configuration of computers (setup of 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.

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.

However, for performance reasons, this configuration is not recommended. One or more databases can be installed on each server.

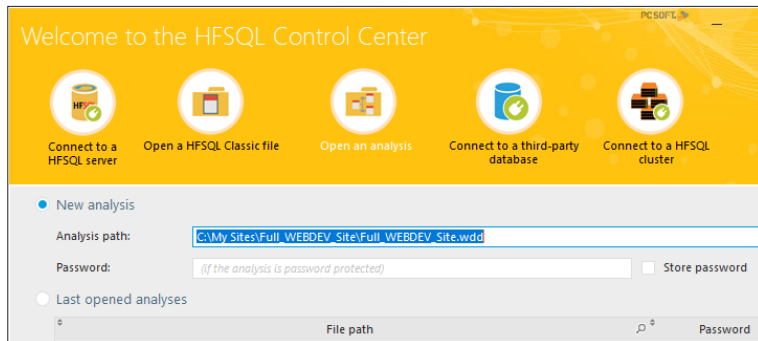
The HFSQL Control Center

The HFSQL Control Center is used to perform all management operations on the HFSQL Client/Server servers and databases.

We are going to present the most important features.

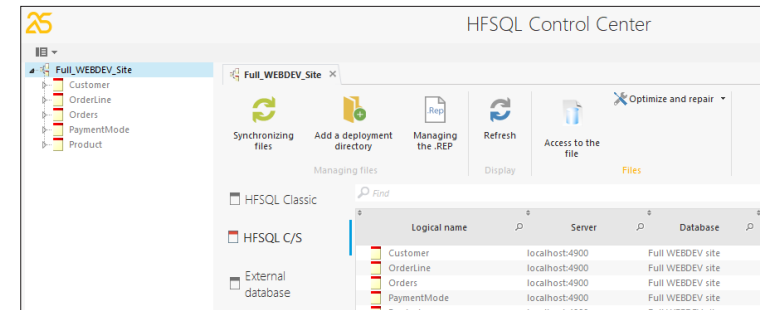
First, we are going to start the HFSQL Control Center from the WEBDEV project.

- ▶ 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 HFSQL Control Center is displayed. The analysis of current project is automatically selected.



3. Validate the screen. The HFSQL Control Center is displayed. This start mode allows you to see the different data files linked to the analysis of the current project.

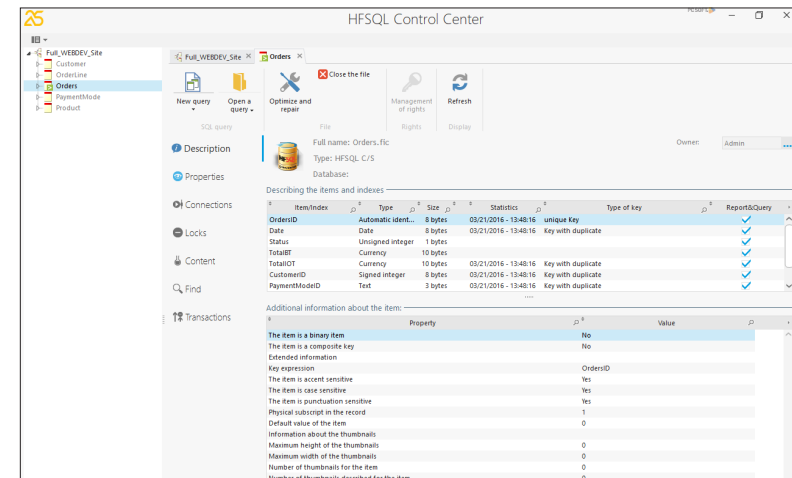
4. Click the vertical tab named "HFSQL C/S" if necessary. The list of data files in HFSQL Client/Server format is displayed.



The Control Center lists the Client/Server data files found in the analysis linked to the current project. No connection is established: the names of the data files are grayed out.

5. To view the records in the data files, double-click one of the data files in the list on the left (for example, "Orders"). If the HFSQL Control Center does not recognize all 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 uses this connection is displayed in a new "Orders" pane. This pane presents this information in several vertical tabs:



- The "Description" tab gives information about the data files (data file items, etc.).
- 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

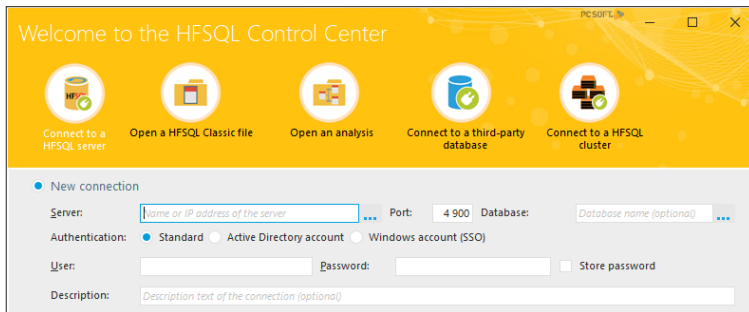
A single user account is created when installing a HFSQL server and when creating a database: the administrator account. By default, the administrator has the login "Admin" and no password. Reminder: it is possible to customize the password when installing the HFSQL server. Using a user account allows you to secure the access to data. Indeed, all the users of the site are not administrators. Specific rights can be granted to each user (or group of users).



Caution! The user rights specified in the HFSQL Control Center are granted for the database and not for accessing the site. Do not confuse right management for the Client/Server databases with the user groupware, which we've seen in a previous lesson.

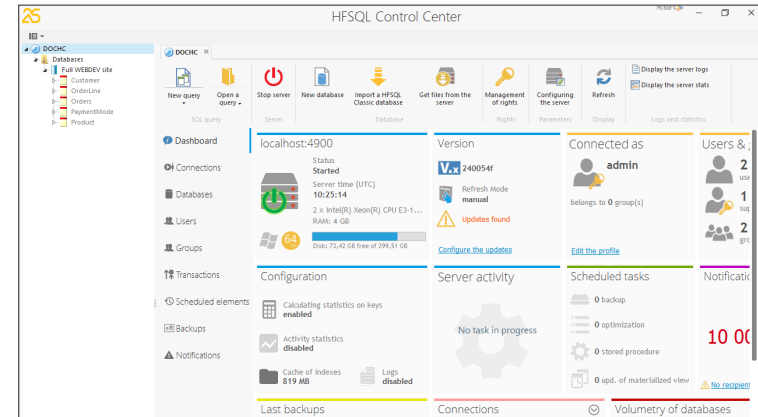
Some users may not have the rights to write into some data files for example. To run a simple test, we are going to create a user and allow this user to see the records in the Customer data file.

- ▶ To directly connect to the database found on the server:
 1. Expand the menu found at the top left of HFSQL Control Center and select "Connect to a HFSQL server".
 2. The home window of HFSQL Control Center is displayed.

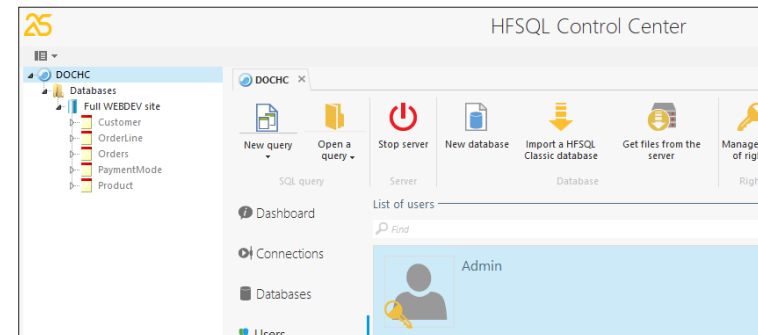


3. The option "Connect to a HFSQL server" is selected by default. Specify the characteristics of the server installed in the previous lesson.

4. The characteristics of HFSQL server are displayed in the different panels:
 - the name of HFSQL server as well as the list of databases found on this server are displayed in the left panel.
 - in the right section of the screen, a new tab allows you to see the characteristics of HFSQL server.



5. In the right section of the screen, select the "Users" tab. This tab is used to manage the server users.
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 user characteristics is displayed.

8. Type the following information:

(use "Test" as password for example).

Remark

Several characteristics can be noticed:

- **Super User:** The users defined as "Super user" are allowed to perform all types of actions on the server, on the databases and on all the data files.
- **Account enabled:** If this option is not checked, the user exists but he is not enabled (users on holiday for example).
- **Password expiration:** You can specify a password valid for a limited number of days. (configurable).

9. Validate the user creation. By default, no rights are granted to this user.

We are now going to grant rights to the user: the "Test" user can connect to the database and read the Customer data file.

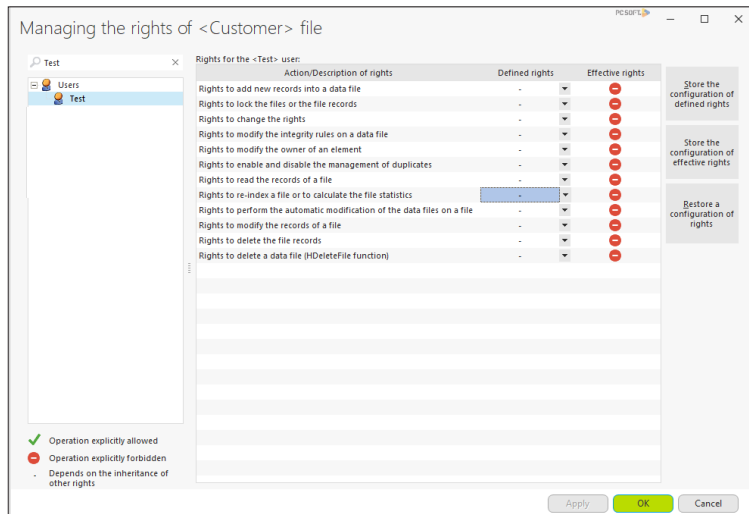
► To grant rights to connect to the database:

1. In the HFSQL Control Center, double-click the "Full_WEBDEV_Site" database. A new tab appears.
2. On the "Full_WEBDEV_Site" pane, in the "Rights" group, click "Management of rights".
3. Select the "Test" user in the list of users.
4. In the list of rights, for the action named "Rights to connect to the server (encrypted and unencrypted connection)", click in the "Defined rights" column and select the green checkmark.

5. Click "Apply" at the bottom of window. The rights become effective.
6. Close the window for managing rights.

► To grant rights to read the Customer data file:

1. In the HFSQL Control Center, expand the "Full_WEBDEV_Site" database, and then double-click the Customer data file (on the left of the screen).
2. On the "Customer" pane, in the "Rights" group, click "Management of rights".
3. Select the "Test" user in the list of users.



4. In the list of rights, for the action named "Rights to read the records of a file", click the "Defined rights" column and select the green checkmark.
5. Click "Apply" at the bottom of window. 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 can only browse the records in the Customer data 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).



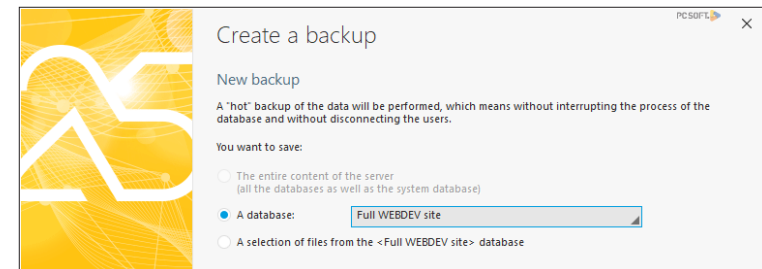
Remark

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 make a backup copy of the database:

1. Go to the tab corresponding to the "Full_WEBDEV_Site" database.
 2. Select the vertical tab named "Backups".
 3. In the menu, in the "Backups" group, expand "New backup" and select "New hot backup".
- Remark: This button is accessible in the "Backups" group:
- on the tab corresponding to the HFSQL server,
 - on the tab corresponding to the database.



4. Validate the backup.

Conclusion

The HFSQL Control Center is a tool for managing databases, allowing you to:

- stop or restart a server if a problem occurs,
- manage the users and their rights,
- reindex the data files if necessary,
- perform database backups.

The HFSQL Control Center is a redistributable tool that can be installed on the computers of users who are working with HFSQL Client/Server databases. The HFSQL Control Center must be used by the database administrator.

PART 12

Source Code
Manager



LESSON 12.1. SCM

This lesson will teach you the following concepts...

- Overview.
- The source code manager.



Estimated time: 15 mn

Introduction

The development of a large IS system requires the participation of several developers. These developers must work on a single WEBDEV project while sharing the different resources (pages, classes, ...).

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 modifications performed (in the code, in the interface, ...).

