



DEVELOP 10 TIMES FASTER



P4SOFT

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

Email address of Free Technical Support: [freetechnicalsupport@windev.com](mailto:freetechnicalsupport@windev.com)

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

All product names or other trademarks mentioned in this publication are registered trademarks of their respective owners.  
© PC SOFT 2014: This publication may not be reproduced in part or in full without the express consent of PC SOFT.

# TABLE OF CONTENTS

## Introduction

|  |    |
|--|----|
| Preliminary points .....                 | 17 |
| Overview of the tutorial .....           | 17 |
| How do I access the online help? .....   | 18 |
| Legend of symbols .....                  | 19 |
| If you are familiar with WebDev 18... .. | 20 |
| What is WebDev used for? .....           | 20 |

## 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 |

### Lesson 1.2. My first page

|  |    |
|--|----|
| Overview .....                                     | 29 |
| Opening the project .....                          | 29 |
| My first page: entering information .....          | 30 |
| Overview .....                                     | 30 |
| Creating the first page.....                       | 30 |
| Inputing and displaying the entered value .....    | 33 |
| Test of a page .....                               | 37 |
| Checks performed on the "First name" control ..... | 38 |

### Lesson 1.3. My first pages

|   |    |
|---|----|
| Overview .....                                | 41 |
| Performing a calculation in server code ..... | 41 |
| Overview .....                                | 41 |
| Implementation.....                           | 41 |
| Displaying the time in real time .....        | 44 |
| Overview .....                                | 44 |
| Adding the Time control .....                 | 44 |
| Implementing the timer .....                  | 45 |
| List the files found in a directory .....     | 46 |
| Overview .....                                | 46 |
| Creating the page .....                       | 47 |
| Creating the Looper control .....             | 47 |
| Programming.....                              | 49 |
| Programming with callback.....                | 49 |
| Anchoring the page and the controls .....     | 51 |

## Lesson 1.4. Concepts

|  |           |
|--|-----------|
| <b>Main concepts and terminology .....</b>                   | <b>54</b> |
| Main concepts .....  | 54        |
| Terminology.....   | 55        |
| <b>The different types of pages proposed by WebDev .....</b> | <b>56</b> |
| The different types of pages .....                           | 56        |
| The different types of sites .....                           | 59        |
| <b>A Web server and browsers ... ..</b>                      | <b>59</b> |
| The different browsers .....                                 | 59        |
| The different Web servers.....                               | 60        |
| What's happening on the browser side ... ..                  | 60        |
| What's happening on the server side ... ..                   | 60        |

## Lesson 1.5. Programming concepts

|  |           |
|--|-----------|
| <b>Introduction .....</b>  | <b>63</b> |
| <b>Server code and browser code .....</b>                            | <b>63</b> |
| Server code .....  | 64        |
| Browser code .....   | 64        |
| <b>Declaring the different types of variables .....</b>              | <b>65</b> |
| Declaring the variables and their scope .....                        | 65        |
| <b>Main statements of WLanguage .....</b>                            | <b>66</b> |
| Conditional statements (server code and browser code).....           | 67        |
| Loop statements (server code and browser code).....                  | 67        |
| The comments .....   | 68        |
| The MyPage keyword (server and browser code).....                    | 69        |
| Practical example on the different main statements .....             | 69        |
| <b>Main WLanguage operators .....</b>                                | <b>69</b> |
| Logical operators .....  | 70        |
| Comparison operators.....  | 70        |
| Indirection operators .....  | 70        |
| <b>Procedures and functions .....</b>                                | <b>71</b> |
| Definition .....   | 71        |
| Local procedure .....  | 71        |
| Global procedure and set of procedures .....                         | 71        |
| How do I choose whether a procedure must be global or local? .....   | 72        |
| About passing parameters.....  | 72        |
| Calling a procedure.....   | 72        |
| Creating a procedure.....  | 72        |
| A practical example about the procedures and functions.....          | 73        |
| <b>Processing strings (server and browser code) .....</b>            | <b>74</b> |
| Practical example .....  | 74        |
| Details (server and browser code).....                               | 74        |
| <b>Processing the numeric values (server and browser code) .....</b> | <b>76</b> |
| Practical example .....  | 76        |
| Details.....   | 76        |
| <b>Processing the currencies (server code only) .....</b>            | <b>78</b> |
| Practical example .....  | 78        |
| Details.....   | 78        |

|  |           |
|--|-----------|
| Processing the dates and times (server and browser code) .....     | 79        |
| <b>The dates (server code and browser code) .....</b>              | <b>80</b> |
| Practical example .....  | 80        |
| Input mask and returned value .....                                | 80        |
| What is today's date? .....  | 81        |
| What is today's date, with the day and the month in letters? ..... | 82        |
| How many days between 2 dates? .....                               | 82        |
| What is the day of the week corresponding to a given date? .....   | 83        |
| Practical exercise .....   | 83        |
| Handling Date variables (server code) .....                        | 84        |
| <b>The times (server code and browser code) .....</b>              | <b>85</b> |
| Practical example .....  | 85        |
| What time is it? .....   | 85        |
| How much time has passed between two given times? .....            | 86        |
| <b>Calculations with dates and times .....</b>                     | <b>87</b> |

## **PART 2 - SITE WITH DATA**

---

### **Lesson 2.1. Overview**

|  |           |
|--|-----------|
| <b>Overview of the site created in this part .....</b> | <b>92</b> |
|--|-----------|

### **Lesson 2.2. Project and analysis**

|  |            |
|--|------------|
| <b>Overview .....</b>  | <b>94</b>  |
| <b>Creating the project .....</b>                              | <b>94</b>  |
| <b>Creating the analysis .....</b>                             | <b>96</b>  |
| <b>Creating the description of the data files .....</b>        | <b>97</b>  |
| Creating a data file and its items in the editor .....         | 98         |
| Importing a CSV file .....                                     | 102        |
| Importing files from the dictionary .....                      | 105        |
| Direct import of existing data files .....                     | 106        |
| <b>Creating the links .....</b>                                | <b>107</b> |
| <b>Generation of the analysis .....</b>                        | <b>109</b> |
| <b>Configuring the analysis for RAD .....</b>                  | <b>110</b> |
| Configuring the RAD in the description of the data files ..... | 110        |
| Configuring the RAD in the description of items .....          | 111        |

### **Lesson 2.3. The full RAD**

|   |            |
|---|------------|
| <b>What is the RAD? .....</b>               | <b>114</b> |
| <b>Generating RAD .....</b>                 | <b>114</b> |
| <b>Running the test of the site .....</b>   | <b>116</b> |
| <b>Customizing the generated site .....</b> | <b>117</b> |

## Lesson 2.4. Edit page

---

|   |            |
|---|------------|
| <b>Overview</b> .....                                 | <b>119</b> |
| <b>Creating an edit page</b> .....                    | <b>119</b> |
| <b>Managing the addition of a record</b> .....        | <b>122</b> |
| Creating the addition button.....                     | 122        |
| Adding an input mask .....                            | 123        |
| Erasing data after the addition.....                  | 124        |
| Viewing the records .....                             | 124        |
| Managing the upload of the book cover .....           | 125        |
| Closing button.....                                   | 126        |
| <b>Managing the search and the modification</b> ..... | <b>127</b> |
| Exact-match search .....                              | 127        |
| Generic search.....                                   | 128        |
| Modifying the form displayed .....                    | 129        |
| Browsing the forms.....                               | 130        |

## Lesson 2.5. Multicriteria search

---

|  |            |
|--|------------|
| <b>Overview</b> .....  | <b>133</b> |
| <b>Creating a query with parameters</b> .....                        | <b>134</b> |
| <b>Creating a page with a Table control based on the query</b> ..... | <b>136</b> |

## Lesson 2.6. Printing the data

---

|  |            |
|--|------------|
| <b>Overview</b> .....                  | <b>139</b> |
| <b>Creating a report</b> .....         | <b>139</b> |
| <b>Starting the report print</b> ..... | <b>140</b> |

## Lesson 2.7. The user groupware

---

|  |            |
|--|------------|
| <b>Overview</b> .....                          | <b>143</b> |
| <b>Integrating the user groupware</b> .....    | <b>143</b> |
| <b>Configuring the user groupware</b> .....    | <b>146</b> |
| Creating the users and the groups.....         | 146        |
| Defining the rights .....                      | 147        |
| Running the test of the site .....             | 148        |
| Disable the management of user groupware ..... | 148        |