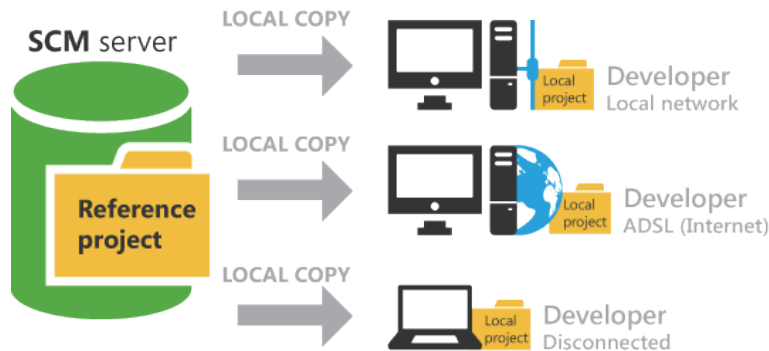
SCM (Source Code Manager)

Principle of SCM

The Source Code Manager is used to store and share the projects and their elements.

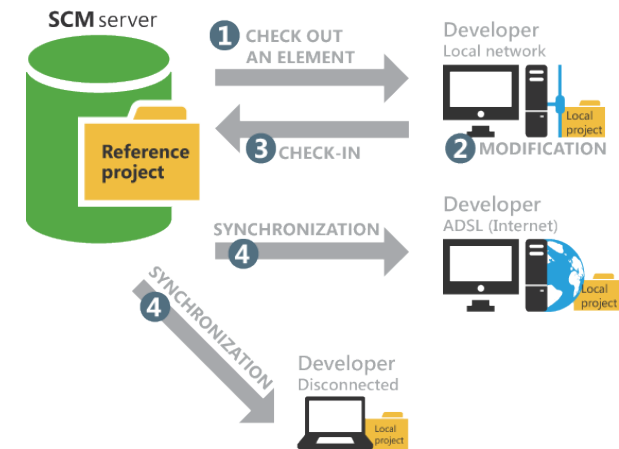
The principle is as follows:

- A reference version of each one of your projects is found on a server. This set of versions is called "Repository".
- Each developer has a local copy of 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, the developer will check out the element from the repository.
- This developer gets exclusive rights on this element: all 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 repository).
- Once the developer has finished, the checked-out element is checked back into the repository.

- 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 repository

To share a project via the Source Code Manager, a repository must be created. This repository must be created once only on a server.

This repository 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 repository 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, at any time. Visit www.pcloud.net for more details.
- SCM Drive. This mode allows you to have the Source Code Manager in the cloud for all your WEBDEV projects. Visit www.scmdrive.com for more details.

► In the next lesson, our repository will be created when importing a project into the SCM.



Remark

We advise you to make backup copies of the repository on a regular basis. To do so, you must:

- connect as administrator to the management tool of SCM.
- on the "Management" pane, in the "Backups" group, select "Full backup".

LESSON 12.2. INTEGRATING A PROJECT IN SCM

This lesson will teach you the following concepts...

- Adding a project into SCM.
- Opening a project from SCM.
- Configuring SCM.

 Estimated time: 15 mn

Integrating a project in SCM

Adding the project into SCM

To use an existing project with the Source Code Manager, simply include it in the repository.

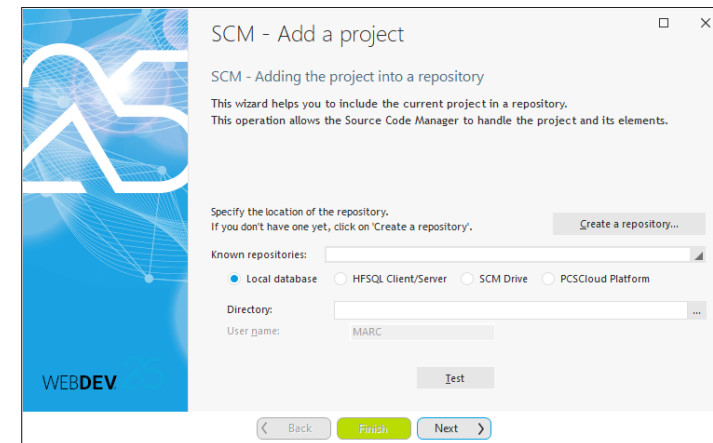


Answer

If you did not perform the operations in the previous lessons, you can follow this lesson by opening a corrected project: in the WEBDEV home page (Ctrl + <), click "Tutorial" and select "Full WEBDEV Site (Answer)".

- We are now going to include the "Full_WEBDEV_Site" project in the repository:

1. In the ribbon, go to "SCM", group "Add project", and click "Add project to SCM". The wizard for adding projects into a repository starts:



The repository has not been created yet. We are going to create one.

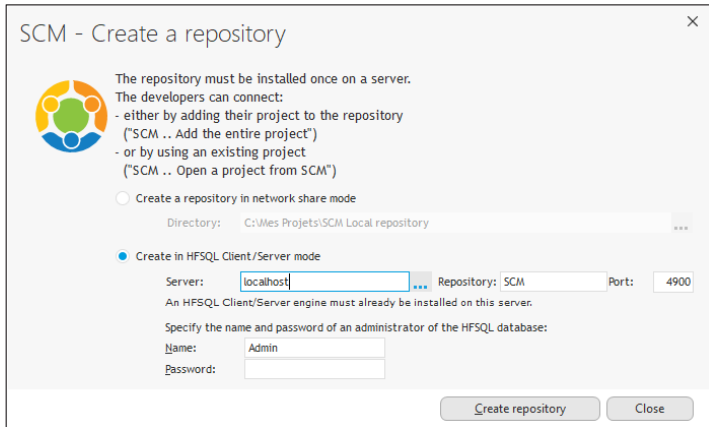


Remark

We are going to create a "Client/Server" repository on the server that was installed in part 11. Please read this part carefully if you haven't already!

2. Click on "Create a repository".

3. The repository creation screen is displayed.



The repository can be in HFSQL Classic format (local or network) or in HFSQL Client/Server format. We are going to create a repository in HFSQL Client/Server format.

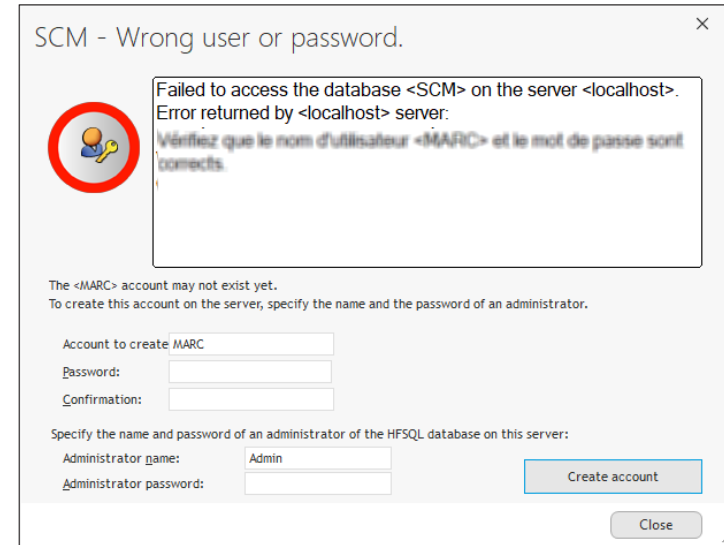


Remark

If the repository is in HFSQL Client/Server format, it can be used remotely.

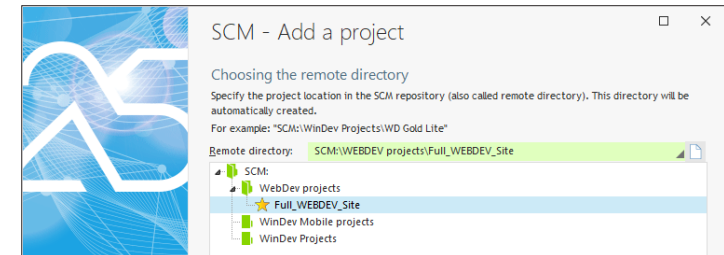
4. Select "Creation in HFSQL Client/Server mode".
5. Specify the parameters of the HFSQL server that was installed beforehand:
 - The name of the server, its port.
 - The name of the administrator, his possible password.
6. Validate the creation of the repository ("Create repository"). This operation can take quite a long time in Client/Server mode.
7. The repository has now been created. We are going to include our project in this repository.

8. Go to the next step. The current user is not saved in the database of users of HFSQL server.



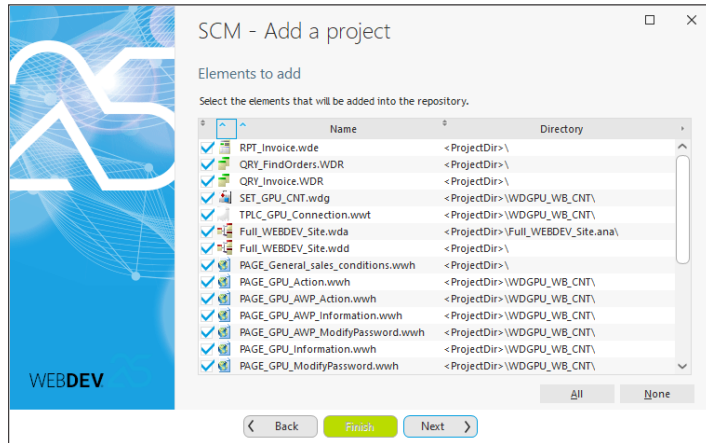
We are going to create this user: click "Create account".

9. The wizard proposes to locate the project in the "WEBDEV projects" subdirectory of the repository.



Accept this location. Go to the next step.

10. The wizard asks you to select the project elements that will be added into the repository.



We want to add all project elements. Click "All" and go to the next step.

11. The wizard asks you to select the project dependencies that will be added into the repository. These dependencies correspond to all external elements required by the project (images, style sheets, etc.).

We want to add all project dependencies. Go to the next step.

12. Validate the project integration in the SCM. The project and its elements have now been added into our repository.

Remark

Sharing project elements

When the projects that share the same resources (same analysis, same windows, etc.) 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.

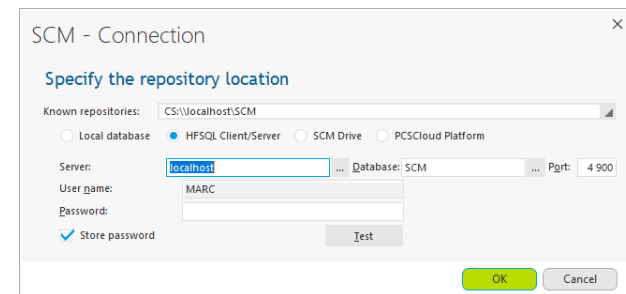
Opening a project from SCM

In our example, the project is integrated in the SCM and it can be used directly.

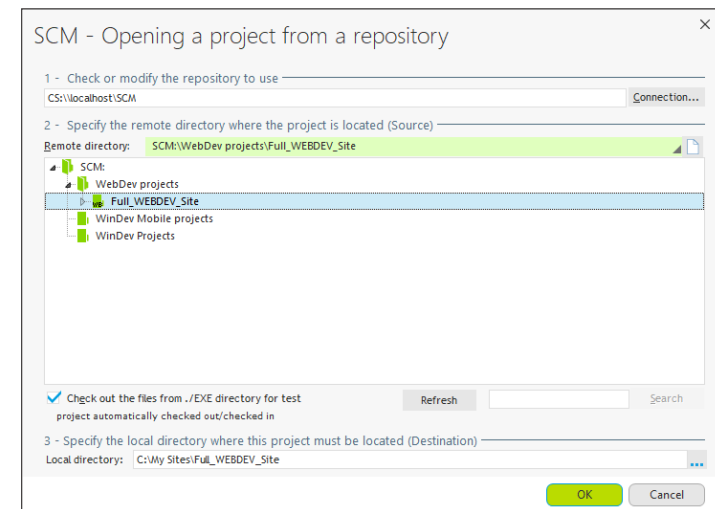
In a real case, in order for other developers to work on a project found in the Source Code Manager, they must get a copy of this project locally.

To do so, the following operations must be performed:

1. Open the project from the Source Code Manager: on the "Home" pane, in the "General" group, expand "Open" and select "Open a project from the SCM".
2. Specify the location parameters of the repository and validate (this step is required only if the current project in the editor does not belong to the SCM):



3. In the window that appears, indicate the connection and local directory if necessary:



Remark: If the project was already opened from SCM, the SCM proposes to open the project as usual or to overwrite the content (to retrieve the entire project).

This operation must be performed once only by each developer who is using the project. The developer who added the project into the Source Code Manager (you in this case!) has no operation to perform.



Remark

The next openings of a project managed by SCM are identical to the openings of a project not managed by SCM: all you have to do is open the project (".WWP" file) corresponding to the local copy.

Configuring SCM

Before you start working on the elements of project found in the SCM, you must configure the mode for checking out the project elements.

When working on the elements of a project in the SCM, the element must be checked out from the repository before it can be modified, then it must be checked back in once the changes have been made. Therefore, the modified element becomes available to all SCM users.

WEBDEV proposes two modes for checking out the project elements:

- **the standard mode:** if you display an SCM element that is not checked out, a panel indicates that the element must be checked out before it can be modified. The element can be checked out immediately (check-out button found in the dialog box).
- **the automatic mode:** if you try to modify an SCM element that is not checked out, the SCM automatically proposes to check it out. Once the check-out is validated, the element can be modified.

Remark: this mode is not recommended when using SCM with a slow Internet connection.

The automatic check-out will be used in this tutorial.

- ▶ To make sure that the automatic check-out is enabled, on the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV". In the "General" tab, check (if necessary) "Check out elements during the first modification".

LESSON 12.3. USING A PROJECT IN SCM

This lesson will teach you the following concepts...

- Modifying a project parameter.
- Modifying a project page.
- Synchronizing the project.
- Offline mode.
- SCM administrator.



Estimated time: 30 mn

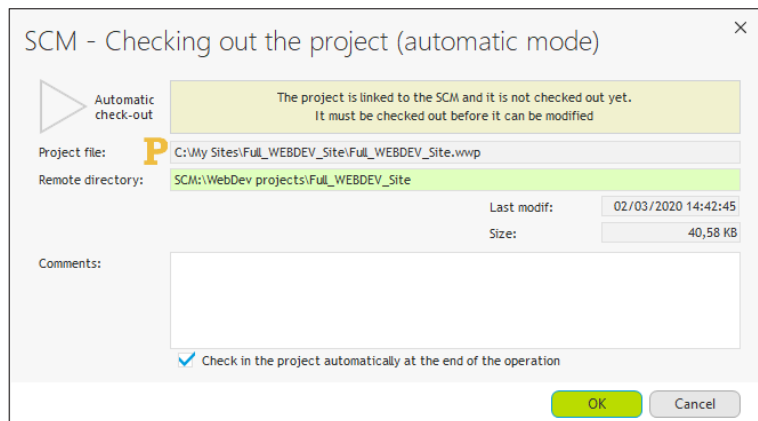
Overview

We are now going to work with SCM in real conditions by performing the following operations:

- Modify a project parameter.
- Modify a project page.

Modifying a project parameter

- ▶ We are going to modify the project by asking to display the skin on the dialog boxes:
 1. Display the project description: in the ribbon, on the "Project" pane, in the "Project" group, click "Description".
 2. Click on the "Skin" tab.
 3. Check "Apply the skin to the dialog boxes (YesNo and OKCancel)".
 4. Validate the description window of project.
- ▶ Several SCM windows appear:
 1. First of all, the window for automatic project check-out is displayed. Indeed, we want to modify a project characteristic therefore the project must be checked out.



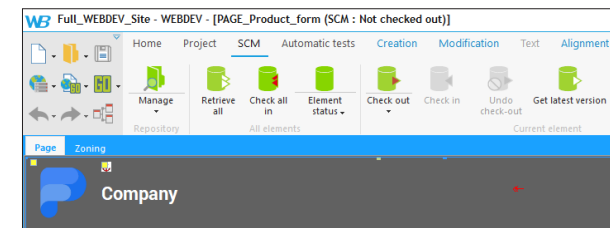
2. "Check in the project automatically at the end of the operation" allows automatically checking the project back in once the changes have been made. Keep this option.
3. Validate this window.
4. WEBDEV proposes to add elements into the repository. Accept.
5. The project description window is closed and the project is automatically checked back into the repository.


Modifying a project page

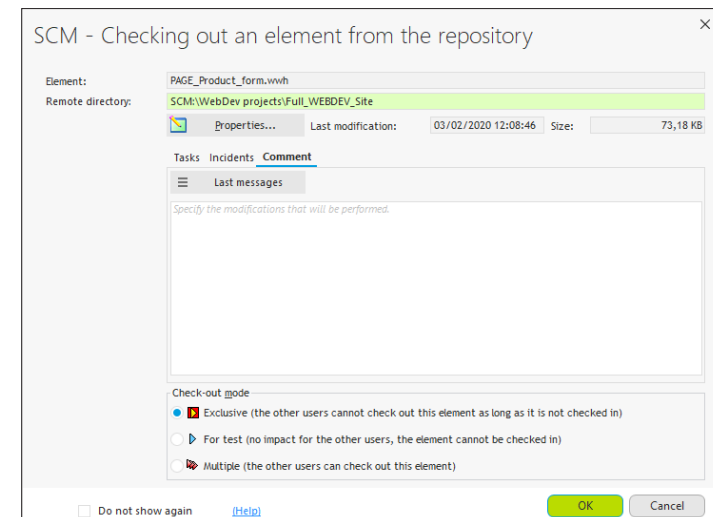
We are now going to modify the "PAGE_Product_form" page. We are going to move the "Cancel" and "Validate" buttons.

To modify a project element, this element must be checked out.

- ▶ To modify the "PAGE_Product_form" page:
 1. Select the "PAGE_Product_form" page in the project explorer and double-click the element in order to open it in the page editor.



2. The automatic check-out is enabled during the first modification: all you have to do is move a control to check the element out.
3. You also have the ability to click the "Check out" icon found in the ribbon of SCM pane ().
4. The check-out window is displayed:



5. The SCM proposes three check-out modes:
 - **Exclusive** (recommended mode): nobody can 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 also be checked out by other users. In this case, the differences between the different element versions can be viewed when the element is checked back in. This mode is reserved to specific cases and to experienced developers.
6. The page will be checked out in exclusive mode. Keep the "Exclusive" option checked.
 7. Type a comment ("Modifying the buttons" for example). This comment will be useful for the other developers.
 8. Validate the check-out.
The page is checked out.

Modifying the checked-out element

The method for modifying a checked-out element (UI, 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 in the repository will be used.

This allows you to make an application evolve while keeping a stable version in the repository.

► 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).

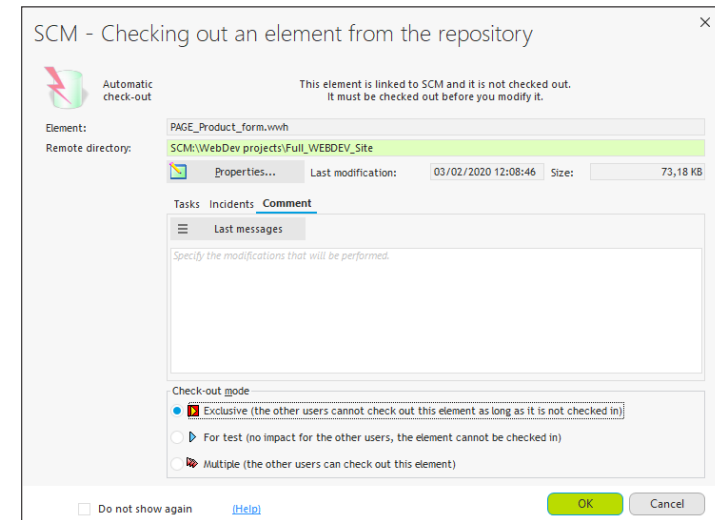
► Test your modifications.

Checking the checked-out element back in

Now that the modifications have been made, we are going to check the page back into the repository. 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 window is displayed:



This window is used to:

- find out the modifications performed by comparing the element found in the repository with the checked-out element ("My Modif" button).



Remark

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 repository ("Properties" button).
- type 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").



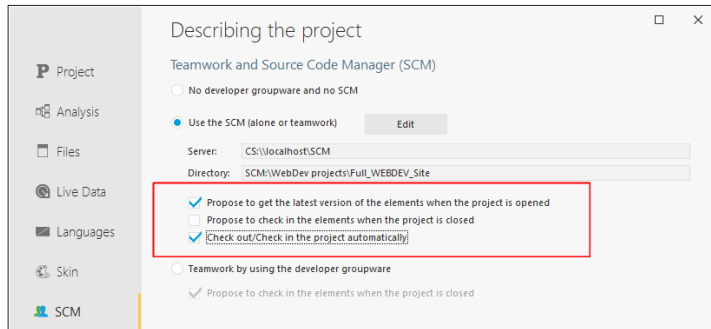
Remark

If you are using the Control Centers, the current task can be ended when the element is checked back into the Source Code Manager. This feature is useful to follow the monitoring of tasks, the corrections of bugs, ...

- Validate the check-in.

Synchronizing the project

Several options can be used to configure a project handled by SCM. These options are grouped in the "SCM" tab of project description (to display it, click "Description" on the "Project" pane).



These options are as follows:

- *Propose to get the latest version of the elements when the project is opened.*
When opening a project found in the SCM, this option proposes to get the latest version of the project elements.
- *Propose to check in the elements when the project is closed.*
When the project is closed, 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 the project is closed.
- *Check out/Check in the project automatically.*
This option allows you to automatically check out or check in the project when using an element.
This option is selected by default.

Offline mode

The SCM allows you to work in offline mode.

This mode allows a developer (who is using a laptop, for example) to continue working on a project found in the repository while being disconnected.

The principle is simple:

- before the disconnection, on the "SCM" pane, in the "Other actions" group, expand "Remote work" and select "Disconnect to work offline".
- 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 offline mode, there are two solutions to check 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 offline 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 offline mode.

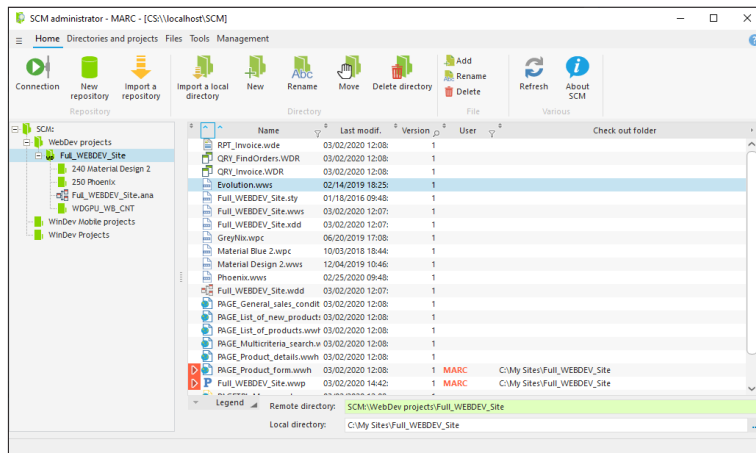
SCM administrator

The SCM administrator is used to directly handle the different projects included in the source code manager.

It allows you to:

- manage the repositories (creation, connection to a repository).
- manage the files and directories found in a project of repository (add, delete, rename, ... files and directories).
- manage the different files of the repository (check-in, check-out, share, etc.).
- start some tools (options, maintenance, etc.).
- show the history of an element.
- show the status of 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 "Repository" group, click "Manage". All project elements are listed in the administrator.



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

Disconnecting from SCM

To stop using SCM on a project:

1. Display the description window of project: on the "Project" pane, in the "Project" group, click "Description".
2. In the "SCM" tab, select "No developer groupware and no SCM".
3. Validate the description window of project.

The WLanguage
basics

APPENDIX 1. VOCABULARY

This lesson will teach you the following concepts...

- The terms used by WINDEV, WEBDEV and WINDEV Mobile.



Estimated time: 30 mn

Main terms used

AAF

Automatic Application Features.

Ajax

AJAX (Asynchronous Javascript and XML) is a technology used to refresh the data modified in an HTML page without having to redisplay the entire page.

Alignment

Method for organizing the controls in a window or page. For example, center a control in a page, define the same width for several controls, ...

Analysis

Description of the structure of data files and relationships.

Anchoring

Mechanism used to define the positioning or resizing rules in order for the window or page content to automatically adapt when a window or a browser is resized.

Application RAD

Fast development method of a program from a program template.

Array

Type of variable that contains several values. The values can be accessed via a subscript. The [] characters are used to access the array elements.

Assignment

Operation that consists in assigning a value to a variable or control. For example:

```
// Assign the value ADDAMS to CustomerName variable
CustomerName = "ADDAMS"
```

The = sign is the assignment operator.

AWP

Page format directly addressable (fixed URL), referenceable, without automatic context.

Block (report)

Element constituting a report. For example, a Page Header block, a Page Footer block, a Body block.

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, the customers are separated whenever changing city.

Class (OOP)

Element defined in Object-Oriented Programming. A class groups methods (actions) and members (data).