## Lesson 2.8. The dashboard

---

|                              |            |
|------------------------------|------------|
| <b>Overview</b> .....        | <b>150</b> |
| <b>Automatic tests</b> ..... | <b>151</b> |

## PART 3 - PAGES AND CONTROLS

---

### Lesson 3.1. The pages

---

|   |     |
|---|-----|
| How do I create a page? .....                     | 158 |
| Description of a page: 7 tabs are available ..... | 159 |
| Practical example .....                           | 159 |

### Lesson 3.2. The controls

---

|                                       |            |
|---------------------------------------|------------|
| <b>Introduction</b> .....             | <b>164</b> |
| Practical example .....               | 164        |
| <b>The standard controls</b> .....    | <b>164</b> |
| Type of control: Static.....          | 165        |
| Type of control: Edit control .....   | 166        |
| Type of control: Button .....         | 170        |
| Type of control: Link .....           | 172        |
| Type of control: Image.....           | 172        |
| Type of control: Clickable image..... | 176        |
| Type of control: Radio button.....    | 178        |
| Type of control: Check box.....       | 179        |
| Type of control: List box.....        | 180        |
| Type of control: Combo box.....       | 185        |
| Type of control: Table .....          | 186        |
| Type of control: Looper.....          | 190        |
| Type of control: TreeView .....       | 192        |
| Type of control: Cell .....           | 192        |
| Type of control: Chart .....          | 193        |
| Type of control: Rating.....          | 193        |
| <b>Specific controls</b> .....        | <b>194</b> |
| Type of control: Calendar .....       | 195        |
| Type of control: Scheduler .....      | 195        |
| Type of control: Organizer .....      | 195        |
| Type of control: TreeView Table.....  | 196        |
| Type of control: Captcha .....        | 196        |
| Type of control: Bar Code.....        | 196        |
| Type of control: HTML table .....     | 197        |
| Type of control: Slider.....          | 197        |
| Type of control: Horizontal rule..... | 197        |
| Type of control: HTML.....            | 198        |
| Type of control: Web Camera.....      | 198        |
| Type of control: Java Applet.....     | 198        |
| Type of control: Flash and Flex ..... | 199        |
| Type of control: Silverlight.....     | 201        |
| Type of control: IFrame.....          | 201        |
| Type of control: Site map path.....   | 202        |
| Type of control: Site map .....       | 202        |
| Type of control: Pager.....           | 202        |
| Type of control: Thumbnail.....       | 203        |

### Lesson 3.3. Ergonomics of the site

---

|  |            |
|--|------------|
| Overview .....   | 205        |
| Handling the styles of controls .....                      | 205        |
| Defining the tab order of controls .....                   | 206        |
| Displaying a help in the pages .....                       | 207        |
| The tooltip .....  | 207        |
| Help message .....   | 207        |
| Indication text .....                                      | 207        |
| <b>Create your pages by using the zoning mode .....</b>    | <b>208</b> |
| Overview .....   | 208        |
| How to use the zoning? .....                               | 209        |
| Practical example .....                                    | 210        |
| <b>Anchor the controls .....</b>                           | <b>211</b> |
| <b>Use the rulers and the alignment options .....</b>      | <b>212</b> |
| <b>Use the tabs to group your controls .....</b>           | <b>213</b> |
| <b>Improve the navigation in your site .....</b>           | <b>213</b> |
| <b>Use modern dialog boxes .....</b>                       | <b>214</b> |
| <b>Interface of the site: Use the page templates .....</b> | <b>215</b> |
| What is a page template? .....                             | 215        |
| How do I create a page template? .....                     | 216        |
| How do I apply a page template? .....                      | 216        |
| Updating the template .....                                | 217        |
| How do I dissociate a template imported into a page? ..... | 217        |

### Lesson 3.4. Reusability

---

|   |            |
|---|------------|
| <b>What is the reusability? .....</b>   | <b>219</b> |
| <b>Re-using a set of controls .....</b> | <b>219</b> |
| Practical example .....                 | 219        |
| The supercontrol .....                  | 219        |
| Internal page .....                     | 220        |
| Control template .....                  | 221        |

### Lesson 3.5. Questions/Answers

---

|   |     |
|---|-----|
| How do I display a page in a new window of the browser? .....                         | 224 |
| By programming .....  | 224 |
| By using the page editor .....  | 224 |
| How do I modify the color of a static by programming? .....                           | 225 |
| How do I make a button invisible? .....   | 226 |
| How do I create a vertical menu in a WebDev page? .....                               | 226 |
| How do I add a popup menu to a table or to a looper? .....                            | 227 |
| How do I pass parameters to a page? .....   | 227 |
| How do I group the controls in order to modify their properties by programming? ..... | 228 |
| How do I transform a check box into a radio button? .....                             | 228 |
| How do I add an image to the page background? .....                                   | 228 |
| How do I display the date of site update? .....                                       | 229 |

## PART 4 - DATABASES AND ANALYSES

---

### Lesson 4.1. Introduction

---

|   |     |
|---|-----|
| Overview .....  | 234 |
| The different modes for accessing the databases ..... | 235 |
| Native Access.....                                    | 235 |
| Direct ODBC access.....                               | 235 |
| OLE DB access.....                                    | 235 |
| ODBC access via OLE DB .....                          | 236 |

### Lesson 4.2. Browsing the data files and the queries

---

|  |     |
|--|-----|
| HReadFirst browse .....                    | 238 |
| HReadSeek browse .....                     | 238 |
| FOR EACH browse .....                      | 239 |
| FOR EACH WITH browse .....                 | 239 |
| What type of browse should I choose? ..... | 240 |

### Lesson 4.3. Managing the errors

---

|   |     |
|---|-----|
| Overview .....  | 242 |
| Practical example .....                               | 242 |
| Managing the errors in automatic mode .....           | 242 |
| Managing the errors in advanced programmed mode ..... | 244 |

### Lesson 4.4. Data encryption

---

|   |     |
|---|-----|
| What is data encryption used for? .....             | 247 |
| How do I encrypt the data files? .....              | 248 |
| Implementing the encryption of the data files ..... | 248 |
| Managing the encryption in WLanguage .....          | 250 |

### Lesson 4.5. Advanced concepts

---

|   |     |
|---|-----|
| Overview .....  | 252 |
| The log process .....   | 252 |
| What is the log process used for? .....                                       | 252 |
| The transactions .....  | 252 |
| A simple example of the need for transactions in some types of processes..... | 252 |
| The transactions on HFSQL .....   | 253 |
| The replication .....   | 254 |
| The triggers .....  | 254 |

## PART 5 - MANAGING A HFSQL CLIENT/SERVER DATABASE

---

### Lesson 5.1. Introduction

---

|  |     |
|--|-----|
| Overview .....                                       | 258 |
| Why switch a site to HFSQL Client/Server mode? ..... | 259 |

### Lesson 5.2. Implementing a HFSQL Client/Server site

---

|  |     |
|--|-----|
| Overview .....   | 261 |
| Installing a local HFSQL server .....                          | 261 |
| Creating a site in HFSQL Client/Server mode .....              | 261 |
| Migrate a HFSQL Classic site to HFSQL Client/Server mode ..... | 262 |
| Overview .....   | 262 |
| Migrating the example.....                                     | 262 |
| Features available in HFSQL Client/Server mode .....           | 264 |

### Lesson 5.3. Managing a Client/Server database

---

|   |     |
|---|-----|
| Overview .....  | 266 |
| Configuring the computers .....                           | 266 |
| The HFSQL Control Center .....                            | 266 |
| Creating a user account in the HFSQL Control Center ..... | 268 |
| Saving the database .....                                 | 271 |
| Conclusion .....  | 271 |

## PART 6 - THE QUERIES

---

### Lesson 6.1. Create your first query

---

|                        |     |
|------------------------|-----|
| Introduction .....     | 276 |
| Your first query ..... | 276 |

### Lesson 6.2. Queries with count

---

|                                   |     |
|-----------------------------------|-----|
| Creating a query with count ..... | 283 |
|-----------------------------------|-----|

### Lesson 6.3. Sum queries

---

|                              |     |
|------------------------------|-----|
| Creating a "Sum" query ..... | 286 |
|------------------------------|-----|

### Lesson 6.4. The SQL language

---

|                              |     |
|------------------------------|-----|
| Source code of a query ..... | 290 |
| The SQL commands .....       | 291 |
| The SELECT statement.....    | 291 |
| The FROM statement .....     | 291 |
| The WHERE clause .....       | 291 |
| The GROUP BY clause .....    | 292 |
| The ORDER BY clause .....    | 292 |