Classic (file)

Type of access to an HFSQL data file. An HFSQL data file is Classic when accessed directly in its directory.

Client/Server (file)

Type of access to an HFSQL data file. An HFSQL data file is Client/Server when accessed by connecting to a server that has this data file via the HFSQL engine.

Context (Page)

Section stored in memory on a Web server that was used to build a page that is visible in a browser.

Control (window or page)

Graphic element used to build the GUI of a program or site.

Control Template

Container of one or more controls (with processes) that can be indefinitely re-used in pages. Main template characteristic: if the initial template is modified, the modifications are automatically applied to the different template uses.

CSS

Language for describing the styles of the different elements found in an HTML page.

Data binding

Method that is used to associate a graphic GUI element to a data (variable, item).

Database

Element containing the program data. The data is organized in data files (also called tables).

Deployment

Action that consists in installing a program on the user computer.

Dynamic site

Project developed in WEBDEV containing static pages and dynamic pages (in Session or AWP mode). The dynamic pages (in Session or AWP mode) are used to manage the access to data stored in a database.

Editor

Program that is used to create a project element (window editor, page editor, ...).

Event-driven

Type of programming. A user action on a window or on a page induces a code to run. The code of the action to run is typed 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.

External component

Software block that is used to export one or more business rules in order to re-use them.

File (Table)

Element that constitutes a database. A data file is used to store the data entered into a program. For example, a CUSTOMER file will contain the customer information that was entered into a program.

Global (variable or procedure)

Corresponds to the scope in memory of a variable or procedure. A global element is accessible from any other project element. The opposite is local.

GUI

Graphical User Interface. Description of windows or pages that constitute a program.. This is what the user will see when using the program.

Homothetic

Method for resizing an image in order to maximize its display without distorting its content.

HTML

HyperText Markup Language.

Language used to describe the elements of a Web page.

Index (file)

Synonym: key.

Integrity constraint

Rule associated with a data file item to ensure data consistency in a database.

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.

Javascript

Programming language that is used by the browsers to perform calculations and processes in an HTML page (on the computer of Web user).

Key (file)

Characteristics of a data file item. A key item is used to optimize the searches and the sorts in a data file.

Synonym: index.

Link (analysis)

Used to describe the nature of relationship or the common point between 2 analysis data files. The integrity rules to write to the relevant data files can be defined depending on the specified link.

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 or structure.

Method

Procedure belonging to a class used to act on the class data (members).

Native access (connector)

Method for connecting to a database from a program.

n-tier

Programming method in layers. Each layer is independent and it can be changed without impacting the other ones.

Benefit: Simplified maintenance.

Drawbacks: Difficulty and development time.

Object-Oriented Programming (OOP)

Advanced programming method, opposed to procedural programming.

In OOP, we handle objects, which means grouped sets of variables and methods associated with entities that include these variables and these methods.

In procedural programming, we define functions that call each others. Each function or procedure is associated with a specific process that can be divided into sub-processes until we get basic functions.

OOP

Abbreviation of Object-Oriented Programming.

Page

Element of a WEBDEV project where the graphical site interface is defined. In most cases, a site includes several pages linked together.

Page template

Container representing a standard page that can be applied to one or more project pages.

Main template characteristic: if the initial template is modified, the modifications are automatically applied to the different template uses.

Parameter (window, page, procedure, method, ...)

Element expected in a window, page, procedure, method, ... during the call to this one. Each value passed in parameter must be assigned to a variable.

Popup menu

Drop-down menu containing the possible actions depending on the location where the right mouse click was performed and on the type of element onto which this right click was performed.

Private

Variable or procedure that can only be used in the current element.

Procedure

Project element containing the code of a process to run.

Project

Element that groups all the elements that constitute a program or a site. A project contains for example an analysis, pages, reports, queries, ...

Project code

Code run when starting a program or a site.

Project configuration

Description of output format of project: Windows executable, Linux, JAVA, ...

Property (control, window, ...)

Keyword representing an element characteristic. The properties are used to handle and modify the characteristics of project elements by programming.

Public

Variable or procedure that can be used from all elements.

Query

Element written in SQL language that is used to access (in read-only or in read/write) the content of a relational database.

RAD

Acronym for Rapid Application Development

Fast development method of a program from an analysis (description of data files).

Report

Project element that defines a printout.

Report template

Container representing a standard report that can be applied to one or more project reports.

Main template characteristic: if the initial template is modified, the modifications are automatically applied to the copies.

SCM

Source Code Manager. Tool for organizing, sharing the source codes of projects, managing rights, ...

Skin

Style book of a WEBDEV site.

Element in which is defined the graphic layout of the pages found in a WEBDEV project.

Static site

Project developed in WEBDEV containing static pages only, which means pages that do not have access to a database.

Structure

Type of variable that includes several sub-variables.

Style

Element used to describe the graphic style of a control in a window or in a page. A style includes for example a type of font, the size of characters, the color of characters, etc.

Style sheet

Contains the list of styles used in a project.

Table (control)

Graphic element found in a window or page. A table control includes one or more columns and rows.

Table (data file)

Element that constitutes a database. A table is used to store the data typed in a program. For example, a CUSTOMER table will contain the customer names and addresses that have been typed in a program.

User groupware

Tool for describing and managing the access rights to GUI for the users of a program or site. For example, prevent a user from being able to click a "Delete" button according to his login or to his group.

Variable

Element used to store a program value in memory. Several types of variables are available. Each type corresponds to the nature of the value that will be stored. For example, a string variable to store the name of a person, a currency variable to store an amount.

Webservice

Program installed on a Web server whose processes are accessible via the Web.

XML

Language for organizing data in order to normalize and simplify the exchange of data (mainly used in the exchanges with the Webservices).

Zoning

Method used to define the splitting of a Web page.

APPENDIX 2. THE VARIABLES

This lesson will teach you the following concepts...

- What is a variable?
- The different types of variables.
- The scope of variables.
- The String type in details.
- The Array type in details.



Estimated time: 30 mn

What is a variable?

In a programming language, a variable is used to store data. These memory sections contain strings, numbers, etc.

The variables are used to perform calculations, to perform comparisons or to store information that will be used later.

Declaring a variable

```
Price is currency
```

↑ ↑

Variable Variable
name type

Initializing a variable

```
Price = 500.32
```

↑

Variable value

A variable is represented by:

- a name: Name given to the variable so that it can be used by the language.
- a type: Nature of data stored in the variable (see The types of variables).
- a value: Information stored in the variable.
- a scope: Limit for using the variable in the program (see The scope of variables). The scope is mainly defined by the location where the variable is declared.

Declaring a variable

The variable must be declared (which means created) before it can be used.

- Example of simple declaration:

```
Price is currency
```

- **Price** represents the variable name.
- **is** is used to declare the variable. The everyday language is used in WLanguage.
- **currency** corresponds to the variable type.
- Example of multiple declaration:

```
LastName, FirstName are strings
```

- **LastName, FirstName** represent the names of variables.
- **are** is used to declare a set of variables.
- **strings** represents the type of variables.

Assignment and use

When the variable is declared, you have the ability to assign it (or to give it a value).

For example:

```
// Assign a currency variable
Price = 1256.67
// Assign a string variable
LastName = "Doe"
```

The = operator is used to perform this assignment.



Remark

In WLanguage, the " character (double quote) is the character used to delimit a character string. In the above example, the doubles quotes are used to assign the Doe value to the LastName variable.

The variable content can be read and handled: all you have to do is use the name given to the variable to access it.

The following example is used to read and display the content of Price variable on the screen:

```
Info(Price)
```

The types of variables

The variable type is used to specify the kind of information that will be stored in the variable. The most common types are:

- boolean (True or False),
- string ("Doe"),
- integer (1234),
- currency (12,32),
- real (7,766666),
- etc.



Important

Use the type corresponding to the information that must be stored. Therefore, you will optimize the memory and you will avoid calculation or process errors when using variables in the WLanguage functions.

See the online help regarding the relevant type for more details.



Remark

Other types are available, such as arrays, structures, dates, times, ... Advanced variables are also available. These advanced types group all characteristics of the element currently used in a single variable. Advanced types can be used to handle XML documents, emails, XLS files, ... This type of variable will be used later in this tutorial.

The scope of variables

The variables can be declared anywhere in the code. However, according to the position of its declaration, the variable cannot be used to perform processes or calculations. We talk of variable scope.

Two types of scope are available:

- Global.
- Local.

Global scope

Global means that the variable has an extended visibility in the code. The variable is visible outside the location where it was declared. Several levels are available:

- Project and Set of procedures,
- Window, Mobile Window, Page, Report.

A variable declared **at project level** has the greatest visibility in the program. The variable is visible anywhere, in all program processes. However, you should not declare too many variables with this scope ; indeed, the memory occupied by the variable is always reserved even if the variable is not used. Using a large number of global variables is not recommended in the program architecture. To pass variables to a process, we recommend that you use parameters (see " Procedure parameters", page 460) for more details.

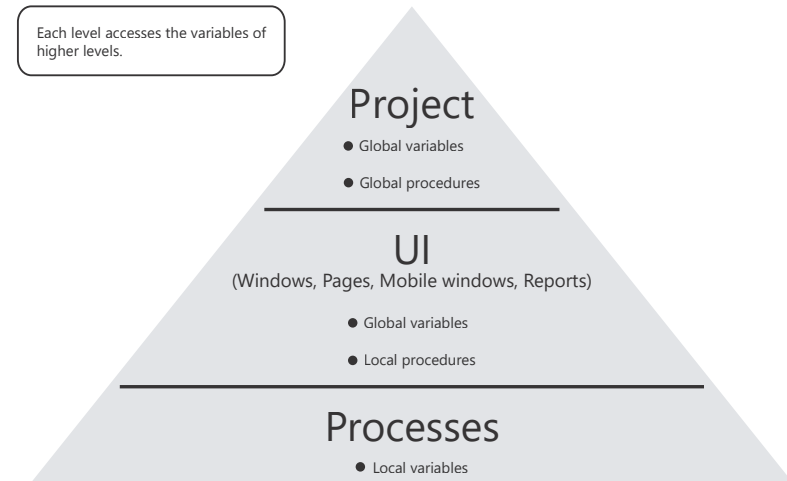
A variable declared **at Set of Procedures level** and a variable declared at project level have the same visibility. The advantage of declaring a variable at Set level is to group (or classify) the variables by theme in order to make the "Initializing" project event more readable.

A variable declared **at Window, Mobile Window, Page or Report level** limits the variable scope to all the events of the element (Window, Mobile Window, Page or Report) and its controls. This makes it possible to encapsulate and limit the uses.

Local scope

Local means that the variable has a limited visibility in the code. The variable is visible in the process where it was declared. This makes it possible to restrict the use of variable to the process.

Summary scope diagram



A variable is global when it is declared:

- in the initialization code of project (or in the declaration code of set of procedures). The variable is global to the project.
- In the declaration code of global variables of window, page or report. The variable is global to the element (window, page or report) where it was declared.

In all other cases, a variable is local to the process where it is declared.

Simple operations on the variables

Several mathematical operators can be used to perform calculations on variables:

- + to perform an addition.
- - to perform a subtraction.
- * to perform a multiplication.
- / to perform a division.

Other operators can be used to perform calculations:

- ++ to increment from 1 (add 1 to the variable).
- -- to decrement from 1 (subtract 1 from the variable).
- += to assign by adding a value.
- -= to assign by subtracting a value.

Examples:

```
// Declaration of variables
Cnt is int
V1 is int
Res is numeric

// Assignment
Cnt = 10
V1 = 3

// Use of operators
Cnt = Cnt + 3 // Cnt is equal to 13
Cnt ++ // Cnt is equal to 14
Cnt -= 8 // Cnt is equal to 6
Cnt = Cnt * V1 // Cnt is equal to 18
Res = Cnt / 5 // Res is equal to 3.6
```

Comparison operators are also available:

- < less than,
- > greater than,
- <= less than or equal to,
- >= greater than or equal to,
- <> different from,
- = equal to.

Other operators are available. See the online help for more details (keyword: "Operators").

Tips

- It is very convenient to name the variables with long names (and to avoid short names such as i, j, k, ...). When reading the program again, you will be able to easily remember the variable purpose.
- To define the name of variables, all Unicode characters (including the accented characters) are accepted. Meaning improved readability! Caution: some characters are not allowed: space, =, dot, comma, ...
- It is very important to give the proper type to the variable according to its use. For example, to store several digits, you may have to:
 - use a numeric variable if this variable must be used for calculations.
 - use a string variable if this variable must be used to store digits without performing calculations (to store the social security number for example).

Details of variable type: the String variables

The String variables are the most often used types of variables.

Let's present in details some features available for this type of variable.

The String type

The String type is used to store and handle characters and character strings.

We have already seen how to initialize a string variable:

```
LastName is string
// Assign a string variable
LastName = "Doe"
```

There is no need to declare the string length: this length automatically adapts when using the variable.



Tip

To initialize a string variable with a text on several lines, use the following syntax:

```
<Variable name> = [
    <Text of line 1>
    <Text of line 2>
]
```

For example:

```
MyString is string
MyString = [
    Example of
    multi-line string
]
```

You also have the ability to assign a string variable with the content of a control handling strings. The following code is used to assign a string variable with the content of an edit control:

```
LastName is string
// Assign a string variable
// with the content of EDT_LastName edit control
LastName = EDT_LastName
```

In addition to the main comparison operators, several powerful operators are used to manage the extractions and concatenations in advanced mode.

Some examples:

- "+" : To concatenate strings
- "~=" : To check the flexible equality

Specific WLanguage functions are used to perform various operations: search, extraction, size, switch to uppercase characters, ...

Examples:

```
str is string
str = "WEBDEV is a great tool"
// Extract a sub-string from left
Info(Left(str,6)) // Displays "WEBDEV"
// Extract a sub-string from right
Info(Right(str,4)) // Displays "tool"
```



Remark

The different WLanguage functions can be nested. A WLanguage function can be used as parameter of another WLanguage function.

For example:

```
Info(Middle(Left(str,13),8,2)) // Displays "is"
```

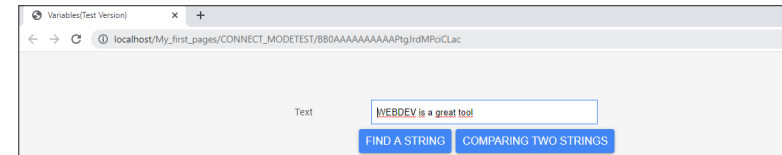
See the help about the character strings and about the functions for handling character strings for more details.

Practical example



To handle the different concepts of this lesson, we are going to create different windows. These windows will be created in the "My_First_Pages" project.

- ▶ Open the "My_First_Pages" project if necessary.
 1. Display the WEBDEV home page (Ctrl + <).
 2. Click "Tutorial", then "My first pages (Exercise)".

To handle the Character String variables, we are going to create the following page:




This page is used to:

- find a string inside another one.
 - compare two strings.
- ▶ Create a new blank page:
 1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Page" then "Page". The page creation wizard appears.
 2. Select "Blank".
 3. Validate. The page is automatically created in the editor. The window for saving an element is displayed.
 4. Specify the page title: "Variables". The page name ("PAGE_Variables") is automatically proposed.
 5. Click on the "OK" to validate the information displayed in the save window.
 - ▶ To create the edit control containing the string:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click the location where the edit control will be created (at the top, in the middle of the page for example).
 3. Right-click the control and select "Description".
 4. In the "General" tab, specify:
 - The control name: "EDT_Text".
 - The control caption: "Text".
 5. Select the "Content" tab. This tab is used to define the default value of edit control. Type "WEBDEV is a great tool".
 6. Validate the description window. The text typed is directly displayed in the control.

Finding a string

► To create the "Find a string" Button control:


1. On the "Creation" pane, in the "Usual controls" group, click .
2. Click the location where the Button control will be created (below the edit control for example).
3. Click the control that was just created. The text displayed in the control becomes editable. Type the caption: "Find a string".
4. Press Enter to validate the input.
5. Display the events associated with the Button control (press F2).
6. Write the following code in the server "Click" event of the button:

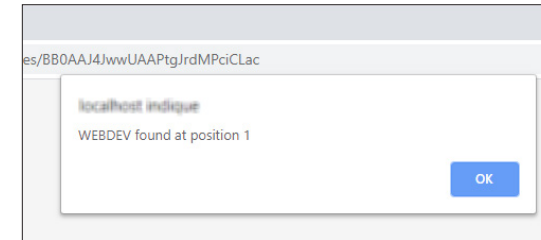
```
sSoughtString is string = "WEBDEV"
nPos is int
nPos = Position(EDT_Text, sSoughtString)
IF nPos = 0 THEN
    Info(sSoughtString + " not found in the text")
ELSE
    Info(sSoughtString + " found at position " + nPos)
END
```

Let's study this code:

- Two variables are declared: a String variable corresponding to the sought string and an Integer variable corresponding to the position of sought string.
 - **Position** is used to find a string inside another one. In our example, we search for the content of sSoughtString in the content of EDT_Text control.
 - The IF statement is used to check the result of **Position**. This type of statement will be presented later in this section on WLanguage.
 - If **Position** returns 0, it means that the sought string was not found. A message is displayed by **Info**.
7. Close the code window (click X at the top right corner of code editor).

► Let's now run the page test:


1. Click  among the quick access buttons (or press F9).
2. The window that was just modified is automatically saved then it is started in execution. Click on the button. The WEBDEV word is found.



3. Modify the content of edit control (by replacing WEBDEV by WB for example) and click the button. The WEBDEV word is not found.
4. Close the browser.

Comparing two strings

► To create the "Compare two strings" Button control:


1. On the "Creation" pane, in the "Usual controls" group, click .
2. Click the location where the Button control will be created (for example, to the right of the existing Button control).
3. Click the control that was just created. The text displayed in the control becomes editable. Type the caption: "Compare two strings".
4. Press Enter to validate the input.
5. Adapt the control size.
6. Display the events associated with the Button control (press F2).
7. Write the following code in the event "Click on (Server)":

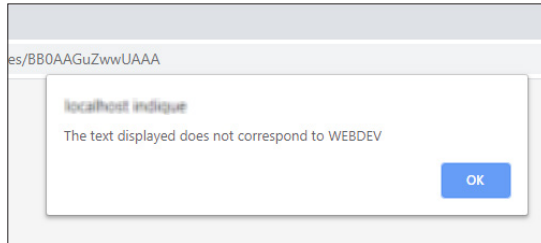
```
sStringToCompare is string = "WEBDEV"

IF EDT_Text ~= sStringToCompare THEN
    Info("The displayed text corresponds to " + sStringToCompare)
ELSE
    Info("The displayed text does not correspond to " + sStringToCompare)
END
```

Let's study this code:

- A String variable is declared. This variable contains the string to compare.
- The `~=` operator corresponds to a flexible equality. This operator is used to perform a comparison while ignoring the case, the spaces found before and after the string and the accented characters.
Remark: to type the `~=` sign, press AltGr + 2.
- If the comparison is true, the edit control corresponds to the word found in the string to compare regardless of the case used.

8. Close the code window (click X at the top right corner of code editor).
- Let's now run the page test:
 1. Click  among the quick access buttons (or press F9).
 2. Click the "Compare two strings" button. The message indicates that the comparison is not correct.



3. In the edit control, type "WEBDEV" in uppercase characters and click the "Compare two strings" button. The WEBDEV word is found.
4. Modify the content of edit control (by replacing "WEBDEV" by "WEBDEV" for example) and click the button. The WEBDEV word is also found.
5. Close the browser.

Details of another variable type: the arrays

The arrays are a common type of variable.

An array is a structured type that is used to group a set of elements of the same type. Each array element can be directly accessed by its subscript.

Specific WLanguage functions are used to handle the arrays and their elements. These functions start with "Array".

Declaration

The declaration of an Array variable is performed as follows:

```
<Array name> is array of <Type of array elements>
```

For example:

```
arrString is array of strings
arrInt is array of int
```

Filling an array and accessing the elements

During its declaration, the array is empty. The Add function is used to add elements via the following syntax:

```
Add(<Array name>, <Element value>)
```

To access the array elements, use the following syntax:

```
<Array name> [<Element subscript>]
```



Important

The subscripts of array elements start from 1.

Example:

```
// Create an array of strings
MyArray is array of strings
// Add elements
Add(MyArray, "WINDEV")
Add(MyArray, "WEBDEV")
Add(MyArray, "WINDEV Mobile")
// Display the content of third element
Trace(MyArray[3]) // Displays "WINDEV Mobile"
```



Tip

Fast array initialization

To initialize an array, you also have the ability to use the following syntax:

```
// Declares an array
arrDay is array of strings
// Initialization with
// the names of the days of week
arrDay = ["Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday", "Sunday"]
```

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

Advanced arrays

Advanced arrays are also available: array with several dimensions, array of arrays, associative array, array of structures, array of objects, ...

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

APPENDIX 3. THE CONDITIONAL STATEMENTS

This lesson will teach you the following concepts...

- The IF statement.
- The SWITCH statement.
- Practical example.

 Estimated time: 30 mn

Overview

WLanguage is a set of statements used to handle data.

The conditional statements are used to compare variables and/or values between themselves in order to perform different processes. Several conditional statements are available:

- IF... THEN... ELSE... END
- SWITCH...

The IF and SWITCH statements

The IF statement

This statement is used to run an action or another one according to the result of an expression. If the expression is checked, a process is run ; if the expression is not checked, another process can be started.

The IF statement can be used as follows:

```
IF <Expression to check> THEN
    Process to run if the expression is checked
ELSE
    Process to run otherwise
END
```

Code example: The following code selects a number at random and displays a message according to the value.

```
Tot is Currency
// Selects a number at random between 100 and 4000
Tot = Random(100, 4000)
IF Tot>2000 THEN
    Info("The amount is greater than 2000")
ELSE
    Info("The amount is less than or equal to 2000")
END
```

In this case, the expression to check corresponds to "Tot>2000".

Remark: Several code lines can be run during the process corresponding to a condition. In this case, the following syntax must be used:

```
IF <Expression to check> THEN
    Code line 1
    Code line N
ELSE
    Code line 1
    Code line N
END
```


The SWITCH statement

This statement is used to evaluate an expression and to run a process for each possible expression value.

The SWITCH statement can be used as follows:

```
SWITCH <Expression>
CASE Value 1:
  Process 1...
CASE Value 2:
  Process 2...
...
CASE Value N:
  Process N...

OTHER CASE
  Process ...
END
```

Example: The following code retrieves today's date and displays a different message according to its value. A specific message is displayed for the 1st and for the 15th of the month. In the other cases, today's date is displayed.

```
D is Date
D = Today()
SWITCH D..Day // Checks the day of the date
CASE 1: Info("We are the first day of the month")
CASE 15: Info("We are the 15th of the month")
OTHER CASE:
  Info("We are the: " + DateToString(D))
END
```

Remarks:

- If the code line "CASE 1:..." is run, the other code lines corresponding to the possible values are not run.
- Several values can be grouped in the same case. The different values are separated by a comma. For example:

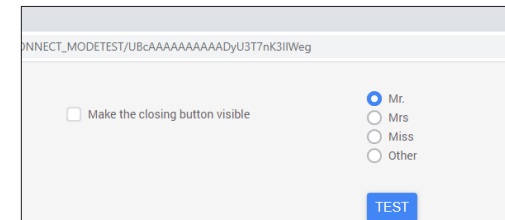
```
Sub is int = 2
SWITCH Sub
CASE 1,2: Info("Case 1 or 2")
CASE 3: Info("Case 3")
OTHER CASE: Info("Other case")
END
```

- Several code lines can be run during the process corresponding to an expression value. In this case, the following syntax must be used:

```
SWITCH <Expression>
CASE Value 1:
  Process 1 - Code line 1...
  Process 1 - Code line 2...
CASE Value N:
  Process N - Code line 1...
  Process N - Code line 2...
OTHER CASE
  Process ...
END
```

Practical example: Using the IF and SWITCH statements

To use the conditional instructions we have just studied, we will create the following page:



The screenshot shows a web form with a title bar containing the text "CONNECT_MODETEST/UBcAAAAAAAAAADyU3T7nK3IIWeg". The form contains a checkbox labeled "Make the closing button visible" which is currently unchecked. To the right of the checkbox are three radio buttons labeled "Mr.", "Mrs.", "Miss", and "Other". The "Mr." radio button is selected. At the bottom right of the form is a blue button labeled "TEST".

Two operations can be performed:


- If the user clicks the check box, the "Close" button is displayed.
- If the user clicks the "Test" button, the box checked in the check box is displayed.

Project used

To handle the different concepts of this lesson, we are going to create different pages. These pages will be created in the "My_First_Pages" project.


- ▶ Open the "My_First_Pages" project if necessary.
 1. Display the WEBDEV home page (Ctrl + <).
 2. Click "Tutorial", then "My first pages (Exercise)".

Creating the page

- ▶ Create a new blank page:
 1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Page" then "Page". The page creation wizard appears.
 2. Select "Blank".
 3. Make sure that "Responsive Web Design mode" is unchecked.
 4. Validate. The page is automatically created in the editor. The window for saving an element is displayed.
 5. Specify the page title: "Conditional statements". The page name ("WIN_Conditional_statements") is automatically proposed.
 6. Click on the "OK" to validate the information displayed in the save window.