**Lesson 6.5. Using queries**

|  |     |
|--|-----|
| Using a query from a report .....                  | 294 |
| Using a query from a table or from a loopier ..... | 294 |
| Using a query from WLanguage .....                 | 294 |

**PART 7 - THE REPORTS****Lesson 7.1. Introduction**

|  |            |
|--|------------|
| <b>Principles for printing on Internet</b> ..... | <b>298</b> |
| Direct print .....                               | 298        |
| Generation of documents (HTML, PDF, XML), ... .. | 298        |
| <b>Basic vocabulary</b> .....                    | <b>299</b> |

**Lesson 7.2. Your first report**

|   |            |
|---|------------|
| <b>Overview</b> .....                                       | <b>302</b> |
| The reports to create .....                                 | 302        |
| <b>Creating your first report</b> .....                     | <b>303</b> |
| Creating the report: List of customers sorted by city ..... | 303        |

**Lesson 7.3. Statistical reports**

|                           |            |
|---------------------------|------------|
| <b>Overview</b> .....     | <b>312</b> |
| <b>Step by step</b> ..... | <b>313</b> |

**Lesson 7.4. Crosstab reports**

|                           |            |
|---------------------------|------------|
| <b>Overview</b> .....     | <b>319</b> |
| <b>Step by step</b> ..... | <b>320</b> |

**Lesson 7.5. Running reports**

|   |            |
|---|------------|
| <b>Running reports in WLanguage</b> ..... | <b>324</b> |
| iDestination function .....               | 325        |
| iPrintReport function .....               | 325        |

**PART 8 - ADVANCED PROGRAMMING****Lesson 8.1. WEB features**

|   |            |
|---|------------|
| <b>Overview</b> .....   | <b>330</b> |
| Practical example .....   | 330        |
| <b>The cookies</b> .....  | <b>330</b> |
| What is a cookie made of?.....  | 331        |
| Managing the cookies with WebDev .....                                    | 331        |
| Practical example .....   | 332        |
| <b>Uploading files</b> .....  | <b>333</b> |
| Managing the upload with a single-file or multi-file upload control ..... | 333        |
| Practical example .....   | 334        |

|  |            |
|--|------------|
| <b>The file download</b> .....                                 | <b>334</b> |
| Managing the download.....                                     | 335        |
| Practical example .....  | 335        |
| <b>Securing the pages</b> .....                                | <b>336</b> |
| Protecting the access to your pages by a password .....        | 336        |
| Practical example .....  | 336        |
| <b>Securing the data and pages via TSL/SSL</b> .....           | <b>338</b> |
| Implementing secure transactions via the TLS/SSL protocol..... | 338        |
| Transactions secured by TLS\SSL in a WebDev site .....         | 338        |
| <b>Secure payment</b> .....                                    | <b>338</b> |
| System for secure payment in a WebDev site.....                | 340        |
| <b>Ajax</b> .....  | <b>341</b> |
| What are benefits of Ajax?.....                                | 341        |
| How to program AJAX in WEBDEV? .....                           | 341        |
| Practical example .....  | 342        |

## Lesson 8.2. Automatic management of errors

---

|   |            |
|---|------------|
| <b>Overview</b> .....                                 | <b>345</b> |
| Operating mode.....                                   | 345        |
| Implementation.....                                   | 345        |
| Types of errors .....                                 | 345        |
| <b>Using the automatic management of errors</b> ..... | <b>346</b> |

## Lesson 8.3. Managing the emails

---

|   |            |
|---|------------|
| <b>Overview</b> .....                               | <b>348</b> |
| <b>The POP3 and SMTP protocols: Principle</b> ..... | <b>348</b> |
| Practical example .....                             | 349        |
| <b>Sending an email: the different steps</b> .....  | <b>349</b> |
| Starting an SMTP messaging session.....             | 349        |
| Preparing the email.....                            | 350        |
| Sending the email.....                              | 350        |
| Disconnecting .....                                 | 350        |
| <b>Other possibilities</b> .....                    | <b>350</b> |

## Lesson 8.4. OOP

---

|  |            |
|--|------------|
| <b>Concepts</b> .....                    | <b>352</b> |
| The classes .....                        | 352        |
| The objects.....                         | 352        |
| The members.....                         | 352        |
| The methods.....                         | 352        |
| Concept of inheritance.....              | 352        |
| Constructor and Destructor .....         | 352        |
| Data encapsulation .....                 | 353        |
| Creating an object-oriented program..... | 353        |
| Declaring a class .....                  | 353        |
| Describing the methods.....              | 354        |
| Declaring and handling the objects ..... | 354        |

## PART 9 - ADVANCED PROJECT MANAGEMENT

---

### Lesson 9.1. Dashboard

---

|  |            |
|--|------------|
| <b>Overview</b> .....                                  | <b>358</b> |
| Example .....  | 358        |
| <b>The different dashboard elements</b> .....          | <b>358</b> |
| "Project" Widget .....                                 | 359        |
| "SCM" Widget .....                                     | 359        |
| "Static audit and compilation" Widget .....            | 359        |
| "Runtime" Widget .....                                 | 359        |
| "Performances" Widget .....                            | 359        |
| "Fast selection" Widget .....                          | 360        |
| The Widgets in relation with the Control Centers ..... | 360        |

### Lesson 9.2. SCM

---

|  |            |
|--|------------|
| <b>Introduction</b> .....                      | <b>362</b> |
| <b>SCM (Source Code Manager)</b> .....         | <b>362</b> |
| Principle of SCM .....                         | 362        |
| Creating the SCM database .....                | 363        |
| Including a project in the SCM .....           | 364        |
| Opening a project from the SCM .....           | 367        |
| Configuring the SCM .....                      | 367        |
| Checking out an element .....                  | 369        |
| Modifying the checked-out element .....        | 370        |
| Checking the checked-out element back in ..... | 370        |
| Synchronizing the project .....                | 371        |
| Off-line mode (or mobile mode) .....           | 372        |
| SCM administrator .....                        | 372        |
| Disconnecting from the SCM .....               | 372        |

### Lesson 9.3. Control Centers

---

|  |            |
|--|------------|
| <b>Overview</b> .....                      | <b>374</b> |
| <b>The Project Monitoring Center</b> ..... | <b>374</b> |
| <b>Managing your time</b> .....            | <b>376</b> |

### Lesson 9.4. Business rules

---

|  |            |
|--|------------|
| <b>Overview</b> .....                      | <b>379</b> |
| Example .....                              | 379        |
| <b>Application on a real example</b> ..... | <b>379</b> |
| Creating a business rule .....             | 379        |

## Lesson 9.5. Internal components

---

|  |            |
|--|------------|
| <b>Overview</b> .....  | <b>383</b> |
| Teamwork.....  | 383        |
| The databases accessed by several projects.....              | 383        |
| The processes used in several projects.....                  | 383        |
| The ability to distribute a feature or set of features ..... | 384        |
| Multi-product component.....                                 | 384        |
| <b>Step by step</b> .....                                    | <b>384</b> |
| Step 1: Creating an internal component.....                  | 384        |
| Step 2: Using the internal component .....                   | 388        |

## Lesson 9.6. Multi-configuration

---

|  |            |
|--|------------|
| <b>Overview</b> .....                            | <b>392</b> |
| <b>Creating a project configuration</b> .....    | <b>392</b> |
| <b>Managing the project configurations</b> ..... | <b>394</b> |
| <b>Multiple generation</b> .....                 | <b>394</b> |

## Lesson 9.7. The debugger

---

|   |            |
|---|------------|
| <b>Overview</b> .....                     | <b>396</b> |
| <b>Debugging a page</b> .....             | <b>396</b> |
| Starting the debugger.....                | 396        |
| <b>Trace window of the debugger</b> ..... | <b>398</b> |
| <b>Debugging a project</b> .....          | <b>399</b> |

## Lesson 9.8. The performance profiler

---

|   |            |
|---|------------|
| <b>Overview</b> .....   | <b>401</b> |
| <b>Using the performance profiler</b> .....                     | <b>401</b> |
| <b>Optimizing a process with the performance profiler</b> ..... | <b>402</b> |
| Reading the result of the performance profiler.....             | 402        |
| Choosing a process to optimize.....                             | 402        |
| Optimizing a process .....                                      | 403        |