Creating the page controls for the conditional IF statement


Two controls must be created:

- a Check Box control, to display (or not) the "Close" Button control.
 - a "Close" Button control, to close the browser window.
- ▶ To create the Check Box control:
 1. On the "Creation" pane, in the "Usual controls" group, click "Check Box" then click the position where the control will be created (at the top of the page for example)
 2. Click the control that was just created: the "Option 1" caption becomes editable.
 3. Type the option caption: "Make the closing button visible".
 4. Press Enter to validate the input.
 - ▶ To create the "Close" Button control:
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click the location where the Button control will be created (below the Check Box control for example).
 3. Click the control that was just created. The text displayed in the button becomes editable. Type the caption: "Close".
 4. Press Enter to validate the input.
 - ▶ This Button control allows you to close the browser window. We are going to type the corresponding WLanguage code:
 1. Select the "Close" Button control.
 2. Press F2: the code editor displays the events associated with the control.
 3. Write the following code in the event "Click on BTN_Close (browser code)":

```
BrowserClose()
```

BrowserClose is used to close the current window of Internet browser.

4. Close the code editor.

- ▶ The "Close" Button control must be invisible when the page is opened. This information corresponds to its "initial status". We are going to modify this status in the description window of control:
 1. Select the "Close" Button control.
 2. Display the popup menu of control and select "Description".
 3. In the "UI" tab, modify the initial status of control by clicking "Visible". This option is now unchecked.
 4. Validate the control description window.
 5. The "Close" Button control is still visible in the editor. Its status was changed in execution only.
 6. Save the modifications by clicking  among the quick access button (on the left of ribbon) or by pressing Ctrl + S.

Conditional IF statement

In our example, a click on the check box must trigger the display of Close button. We are going to type the WLanguage code associated with the check box.

- ▶ To type the WLanguage code associated with the Check Box control:
 1. Select the Check Box control.
 2. Press F2.
 3. Write the following code in the event "Exit with modification of CBOX_NoName1 (browser code)":

```
IF CBOX_NoName1 = True THEN
  BTN_Close..Visible = True
ELSE
  BTN_Close..Visible = False
END
```

Let's study this code:

- CBOX_NoName1 and BTN_Close respectively correspond to the names of Check Box and Button controls.



Tip

To handle the current control in one of the events associated with it, you can use the keyword **MySelf**.

In our example, you also have the ability to write:

```
IF MySelf = True THEN
```

- The Check Box control is a 2-state control: checked/unchecked. It corresponds in programming to a Boolean variable. If the control is checked, its value is set to **True**; if the control is unchecked, its value is set to **False**.
- This code tests the value of the Check Box control.
 - If Check Box control is checked, Button control BTN_Close becomes visible using **..Visible**.
 - If the Check Box control is unchecked, the Button control BTN_Close becomes invisible.



Remark


Several functions and properties are available in WLanguage. The functions can accept parameters and they return results. The properties are directly applied to the controls via the following syntax:

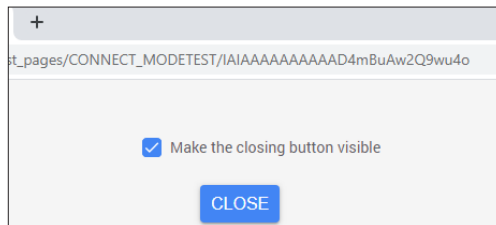
```
<Control name>..<Property name>
```

In our example, **..Visible** is used on the **BTN_Close** control. The online help of a function or property can be displayed at any time by pressing **F1**. See **How to access the online help?** for more details.

4. Close the code window (click X at the top right corner of code editor).

► Let's now run the page test:

1. Click  among the quick access buttons (or press **F9**).
2. In the page in execution, click the check box. The "Close" button becomes visible.



3. Click the "CLOSE" button: the browser tab is closed and the page editor is displayed.



Remark


Depending on your browser configuration, a window may appear asking for confirmation to close the browser.

Creating the page controls for the conditional SWITCH statement

Two controls must be created:

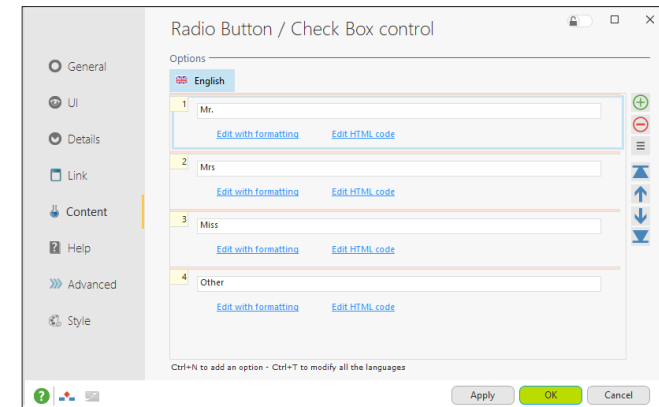
- a Radio Button control used to select the value to check.
- a Button control used to run the test.

► To create the Radio Button control:

1. On the "Creation" pane, in the "Usual controls" group, click "Radio button".
2. Click inside the page to create the control (on the right of the Check Box control created beforehand for example).
3. Display the popup menu of the control (right mouse click) and select "Description".
4. In the "Content" tab, add two new options by clicking the button .


5. Specify the captions for the different options:

- "Option 1" becomes "Mr".
- "Option 2" becomes "Mrs".
- "Option 3" becomes "Miss".
- "Option 4" becomes "Other".



6. Validate the description window of the Radio Button control ("OK").

► To create the "Test" Button control:

1. On the "Creation" pane, in the "Usual controls" group, click .
2. Click the location where the Button control will be created (below the Radio Button control for example).
3. Click the control that was just created. The text displayed in the Button control becomes editable. Type the caption: "Test".
4. Press **Enter** to validate the input.

SWITCH statement

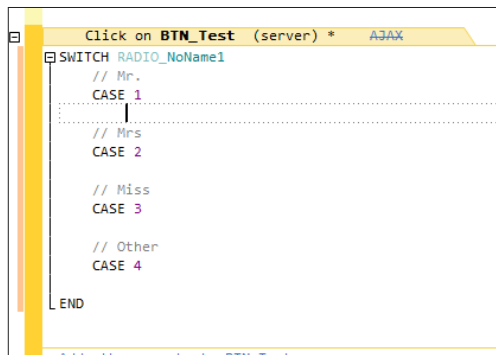
In our example, the Test Button control is used to test the selected value and to display a message.

► We are going to enter the WLanguage code associated with the Test Button control:

1. Select the Test Button control.
2. Press F2.
3. Write the following code in the event "Click BTN_Test (Server code)":

```
SWITCH RADIO_NoName1
```

4. When the Enter key is pressed to go to the next line, the code editor automatically displays the different SWITCH capabilities:



```
Click on BTN_Test (server) * AJAX
SWITCH RADIO_NoName1
// Mr.
CASE 1
// Mrs
CASE 2
// Miss
CASE 3
// Other
CASE 4
END
```

5. Write the following code:


```
SWITCH RADIO_NoName1
// Mr
CASE 1
  Info("You've selected Mr")
// Mrs
CASE 2
  Info("You've selected Mrs")
// Miss
CASE 3
  Info("You've selected Miss")
// Other
CASE 4
  Info("You've selected Other")
END
```

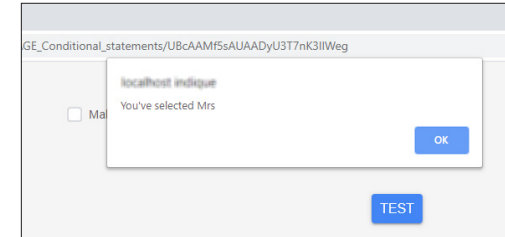
Let's study this code:

- RADIO_NoName1 corresponds to the name of Radio Button control.
- The Radio Button control is a control proposing several options (4 in our case). A single option can be checked at a time. The Radio Button control corresponds in programming to an Integer variable. Each menu option is associated with a value. If this option is checked, the Radio Button control takes for value the identifier of checked option.
- This code tests the value of the Radio Button control. A message is displayed according to the value of Radio Button control.

6. Close the code window (click X at the top right corner of code editor).

► Let's now run the page test:

1. Click  among the quick access buttons (or press F9).
2. In the page whose test is run, select an option and click "TEST": the message corresponding to the selected option is displayed.



3. Close the browser.

APPENDIX 4. THE LOOPS

This lesson will teach you the following concepts...

- The FOR statement.
- The LOOP statement.
- The WHILE statement.
- Practical example.



Estimated time: 30 mn

Overview

The loop statements are used to run a process in a recurring way. A specific loop statement is used according to the number of occurrences. Several statements can be used to perform loops:

- FOR...
- LOOP...
- WHILE...

The FOR statement

The FOR statement is used when the number of occurrences to process is known. This statement is used to manage the number of occurrences via a variable that counts the passages performed in the loop.

The syntax of FOR statement is as follows:

```
FOR Subscript = Start Value TO End Value
  Process to run
END
```

For example, the following code runs the process 2000 times:

```
FOR Cnt = 1 TO 2000
  // Process to run
END
```

Remark: An increment step of subscript can be defined via the STEP keyword. For example, the following code runs the process 2000 times and the Cnt variable decreases by 10:

```
FOR Cnt = 2000 TO 1 STEP -10
  // Process to run
END
```

The LOOP statement

The LOOP statement is used to perform loops when the number of occurrences to process is unknown. In this case, a test must be used to exit from the loop.

The syntax of LOOP statement is as follows:

```
LOOP
  Process to run
  IF <Expression> THEN BREAK
END
```

For example:

```
Counter is int
Counter = 10
LOOP
  // Process to run
  Counter = Counter - 1
  IF Counter = 0 THEN BREAK
END
```



Tip

The LOOP statement and the FOR statement can have the same behavior: all you have to do is use the syntax with exit according to the number of iterations:

```
LOOP (<Number of iterations>)
...
END
```

Example:

```
LOOP(10)
  // Process to run
END
```

The WHILE statement

The WHILE statement and the LOOP statement operate according to the same principle. The difference is that the test of exit condition is performed BEFORE running the loop code. This test is used to compare a variable. This variable starts from a start value and it is modified in the loop until it reaches the value that triggers the exit from the loop.

The syntax of WHILE statement is as follows:

```
<Initialize the variable to its start value>
WHILE <Compare the variable to its end value>
  Process to run
  <Modify the variable>
END
```

For example:

```
Counter is int
Counter = 0
WHILE Counter<10
  // Process to run
  Counter = Counter + 1
END
```

Practical example: Using loops


To check the use of loops, we are going to create a page into which an Image control will be moved.

Project used

To handle the different concepts of this lesson, we are going to create different pages. These pages will be created in the "My_First_Pages" project.

- ▶ Open the "My_First_Pages" project if necessary.
 1. Display the WEBDEV home page (Ctrl + <).
 2. Click "Tutorial", then "My first pages (Exercise)".

Creating the page

- ▶ Create a new blank page:
 1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Page" then "Page". The page creation wizard appears.
 2. Select "Blank".
 3. Validate. The page is automatically created in the editor. The window for saving an element is displayed.
 4. Specify the page title: "Loops". The page name ("PAGE_Loops") is automatically proposed.
 5. Click on the "OK" to validate the information displayed in the save window.

Creating controls


We are going to create the Image control then the buttons used to handle this image.

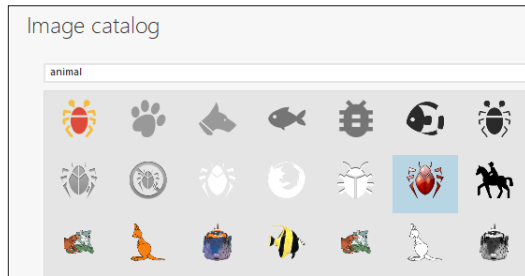
- ▶ To create an Image control:
 1. On the "Creation" pane, in the "Usual controls" group, click "Image". Position the control in the page.
 2. Right-click the control and select "Description".
 3. In the "General" tab, modify the control name: this control is named IMG_Animal.
 4. We are going to associate an image to the tab via the image catalog of WEBDEV.