## Lesson 9.9. Import/Export

---

|   |            |
|---|------------|
| <b>Importing elements</b> .....         | <b>406</b> |
| <b>Exporting elements</b> .....         | <b>407</b> |
| <b>Specific import operations</b> ..... | <b>408</b> |
| Importing a WinDev project .....        | 408        |
| Importing an HTML page.....             | 408        |

## Lesson 9.10. UML and 3-tier

---

|   |            |
|---|------------|
| <b>What is the UML?</b> .....               | <b>411</b> |
| <b>The diagrams managed by WebDev</b> ..... | <b>411</b> |
| Class diagram .....                         | 411        |
| Use case diagram .....                      | 411        |
| Object diagram.....                         | 411        |
| Component diagram.....                      | 411        |
| Activity diagram.....                       | 411        |
| Sequence diagram .....                      | 411        |
| Collaboration diagram.....                  | 411        |
| State-transition diagram .....              | 412        |
| Deployment diagram .....                    | 412        |
| <b>3-tier</b> .....                         | <b>412</b> |
| What the 3-tier is? .....                   | 412        |
| How do I implement the 3-tier? .....        | 413        |
| Handling the different layers .....         | 413        |

## Lesson 9.11. Flexible modeling

---

|                         |            |
|-------------------------|------------|
| <b>Principle</b> .....  | <b>415</b> |
| <b>Operations</b> ..... | <b>415</b> |

## Lesson 9.12. Multilingual

---

|  |            |
|--|------------|
| <b>What is a multilingual site?</b> .....                            | <b>419</b> |
| <b>Choosing the project languages</b> .....                          | <b>419</b> |
| Opening the example .....  | 419        |
| Configuring the project.....   | 419        |
| <b>Localizing the analysis</b> .....                                 | <b>421</b> |
| <b>Localizing the project elements</b> .....                         | <b>423</b> |
| How do I translate this information? .....                           | 424        |
| Direct input of translations .....                                   | 424        |
| Special cases .....  | 425        |
| Translation with WDMSG and WDTRAD.....                               | 426        |
| <b>Programming the change of language</b> .....                      | <b>427</b> |
| <b>Managing the Unicode format and specific character sets</b> ..... | <b>427</b> |

## PART 10 - DEPLOYING SITES

---

### Lesson 10.1. Deploying a dynamic site

---

|   |            |
|---|------------|
| <b>Overview</b> .....   | <b>432</b> |
| <b>Required configuration</b> .....                               | <b>432</b> |
| Installing the "WebDev Application Server - 10 connections" ..... | 433        |
| Configuring the WebDev account for deployment.....                | 433        |
| Creating a Windows user account.....                              | 434        |
| <b>Deploying your dynamic site</b> .....                          | <b>435</b> |
| Preparing the setup .....   | 435        |

---

## Lesson 10.2. Remote management

---

|                          |     |
|--------------------------|-----|
| Overview .....           | 441 |
| Using WAdminWeb190 ..... | 442 |

---

## Lesson 10.3. Site statistics

---

|                            |     |
|----------------------------|-----|
| Introduction .....         | 444 |
| Server configuration ..... | 444 |
| Using WDStatistic .....    | 445 |
| Starting WDStatistic ..... | 445 |
| Setting .....              | 445 |

---

## Lesson 10.4. WinDev Back Office

---

|                |     |
|----------------|-----|
| Overview ..... | 447 |
|----------------|-----|

---

## Conclusion

---

# INTRODUCTION

## Preliminary points

---

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

This manual does not cover all the features of WebDev.

You should plan on spending two days to follow this tutorial and to learn WebDev: you'll find it well worth it!

If you try to develop a WebDev site before following this tutorial, you will loose time, and a lot more than two days.

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

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

WebDev evolves all the time, therefore the screen shots found in this guide may differ from the ones found in your product.

## Overview of the tutorial

---

The 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 Programming Guide or the online help (accessible from the editors or from the guide).

We remind you that most of the information found in this tutorial can be accessed from the online help.

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!**



Tip

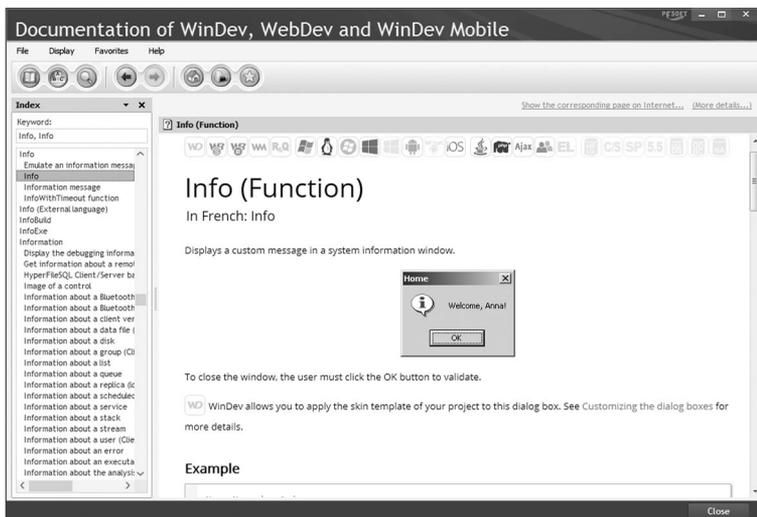
The tutorial may have evolved since this document was published. Don't hesitate to check the online version of this tutorial (PDF file accessible from the WebDev menu: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Tutorial (PDF)".

## How do I access the online help?

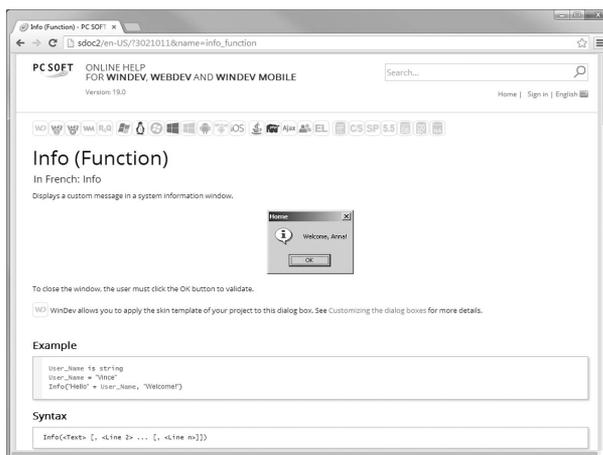
The online help of WebDev enables you to get detailed information about the 2500 WLanguage functions. The online help also contains the help about the editors and the controls, tips, ...

The online help is available at any time in WebDev:

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



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





Notes

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

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

### To start the Internet online help from the product:

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



- the access mode to the help database (local help or Internet)
- the content of the help: common help or help for the product currently used.

## Legend of symbols



This symbol indicates the duration of the lesson. Caution: the actual time may vary according to your level of experience.



An example is available to complement the lesson.



This symbol introduces a "Tip", we strongly advise you to read the associated text.



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



This symbol introduces a "Note", we advise you to read the associated text.



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



Intended for developers who are already familiar with the earlier versions of WebDev.

## If you are familiar with WebDev 18...

---

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

## What is WebDev used for?

---

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

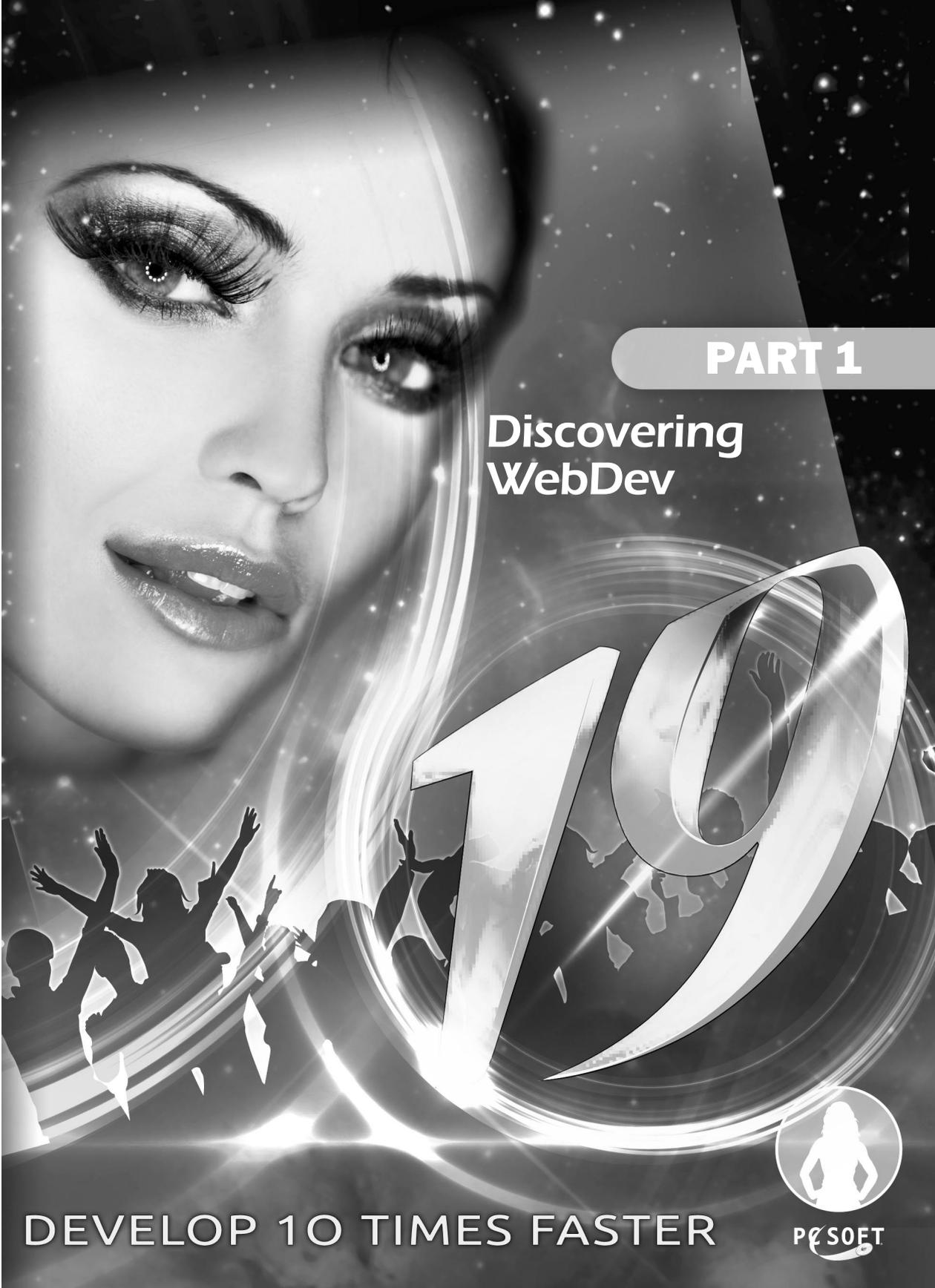
- E-commerce (sales, rentals, bookings, ...)
- Multimedia (description of companies, "showroom" sites, ...)
- Intranet (logins, secure accesses, ...)
- ...

WebDev is a complete 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!



**PART 1**

**Discovering  
WebDev**

**10**

**DEVELOP 10 TIMES FASTER**



**PICO SOFT**



# LESSON 1.1. DISCOVER WEBDEV

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

---

- Starting WebDev



Estimated time: 5 min

## Overview

---

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

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

WebDev allows you to create:

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

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 19 (if not already done).
- ▶ If **WebDev 19 was never started before**, a welcome wizard is displayed. This wizard allows you to:
  - If you own an earlier WebDev version, retrieve the existing configurations.
  - If you are a new user, configure your environment. This allows you to choose the screen configuration used and to configure the Control Centers (the Control Centers will be presented in details in “Control Centers”, page 373).
- ▶ If **WebDev 19 was already started**, identify yourself if necessary. The development environment starts. The home window is displayed.

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

## Development environment

### The editor

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

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



1. Menu of editors, displayed in a 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 view the element currently created or modified in WYSIWYG (What You See Is What You Get).

3. Panes. The WebDev interface includes several 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 "Wizards, Examples and Components" pane (at the bottom) is used to quickly access the full examples, sample pages and controls that can easily be included in your sites.

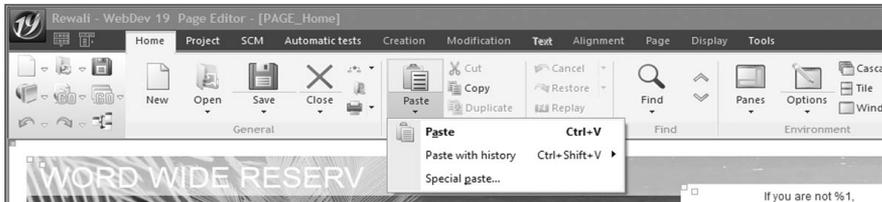
These panes can be hidden by pressing [CTRL] + [W] if necessary.

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

## The menu bar (ribbon) in details

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

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



### The different ribbon elements

The ribbon includes three areas:

- the button area, on the left.
- the pane area, at the top.
- the option area.

Let's take a closer look at these areas.

### The button area



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

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

- The logo 19 is used to display the "About" window and the custom-menus.
- The 2 other logos are used to restore the toolbars and the menus found in the earlier versions.

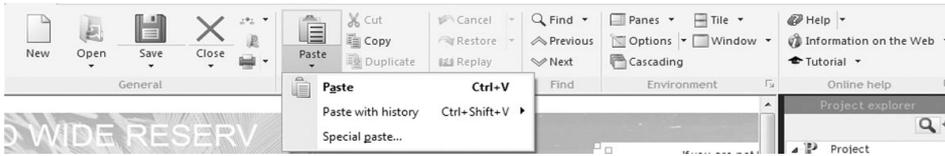
### The pane area



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

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

## The option area



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

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

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

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

For example:

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

## LESSON 1.2. MY FIRST PAGE

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

---

- How do I create a page
- How do I enter and display information
- What are the server code and the browser code used for



Estimated time: 40 min

## Overview

---

To start working with WebDev, we are going to create pages.

**Note**

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

These examples will allow you to understand the operating mode of Internet sites and to get familiar with the programming concepts used in WebDev. And more complex topics will be presented later.

The lessons found in this first part will allow you to:

- create simple pages,
- handle strings, numeric values, currencies,
- handle dates and times.

### Opening the project

- ▶ Start WebDev 19 (if not already done). Close (if necessary) the current project to display the home window.
- ▶ Open the "WW\_My\_First\_Pages" project.  
To do so, in the home window, click "Tutorial" and select the "My first pages (Exercise)" project.  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My first pages (Exercise)".

**Important**

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

**Note**

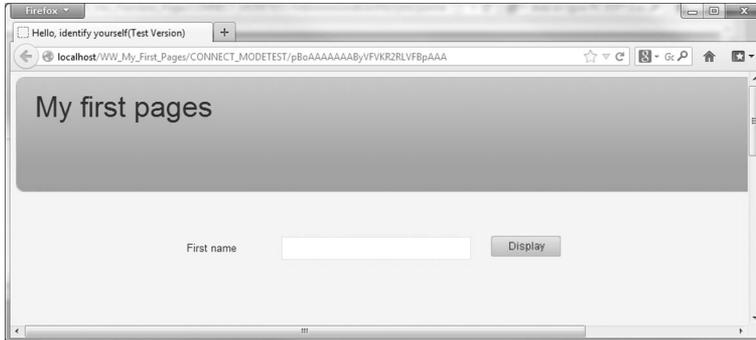
A corrected project is available. This project contains the different pages created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My first pages (Answer)".

## My first page: entering information

---

### Overview

You are going to create the following page:



This page allows the Web user to identify himself and it displays the data entered.

You may think this is a pretty basic idea, ... but we recommend that you create this page. You may well be surprised by how intuitive and easy it is to use the WebDev editor.

Furthermore, this page will teach you principles that are fundamental for the rest of this tutorial.

### Creating the first page

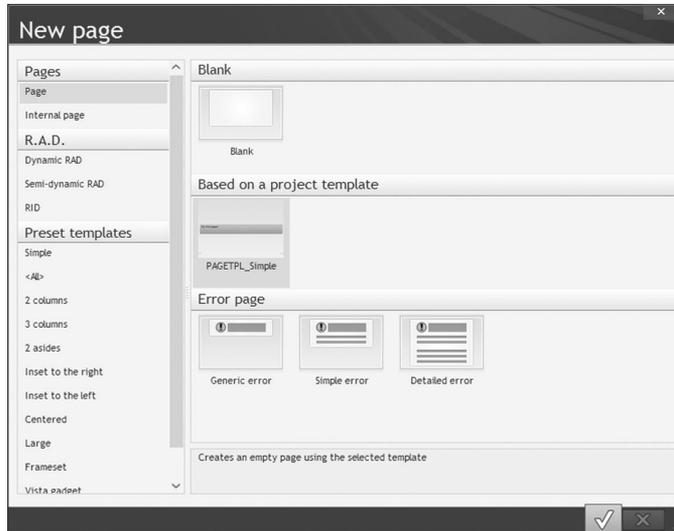
► To create the page:

1. Click  among the quick access buttons of the WebDev menu:



2. A window shaped like a wheel is displayed. This window is used to create all the elements that can be associated with a project.