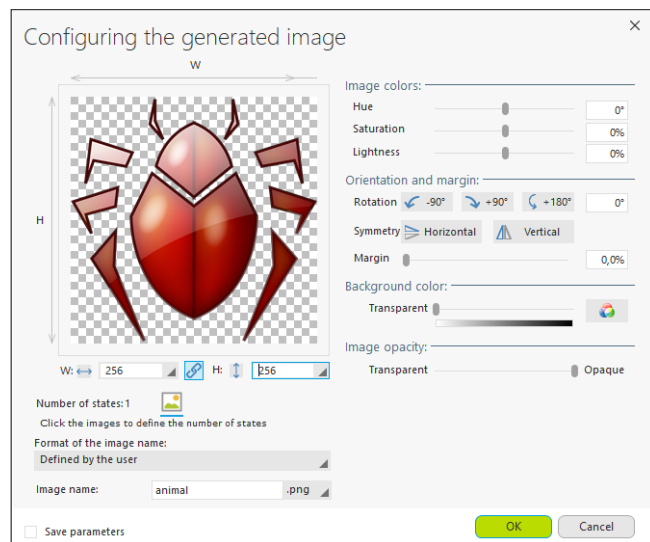
Remark


As soon as an image can be displayed in a control or page, WEBDEV proposes to use the image catalog. This image catalog is started via the "Catalog" option (available by clicking the  button). This catalog contains hundreds of images, cliparts, ...


5. Click the  button on the right of "Image" control. Select "Catalog" from the popup menu that is displayed. The window of image catalog is displayed.
6. Type "Animal" in the search area. Click the magnifier to start the search.



7. Select  and validate ("OK").
8. A window used to configure the image is displayed.



9. Keep the default options and validate.
10. The image path is displayed in the description window of Image control.
11. In the "UI" tab, "Overlay and move" area, check "The control can be overlaid".
12. Validate the description window of control.
13. Position the Image control at the top left of the page.
14. Save the page (press Ctrl + S or click  among the quick access buttons).

- ▶ We will now create a Button control to allow the image to move by 300 pixels to the right in the page. To do so, we will be using the FOR statement of WLanguage.
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click the location where the Button control will be created (below the Image control for example).
 3. Click the control that was just created. The text displayed in the control becomes editable. Type the caption: "FOR statement".
 4. Press Enter to validate the input.
 5. Display the description window of control (press Alt + Enter).
 6. In the "General" tab, in the "Button action" area:
 - for the "Type" option, select "Run browser codes only".
 - for the "During the action" option, select "Do not send anything to the server".
 7. Validate the description window of control.
 8. Display the events associated with the control (F2).
 9. Enter the following code in the "Click (browser)" event:


```
// Moves the image horizontally by 300 pixels
FOR i = 1 TO 300
  IMG_Animal..X++
END
```

This code is used to modify the X coordinate of Image control (..X) in a loop from 1 to 300. At each loop turn, the X coordinate is increased by one pixel.



Remark

When typing conditional statements in the code editor, the different possible syntaxes are proposed in a list by the code editor. If you select one of these syntaxes, the statement structure is automatically inserted into the code editor. Simply write the code that corresponds to each part of the statement.

10. Close the code window (click X at the top right corner of code editor).
- ▶ Now let's create a Button control to move the image until the image border is positioned at 1200 pixels. To do so, we will be using the WHILE statement of WLanguage.
 1. On the "Creation" pane, in the "Usual controls" group, click .
 2. Click the location where the control will be created (for example, to the right of the "FOR statement" Button control).
 3. Click the control that was just created. The text displayed in the control becomes editable. Type the caption: "WHILE statement".
 4. Press Enter to validate the input.
 5. Display the description window of control (press Alt + Enter).
 6. In the "General" tab, in the "Button action" area:
 - for the "Type" option, select "Run the browser click code only".
 - for the "During the action" option, select "Do not send anything to the server".
 7. Display the events associated with the control (F2).

8. Enter the following code in the "Click (Browser)" event:

```
WHILE IMG_Animal..X < 1200
  IMG_Animal..X++
END
```

This code is used to modify the X coordinate of Image control (..X) until a condition is true. In our case, this condition is as follows: the position of the Image control (..X) must correspond to 1200.



Tip

When a code line is too long to be displayed in the window of code editor, you have the ability to cut it in 2 by using a carriage return.

9. Close the code window (click X at the top right corner of code editor).

► Now let's create a last Button control to move the Image control until it reaches the left border of the page. To do so, we will be using the LOOP statement of WLanguage.

1. On the "Creation" pane, in the "Usual controls" group, click **Ok**.
2. Click the location where the button will be created (for example, on the right of WHILE Button control).
3. Click the control that was just created. The text displayed in the control becomes editable. Type the caption: "LOOP statement".
4. Press Enter to validate the input.
5. Display the description window of control (press Alt + Enter).
6. In the "General" tab, in the "Button action" area:
 - for the "Type" option, select "Run the browser click code only".
 - for the "During the action" option, select "Do not send anything to the server".
7. Validate the description window of control.
8. Display the events associated with the control (F2).
9. Enter the following code in the "Click (browser)" event:

```
// Moves the image to the left
LOOP
  IMG_Animal..X--
  IF IMG_Animal..X <= 0 THEN BREAK
END
```

This WLanguage code allows you to modify the X coordinate of the Image control (..X property). At each turn, a condition is checked. If this condition is true, the program exits from the loop. In our case, the condition is as follows:

- the position of the Image control is 0,

- the position of the Image control is less than 0 (used to manage the case where the user presses the button several times).




Remark

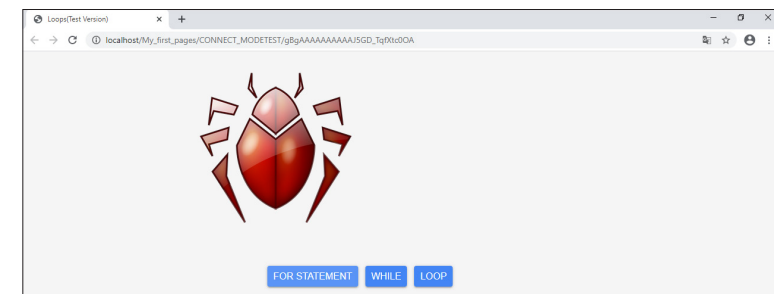
When typing this code, the LOOP statement may be underlined by a green line and a warning may appear in the pane of compilation errors.

This warning reminds you that an exit statement must be found in the loop code to avoid an infinite loop.

10. Close the code window (click X at the top right corner of code editor).

Page test

- Let's now run the page test:
1. Click  among the quick access buttons (or press F9).
 2. The created page is started in execution.
 3. Click the different buttons.



4. Close the test page.

APPENDIX 5. THE PROCEDURES

This lesson will teach you the following concepts...

- Types of procedures.
- Creating and calling a procedure.
- Procedure parameters.
- Procedure result.
- Practical example.

 Estimated time: 30 mn

Overview

A procedure is used to associate an identifier with a code section in order to re-use it. In this lesson, we are going to present the different types of procedures available in WLanguage, their creation mode, how to call them, pass parameters and retrieve a result.

Types of procedures

Three types of procedures are available:

- **Global procedure:** can be used in all project the events and processes of the project (declared in a set of procedures).
- **Local procedure** in a Window, Page or Mobile Window: can be used in all the events and processes that depend on the object in which this procedure was declared.
- **Internal procedure** in a process or event: can only be used in the process or event where it was declared.



Remark

Scope of procedures

The procedures comply with the scope rules presented for the variables (see "The scope of variables").

Creating and calling a procedure

Global procedure

To create a **global procedure**, you must:

1. Create (if necessary) a set of procedures (via the "Project explorer" pane, "Procedures" folder). Give a name to the set of procedures.
2. Create a global procedure in the set of procedures (via the "Project explorer" pane, "Procedures, Set name" folder). Give a name to the procedure.
3. Type the code of global procedure. The procedure code has the following format:

```
PROCEDURE <Name of global procedure>()
```

Local procedure

To create a **local procedure**, you must:

1. Select the element associated with the procedure (window, page, etc.).
2. Create a local procedure (via the "Project explorer" pane, expand the element name, "Local procedures" folder).
3. Give a name to the procedure.
4. Type the code of local procedure. The procedure code has the following format:

```
PROCEDURE <Name of local procedure>()
```

Internal procedure

To create an internal procedure, type the following code in the requested process:

```
INTERNAL PROCEDURE <Procedure name>()
<Code of internal procedure>
END
```

Calling a procedure

To call a procedure, use the procedure name (with the possible parameters that will be passed to it).

```
<Procedure name>(<Parameter 1>, ..., <Parameter N>)
```

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

Procedure parameters

What is a parameter?

A parameter is a value sent to a procedure during the call to the procedure.

The following example is used to call the **Multiply10** procedure and to pass in parameter the value that will be handled in the procedure:

```
Multiply10(50)
```

You have the ability to pass from 0 to several values in parameter to a procedure. These values can have any type (as with the variables).

The parameter is specified in the procedure declaration in the format of a variable. For example, for the **Multiply10** procedure, the procedure code is:

```
PROCEDURE Multiply10(P)
P=P*10
```

P is the parameter expected by the procedure.

**Remark**

To specify the parameter role in the procedure, you have the ability to typecast the parameter in the procedure declaration.

For example, to use numeric values only, you have the ability to declare:

```
PROCEDURE Multiply10(P is numeric)
```

In the following example, the **Multiplication** procedure expects two Integer parameters and returns the multiplication result.

The procedure code is as follows:

```
PROCEDURE Multiplication(Nb1 is int, Nb2 is int)
MyResult is int
MyResult = Nb1 * Nb2
RESULT MyResult
```

The code used to call the procedure is as follows:

```
res is int
Res = Multiplication(10, 50)
// Res is equal to 500
```

How to use the parameters?

By default, **passing parameters in WLanguage is performed by reference** (or by address). The parameter in the procedure represents (references) the variable passed during the call.

Therefore, when a procedure statement modifies the parameter value, the value of the variable corresponding to this parameter is modified.

Example:

- The procedure code is as follows:

```
PROCEDURE Test_address(P1)
P1 = P1 * 2
```

- The code used to call the procedure is as follows:

```
T is int
T = 12 // T is equal to 12 before the call
Test_address(T)
// T is equal to 24 after the call
```

To avoid modifying the value of the variable corresponding to the parameter, the **parameters must be passed by value**. Passing parameters by value allows you to handle a copy of parameter value. If the procedure code modifies the variable value, the value of the variable corresponding to the parameter is not modified.

To force a parameter to be passed by value to a procedure, the LOCAL keyword must be used in front of the parameter name in the procedure declaration. This keyword indicates that the following parameter will not be modified by the procedure.

Example:

- The procedure code is as follows:

```
PROCEDURE Test_value(LOCAL P1)
// Local indicates that the parameter will be passed by value
P1 = P1 * 2
```

- The code used to call the procedure is as follows:

```
T is int
T = 12 // T is equal to 12
Test_value(T)
// T does not change
```

**Remark**

In the same procedure, some parameters can be passed by reference while other parameters can be passed by value. All you have to do is used the LOCAL keyword in front of each parameter passed by value.

Mandatory or optional parameters?

The parameters received in the procedure can be mandatory or optional parameters. A mandatory parameter must be filled during the call to the procedure while an optional parameter can be omitted: in this case, it will take the default value defined when declaring the procedure parameters.



Remark

When declaring a procedure, the optional parameters are the last parameters (they are always specified after all mandatory parameters).

In the following example, the **Multiplication** procedure is using an optional parameter, Nb2. This optional parameter is indicated after the mandatory parameters, by specifying its default value. In this example, the default value of optional parameter is set to 10.

```
PROCEDURE Multiplication(Nb1 is int, Nb2 is int=10)
MyResult is int
MyResult = Nb1 * Nb2
RESULT MyResult
```

The code used to call the procedure is as follows:

```
res is int
res = Multiplication(6)
// Res is equal to 60
```

In this example, the second parameter was not specified. Therefore, its default value will be used.

Procedure result

The procedures can return one or more results. The result can be typecasted. The RESULT keyword must be used to return a value.

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

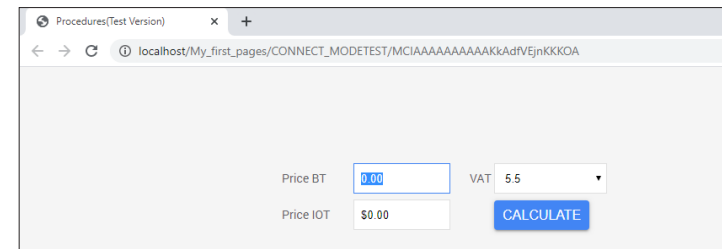
Practical example: Using a procedure

In a new page, we are now going to:

- Create two numeric edit controls containing the value BT and the value IOT.
- Create a Combo Box control used to choose the VAT rate.
- Create a Button control used to calculate and display the value IOT of the amount BT entered.

The calculation result will be displayed in the "Price IOT" control.