3. Click "Page". The wizard for page creation is displayed.



4. Select "Page" on the left of the window, then the "PAGETPL\_Simple" template in the list of project templates.



Note

The templates are used to define the style of pages. This concept will be explained later in this tutorial.

5. Validate (green button at the bottom of the screen). The page is automatically created in the editor.

- ▶ We are going to enter the main characteristics of the page: its name and the title that will be displayed in the title bar (or in the tab) of the browser. Double-click the page that was just created. The description window of the page is displayed on the "General" tab. The page is named "PAGE\_NoName1" by default.



Note

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

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

- a page starts with PAGE,
- a button starts with BTN, etc.

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

- ▶ Enter the following information:

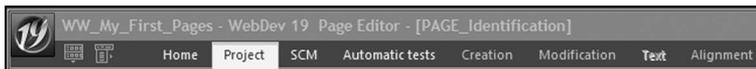


1. The name of the page: replace "PAGE\_NoName1" by "PAGE\_Identification".

This name will correspond to the name used to save the page on disk (with the "WWH" extension) in the directory of the project sources. This name will also be used to handle the page by programming.

2. The title of the page: "Hello, identify yourself". This title is displayed in the title bar of the browser. This title is used to inform the Web user about the features of the page.

- ▶ Validate. The name of the page is displayed in the title bar of WebDev.



- ▶ To save the page, click  among the quick access buttons of the WebDev menu. By default:
  - the name of the page corresponds to the name entered in the page description.
  - the proposed location corresponds to the project directory.



- ▶ Click the green button to validate.



Notes

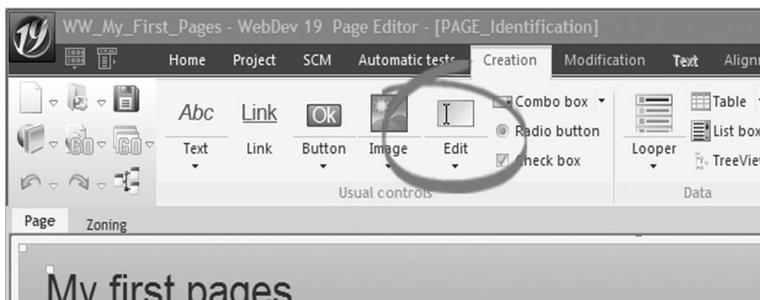
For each page created in the editor, WebDev generates several types of files:

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

## Inputing and displaying the entered value

To manage the input and display of the entered value, you must create:

- a control in which the Web user will type his first name. Therefore, this control is an "edit control".
  - a button used to display the first name.
- ▶ To create the edit control:
1. On the "Creation" pane, in the "Usual controls" group, expand "Edit" (click the arrow):



The list of controls proposed by default is displayed.

2. Click "Simple text" in the list of controls. The control currently created follows the movement of the mouse.
3. Move the mouse in the page toward the position where the control must be created. To drop the control in the page, all you have to do is perform a new left mouse click.

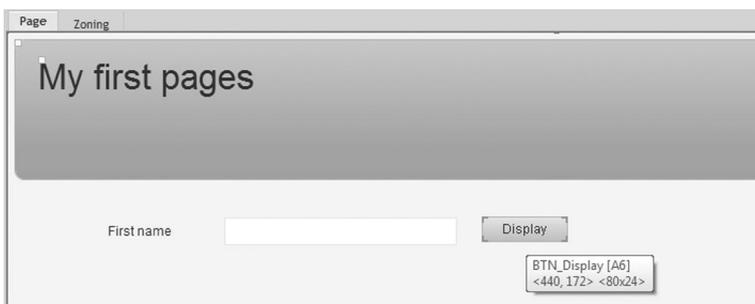
4. Right-click the control that was just created. The popup menu is displayed. Select "Description" from the popup menu. The description window of the edit control is displayed.
5. Modify the characteristics of the control.



- This control is named "EDT\_FirstName".
- The caption of this control is "First name".
- This control was automatically created as a "Text" control. This type of edit control allows the Web user to type a text.

6. Validate the description window of the control (green button). The control appears in the page editor.

- ▶ The caption of the control is truncated in the editor. To display the control properly:
  1. Select the control (all you have to do is click it with the mouse).
  2. In the popup menu (right mouse click), select "Resize".
  3. The control is immediately enlarged.
- ▶ To create the "Display" button:
  1. On the "Creation" pane, in the "Usual controls" group, click **OK**.
  2. Position the control in the page. Click the location where the button must be created (on the right of the edit control for example).
  3. Click the button that was just created. The text displayed in the button becomes editable. Enter the caption: "Display". The name of the button automatically becomes "BTN\_Display". You can see the name of the button:
    - in the tooltip displayed when the control is hovered by the mouse cursor:



- in the status bar of the editor:



- ▶ We are going to display the text in a dialog box (a mini-window proposed by the system). To do so, we will be using our first WLanguage function: **Info**.

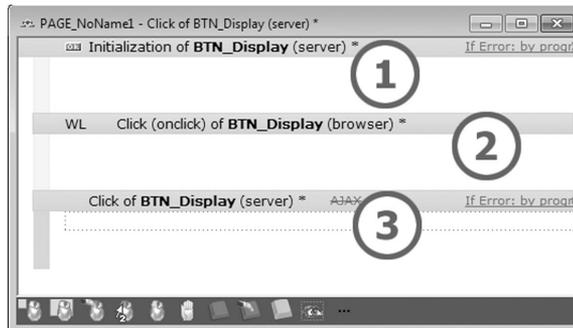


Notes

WLanguage is the programming language supplied with WebDev. It is a 5th generation language (5GL) that uses highly sophisticated commands.

1. Select the "Display" button with the mouse.
2. Display the popup menu of the control (right mouse click).
3. Select "Code". This option opens the code editor of WebDev, in which all the WLanguage statements can be entered.

In the code window that is displayed, the code is divided into several sections of different colors:



- Initialization of BTN\_Display (server), orange code (1)
- Click (onclick) of BTN\_Display (browser), green code (2)
- Click of BTN\_Display (server), orange code (3)

The orange codes are the codes run on the server and the green codes are the codes run on the browser.

**Why are these 2 codes required?**



Notes

Two types of sites are available:

- the static sites, that display pages whose data does not change. These sites include pages whose content is fixed. No specific engine is required to display these pages.
- the dynamic sites (in our example) whose data may change. These sites include pages whose content may change. A specific engine is required to display these pages and to include the data.

For a dynamic site, you have the ability to run processes:

- on the computer of the Web user. It's the browser code. In this case, no request is made to the

engine. In WebDev, the WLanguage code entered is converted to Javascript. The number of WLanguage functions that can be used in this code is limited.

- on the server. It's the server code. In this case, a request is made toward the server to run the specified code. The server redisplay the page.

In this example, we are going to use a browser code to display the information.

- ▶ Enter the following code in the browser click code:

```
Info("Hello " + EDT_FirstName)
```

**Note about the assisted input:** As soon as the first three characters are typed, WebDev proposes all the words of the WLanguage vocabulary containing these characters. The assisted development is very a powerful feature. No more mistake when typing the name of an element: the syntax errors are reduced. All you have to do is select the requested word and press [Enter] to validate. You can focus on the algorithm.



Notes

This is the reason why the programming charter is so important. All the elements handled in the code of the application use the same standard so they can be easily found when typing the code.



Notes

When typing this code in the code editor, you have noticed that different colors are used by the different elements. This is the syntactic coloring. The code editor allows you to easily identify the different elements handled by the code:

- the WLanguage functions are colored in blue,
- the character strings (between quotes) are colored in purple,
- the names of controls are colored in cyan.

**Info** displays the message passed in parameter. Our message is built from the text "Hello" and from the value of the "EDT\_FirstName" control. The "+" sign indicates a "concatenation" operation between two strings.

## Test of a page

To check whether your home page operates properly, you are now going to run the test of your page.

- ▶ Save the modifications made to your page by clicking  among the quick access buttons or by pressing [Ctrl] + [S].
- ▶ Click  among the quick access buttons of the menu. This button is used to run the test of the page in the default browser installed on your computer.



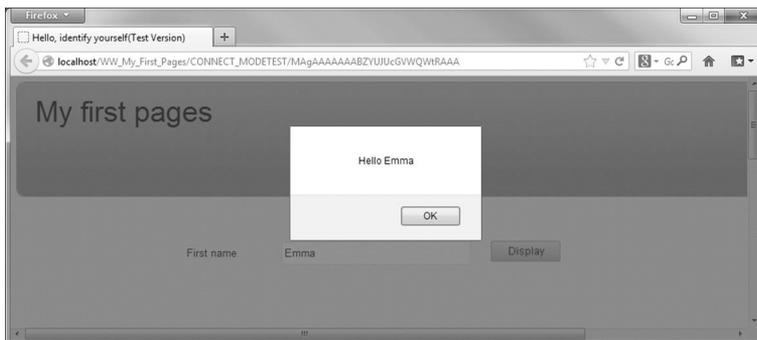
Notes

The icon  is automatically displayed in the taskbar of your computer. This icon indicates that the WebDev engine was started.

Double-click this icon to display the administrator of dynamic WebDev sites. The "Connections" tab allows you to view your current connection to the "WW\_My\_first\_pages" site.

In our case, we are in test mode on the development computer, therefore the user is "TEST". In real configuration, the IP address of the Web user is displayed.

1. Enter a first name in the page that is displayed.
2. Click the "Display" button.



3. Validate the dialog box.
- ▶ We are going to run another simple test.  
We are now going to improve this page:
    - We are going to force the use of an uppercase letter for the first name.
    - We are going to check whether the first name was filled.
  - ▶ Go back to the WebDev editor (to do so, close the test browser for example).

## Checks performed on the "First name" control

### 1. Using an uppercase letter

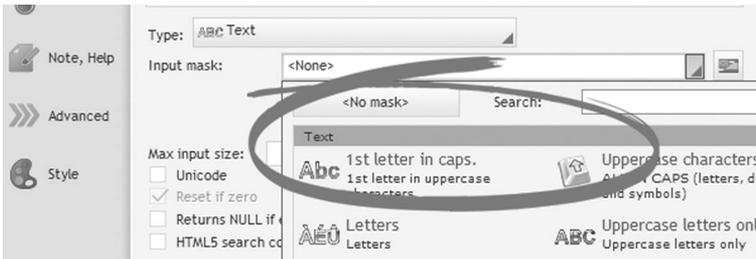
To force the use of an uppercase letter at the beginning of the first name, all you have to do is use an input mask.



Notes

By default, the characters will be entered in uppercase or lowercase characters according to the position of the [CAPSLOCK] key on the keyboard of the Web user. This type of input can be very annoying when performing a search for example. WebDev allows you to manage an input mask for a control. This is used to automatically format the value entered, without any code line.

- ▶ To modify the input mask:
  1. Double-click the "First name" control. The description window of the control is displayed.
  2. In the "General" tab, expand the "Input mask" list and select "1st letter in uppercase".



3. Validate the description window of the control.
4. In the editor, the name of the mask is automatically displayed in the control.

### 2. Checking the information entered

Checking the information entered must be performed in browser code, before the data is sent to the server.

To do so, check whether the value of the edit control corresponds to an empty string. In this case, a message is displayed and the input is forced in the empty control.

- ▶ To run this test, modify the browser click code of the "Display" button ("Code" from the popup menu):

```
IF EDT_FirstName="" THEN
  Info("Enter a first name")
  ReturnToCapture(EDT_FirstName)
ELSE
  Info("Hello " + EDT_FirstName)
END
```

In this code, **ReturnToCapture** is used to force the input in the "First name" control if this one is empty.

- ▶ Save the modifications made to your page by clicking  (among the quick access buttons).
- ▶ Run the test of the page ( among the quick access buttons). A message is displayed if no information is entered.
- ▶ Go back to WebDev.

Your first page was successfully created. In the rest of this lesson, we are going to discover new WebDev concepts.

# LESSON 1.3. MY FIRST PAGES

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

---

- How do I perform a calculation on a date?
- How do I create a page displaying the date and time in real time?
- How do I list the files found in a directory?



Estimated time: 40 min

## Overview

---

In the previous lesson, we have explained how to type and display a text. The associated process was performed in browser code.

In this lesson, we are going to create the following pages:

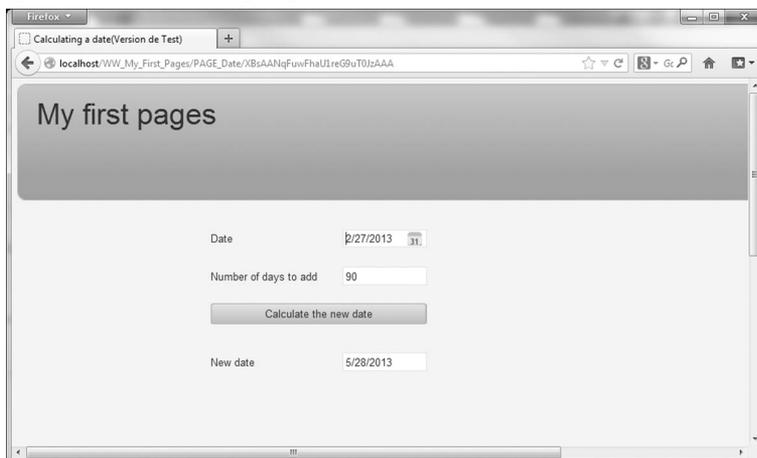
- a page used to perform a calculation on a date in server code.
- a page displaying the time in real time.
- a page used to list the files found in a directory.

## Performing a calculation in server code

---

### Overview

We are going to create the following page:



The button is used to add the number of days to the specified date.

### Implementation

- ▶ Open the "WW\_My\_First\_Pages" project.

To do so, in the home window, click "Tutorial" and select the "My first pages (Exercise)" project.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My first pages (Exercise)"



Note

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

► To create the page:

1. Click  in the quick access buttons.
2. Hover the "Page" category and select "Page". The wizard for page creation is displayed.
3. Select "Page" on the left of the window, then the "PAGETPL\_Simple" template in the list of project templates.
4. Validate. The page is automatically created in the editor.
5. Display the page description: on the "Modification" pane, in the "Fast edit" group, click "Description". In the "General" tab, specify the name of the page ("PAGE\_Date") as well as the title ("Calculating a date").
6. Validate the description window.
7. Save the page (CTRL + S).



Note

It is very important to save the page as soon as it is created.

Saving the page allows WebDev to:

- propose the name of controls in the code.
- propose the automatic renaming of controls in the code.

### Creating the edit controls

We are now going to create the different page controls.

► To create the Date edit control:

1. On the "Creation" pane, in the "Usual controls" group, expand "Edit" (click the arrow). The list of preset edit controls is displayed.
2. Click the "Date" control with a calendar.
3. Click the location where the control must be created in the page.
4. Display the description window of the control ("Description" from the popup menu). We are going to modify its characteristics in the "General" tab. This control:
  - is named "EDT\_Date"
  - has the following caption: "Date".
5. Validate the description window (green button).

► To create the edit control used to enter the number of days to add:

1. On the "Creation" pane, in the "Usual controls" group, expand "Edit" (click the arrow). The list of preset edit controls is displayed.
2. Click the "Integer" control.
3. Click the location where the control must be created in the page (below the "Date" control that was just created for example).
4. Display the description window of the control ("Description" from the popup menu). We are going to modify its characteristics in the "General" tab. This control:
  - is named "EDT\_Day".
  - has the following caption: "Number of days to add".
5. Validate the description window of the control (green button).

6. The caption of the control is truncated in the page. To adapt the size of the control:
  - Select the control.
  - Display the popup menu (right mouse click).
  - Select "Resize".
  - The control is automatically resized.
- ▶ To create the Date edit control used to display the result of the calculation:
  1. On the "Creation" pane, in the "Usual controls" group, expand "Edit" (click the arrow). The list of preset edit controls is displayed.
  2. Click the "Date" control without calendar.
  3. Click the location where the control must be created in the page (below the control that was previously created for example).
  4. Display the description window of the control ("Description" from the popup menu). We are going to modify its characteristics.
    - In the "General" tab: this control:
      - is named "EDT\_NewDate".
      - has the following caption: "New date".
    - In the "GUI" tab, specify that this control is a "Read-only" control. The Web user will not be able to perform an input in this control.
  5. Validate the description window of the control (green button).
  6. Adapt the size of the control.