This page is as follows:




Project used

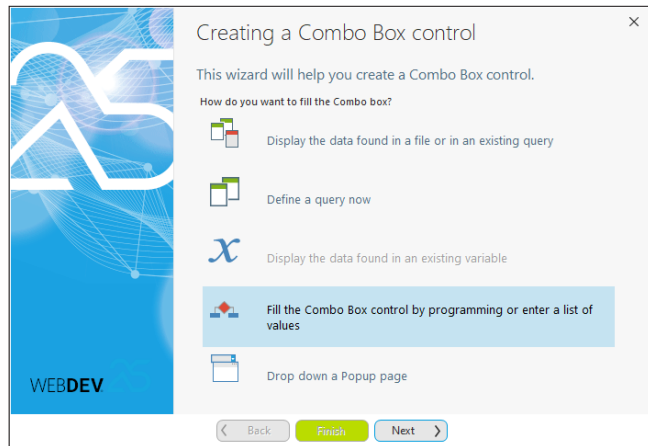
To handle the different concepts of this lesson, we are going to create different pages. These pages will be created in the "My_First_Pages" project.

- Open the "My_First_Pages" project if necessary.
 1. Display the WEBDEV home page (Ctrl + <).
 2. Click "Tutorial", then "My first pages (Exercise)".

Implementation

- Create a new blank page:
 1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Page" then "Page". The page creation wizard appears.
 2. Select "Blank".
 3. Validate. The page is automatically created in the editor. The window for saving an element is displayed.
 4. Specify the page title: "Procedures". The page name ("PAGE_Procedures") is automatically proposed.
 5. Validate the save window ("OK").
- To create the edit control corresponding to the price BT:
 1. On the "Creation" pane, in the "Usual controls" group, expand "Edit" (click the arrow). The list of available edit controls is displayed. Select the "Currency" control and position the control in the page.
 2. Right-click the control and select "Description".
 3. In the description window:
 - Type the control name: "EDT_PriceBT".
 - Type the caption: "Price BT".
 4. Validate.

- ▶ To create the control where the result will be displayed:
 1. On the "Creation" pane, in the "Usual controls" group, expand "Edit" (click the arrow). The list of available edit controls is displayed. Select the "Currency" control and position the control in the page (below the edit control displaying the price BT for example).
 2. Type the control information: right-click the control and select "Description".
 - Specify the control name: "EDT_PriceIOT".
 - Modify the caption to "Price IOT".
 3. The result displayed in this control must not be modifiable. Click the "UI" tab and choose "Read-only" for the initial status.
 4. Validate.
 5. Save the page (Ctrl + S).
- ▶ To create the Combo Box control for VAT selection:
 1. On the "Creation" pane, in the "Usual controls" group, click "Combo box" then click the position where the control will be created in the page (beside the "Price BT" control).
 2. The Combo Box control creation wizard for is displayed. This wizard is used to define the main control characteristics.
 3. Select the option "Fill the Combo Box control by programming or enter a list of values".

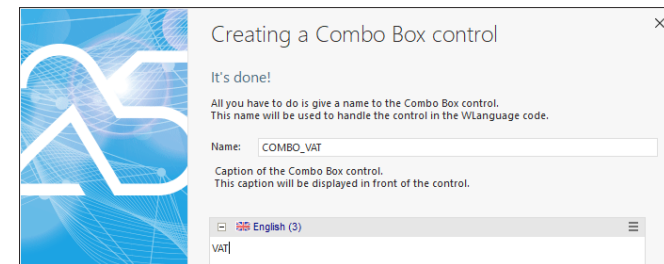


4. Go to the next step of the wizard.
5. Keep the default options. Go to the next step of the wizard.

6. In the "Initial content" step, type the list of possible VAT values:
 - 5.5
 - Press the Enter key.
 - 10
 - Press the Enter key.
 - 20.



7. Display the next wizard step: give a name (COMBO_VAT) and a caption (VAT) to the control.




8. Validate.
 9. The Combo Box control is created in the page.
- ▶ Resize the control if necessary:
 1. Select the control.
 2. Reduce the control width via the sizing handles.
 - ▶ To create the "Calculate" Button control:
 1. On the "Creation" pane, in the "Usual controls" group, click **OK**.
 2. Click the location where the control will be created (below the Combo Box control for example).
 3. Click the control that was just created. The text displayed in the button becomes editable. Type the caption: "Calculate".
 4. Press Enter to validate the input.

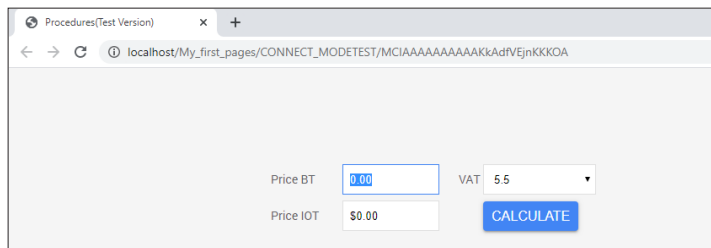
► The amount IOT will be calculated via the code of the "Calculate" Button control.

1. Display the code of "Calculate" control ("Code" from the popup menu).
2. Write the following code in the "Click on BTN_Calculate (server code)" event:

```
SWITCH COMBO_VAT..DisplayedValue
// 5.5 %
CASE 5.5
    EDT_PriceIOT = EDT_PriceBT * 1.055
// 10 %
CASE 10
    EDT_PriceIOT = EDT_PriceBT * 1.1
// 20 %
CASE 20
    EDT_PriceIOT = EDT_PriceBT * 1.2
END
```

This code calculates the amount IOT by using the value selected in the Combo Box control (returned by `..DisplayedValue`).

3. Run the page test ( among the quick access buttons). Give a price BT. Select the different values in the Combo Box control and click "CALCULATE".



Our page operates properly now. **However, instead of using a formula 3 times to calculate the price, a procedure can be used to perform this calculation.**

- Close the test page to go back to the code editor.

Creating and using a procedure

- To create the procedure for calculating the amount IOT:
 1. Click the page background.
 2. Display the events associated with the page (F2).
 3. In the code editor, on the "Code" pane, in the "Procedures" group, expand "New" and select "New local procedure (Server)".
 4. In the window that is displayed, type the name of local procedure ("Calc_IOT") and validate ("Add" button).
 5. The new procedure local to the page is automatically created and its code is displayed in the code editor.

6. Write the following code:

```
PROCEDURE Calc_IOT(PriceBT, VATRate)
cyIOT is currency
cyIOT = PriceBT * (1 + VATRate/100)
RESULT cyIOT
```

Let's study this code:

- The Calc_IOT procedure expects 2 parameters: the price before tax and the VAT rate.
- This procedure declares a currency variable. This variable is used to store the calculated price IOT.
- This procedure calculates the price IOT and returns the calculated value.

7. Close the code editor.



Remark


When creating a procedure, comments are automatically generated BEFORE the procedure code. These comments are mainly used to specify the content of parameters and return value.

It is important to fill these comments. Indeed, they will be automatically displayed in a tooltip when typing the call to the procedure in the code editor.

- All you have to do is call the procedure from the Button control.
 1. Select the "Calculate" Button control.
 2. Display the events associated with the control (press F2, for example).
 3. Replace the existing WLanguage code with the following code:

```
SWITCH COMBO_VAT..DisplayedValue
// 5.5 %
CASE 5.5
    EDT_PriceIOT = Calc_IOT(EDT_PriceBT, 5.5)
// 10 %
CASE 10
    EDT_PriceIOT = Calc_IOT(EDT_PriceBT, 10)
// 20 %
CASE 20
    EDT_PriceIOT = Calc_IOT(EDT_PriceBT, 20)
END
```

This WLanguage code calls the Calc_IOT procedure to calculate the amount IOT. Two parameters are passed in parameter: the price BT and the VAT rate. The result returned by the procedure is assigned to the EDT_Price_IOT control.

- Let's now run the page test:
 1. Run the page test ( among the quick access buttons). Give a price BT. Select the different values in the Combo Box control and click the "Calculate" button.
 2. The amount BT is displayed.
 3. Close the test page.

Conclusion

This section allowed you to get familiar with the main concepts of WLanguage programming in WEBDEV. Several other features are available. Some of them will be presented in this tutorial.

WLanguage is a very powerful language that allows you to develop applications that use:

- the Object-Oriented Programming (OOP),
- the MVP (Model **View** Presenter),
- a 3-tier programming,
- advanced types (XML, Email, ...), ...

See the online help regarding the relevant topic for more details.

APPENDIX 6. QUESTIONS/ANSWERS

This lesson will teach you the following concepts...

- Questions/Answers about the code editor.



Estimated time: 10 mn

Questions/Answers

How to view the element to which the current event belongs?

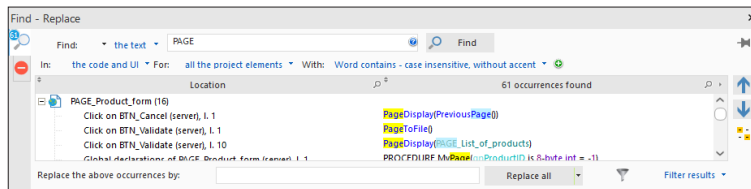
To view the element corresponding to the current event, go to "Code" pane, "Navigation" group, click "Go to element" (Ctrl + Alt + F2). The page containing the requested element is displayed.

How to print the source code?

To print the current source code, click the  icon in the "Home" pane or press Ctrl + P.

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) :



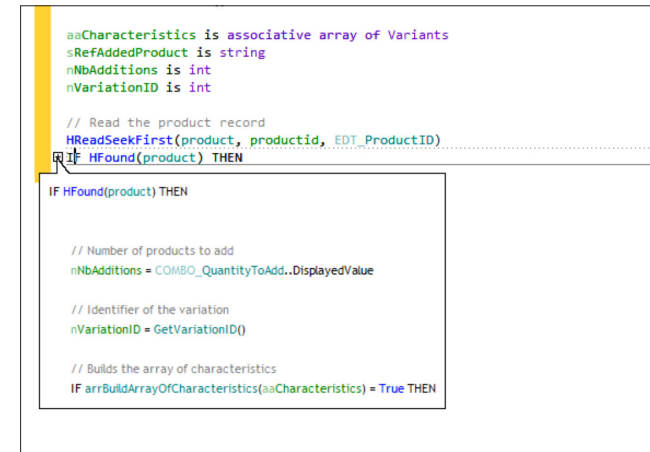
Then, you have the ability to define the different characteristics of the search to perform.

What is the meaning of "+" and "-" signs in the code editor?

The code editor allows you to expand or collapse the WLanguage code. This feature is very useful if your processes use a lot of structured statements (loops, conditions, browses, ...).

To collapse a code section, on the "Display" pane, expand "Collapse" and select "Collapse all" (or press Ctrl + Shift + * (on the numeric keypad)).

Only the comments remain visible. The associated code is displayed in a tooltip when the comment line is hovered by the mouse cursor:



Press Ctrl + * (on the numeric keypad) to expand the entire code. A click performed on the "-" or "+" symbol allows you to collapse or expand the corresponding code section.

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.

Is it possible to find out the code line number?

In the code editor, to enable (or not) the numbering of code lines, on the "Display" pane, in the "Help for edit" group, click "Display the line numbers" (shortcut: Ctrl + Shift + G).

Is there a method to easily display the syntax or the help about a function?

When typing a function in the code editor:

- the name of each parameter is displayed in a tooltip for the code line currently in edit. For the parameter currently in edit, hovering the parameter name with the mouse is used to display the parameter details in a tooltip.
- If several syntaxes are available, it is possible to switch from one syntax to another using Ctrl + up arrow or down arrow.
- the function syntax is displayed in the status bar of editor.

In the help, the parameters enclosed between [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 function use and it automatically generates the corresponding code.

A help page is associated with each WLanguage function and property. This help page can be directly accessed from the editor by pressing F1 on the name of requested function or property.

What are the useful shortcuts in the code editor?

- To display the events of a control, method, class, procedure or report block, press F2 while on the name of the desired element.
- Ctrl + F2 is used to go back to the initial process or event.
To move from one process/event to another, press F2 repeatedly. To go back to the initial process or event, press Ctrl + F2 the same number of times.
- Ctrl + L deletes the current line.
- Ctrl + D duplicates the current line (or the selected lines) on the line below.
- Tab (tab key) and Shift + Tab are used to manage the indent for the selected lines.
- Ctrl + / converts the selected lines into comments, Ctrl + Shift + / removes the comments (Caution: / key on the numeric keypad).
- Ctrl + R is used to automatically re-indent the displayed code.

CONCLUSION

The tutorial is over now!

This course covered a range of topics, but not all WEBDEV features, 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 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 25!**