### Creating the calculation button

We are now going to create the button used to perform the calculation on the date.

- ▶ To create the calculation button:
  1. On the "Creation" pane, in the "Usual controls" group, click **OK**.
  2. Place the button at the requested location in the page (beside the "Number of days to add" control for example). The control is selected.
  3. Click the control again: the caption appears in edit.



4. Enter the caption: "Calculating the new date".
5. Resize the control is necessary ("Resize" from the popup menu).
- ▶ We are going to enter the code run by the button. This code will be run by the server.
  1. Select the button and display the popup menu (right mouse click).
  2. Select "Code". The code editor appears.
  3. In the server click code, enter the following code:

```
d is Date
d = EDT_Date
d..Day += EDT_Day
EDT_NewDate = d
```

Let's study this code:

- A Date variable is declared. This type of variable allows you to easily perform calculations on the dates (we'll see this in the next lesson).
  - The content of the "EDT\_Date" control is assigned to this variable.
  - The number of days is modified by the WLanguage ..Day property.
  - The "EDT\_NewDate" edit control receives the result of the calculation in order to display it.
- Save your page and run its test ( and  among the quick access buttons of WebDev). During the test, enter a date, a number of days and click the button.

## Displaying the time in real time

---

### Overview

We are going to modify the created page in order to display the time. Easy! I guess that WebDev includes a time edit control. Exactly. However, in our case, the time will be updated in real time. To do so, WebDev proposes the "browser timers".

A timer is used to run in parallel a task that must be periodically started (get data in background task, refresh data automatically, ...). Only the procedures (global or local to a page) and the methods of classes can be called by timer.

Two types of timers are available in WebDev: the server timers and the browser timers. In our example, we will be using a browser timer.

Let's create our "Time" control.

### Adding the Time control

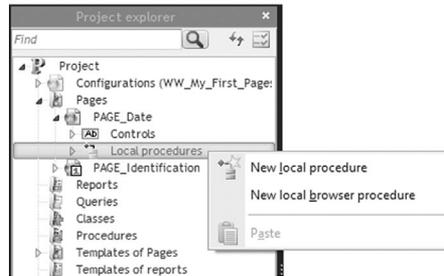
- To create the Time edit control:
1. On the "Creation" pane, in the "Usual controls" group, expand "Edit" (click the arrow). The list of preset edit controls is displayed.
  2. Click the "Time" control.
  3. Click the location where the control must be created in the page (top right corner for example).
  4. Modify the characteristics of this control. This control:
    - is named "EDT\_Time"
    - has the following caption: "The time is".
  5. Validate the description window (green button).
  6. Resize the control if necessary ("Resize" from the popup menu).
  7. Save the page (CTRL + S).

## Implementing the timer

To update the Time control, we are going to call a local browser procedure every second; this procedure will update the time control with the current time.

► To create a local procedure:

1. In the "Project explorer" pane, select the name of the "PAGE\_Date" page.
2. Click the arrow on the left to display the different elements.
3. Click "Local procedures" (right mouse click) and select "New local browser procedure".



4. In the window that appears, specify the name of the procedure ("DisplayTime") and validate ("Add" button).

5. The procedure appears in the code editor. Enter the following code:

```
EDT_Time = Now()
```

This code is used to refresh the control displaying the time. We are now going to call it every second via the timer.

► To implement the timer:

1. Go back to the "PAGE\_Date" page. You can for example click the "PAGE\_Date" button at the bottom:



2. Display the processes associated with the page:

- Display the popup menu of the page (right mouse click on the page).
- Select "Code" from the popup menu. The different processes are displayed.

3. In the "Load (onload)" process of the page, enter the following code:

```
Timer(DisplayTime, 100)
```

4. This code is used to specify that the "DisplayTime" procedure will be called every second (100 represents the milliseconds).

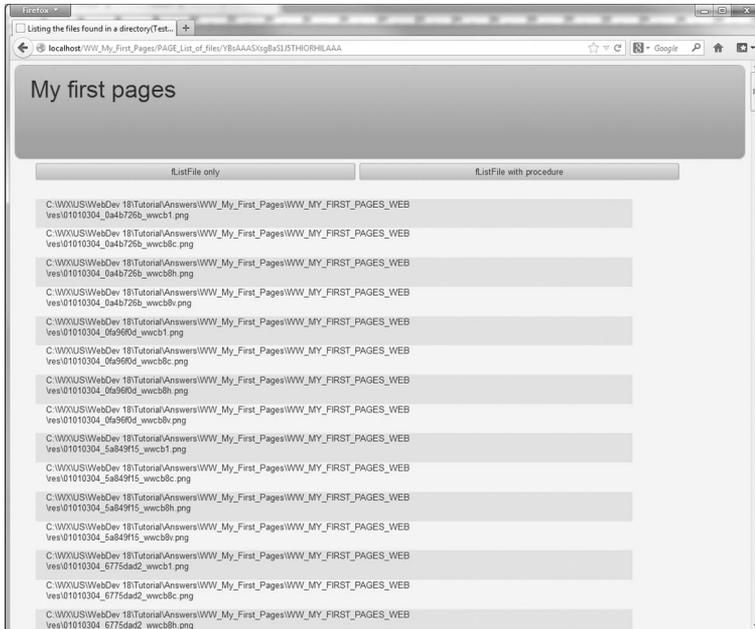
► Our page is created.

Save your page and run its test (by clicking  then .

## List the files found in a directory

### Overview

We are now going to create a page used to list the files found in a directory.



This exercise will familiarize you with several topics:

- the WLanguage functions for file management,
- the Looper controls (often used in Web),
- the programming of "Callback" procedures,
- the anchoring of pages and controls.

Let's create our page.

## Creating the page

► To create the page:

1. Click  among the quick access buttons.
2. Hover the "Page" category and select "Page". The wizard for page creation is displayed.
3. Select "Page" on the left of the window, then the "PAGETPL\_Simple" template in the list of project templates.
4. Validate. The page is automatically created in the editor.
5. Display the description of the page ("Description" from the popup menu of the page). In the "General" tab, specify the name of the page ("PAGE\_List\_of\_files") as well as the title ("Listing the files found in a directory").
6. Save the page (CTRL + S).

## Creating the Looper control

We are going to list the files found in a directory. The best way for displaying elements in list format in Web is to use a Looper control. A Looper control includes an area (containing several control) that is repeated a given number of times.

In our case, the Looper control will display a Static control where the name of the file will be displayed.

► To create the Looper control:

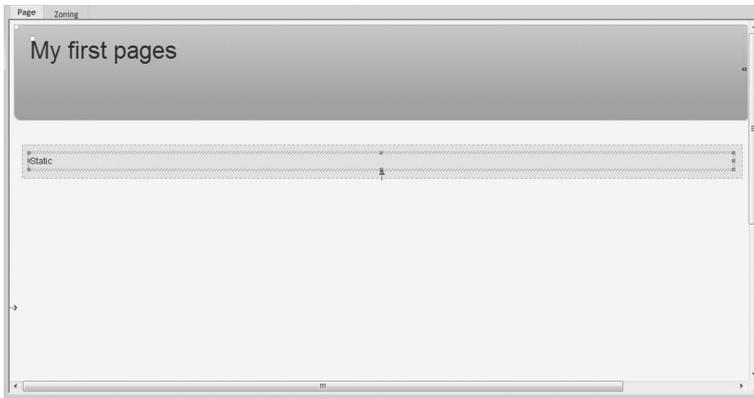
1. On the "Creation" pane, in the "Data" group, click "Looper".
2. Click the location where the control must be created in the page (middle of the page for example). The wizard for creating a Looper control starts.
3. Validate the different wizard screens. In the last screen, specify the name of the Looper control: "LOOP\_File".
4. Modify the size of the Looper control so that it occupies the width of the page.

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

► To create the Static control:

1. On the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Static".
2. This control will be created in the Looper control.
  - Click the location where the control must be created in the page. This control must be created in the Looper control.
  - When hovering the Looper control, a green border indicates that the Static control will be linked to the looper.

3. Position the Static control in the looper and enlarge the control so that it occupies the width of the Looper control. You get the following page:



4. Edit the characteristics of the Static control: this control is named "STC\_File" and its caption is "File". Validate.

- ▶ We are now going to configure the Looper control in order for the content of the static to be modified on each row.
  1. Display the description window of the Looper control (right-click the control and select "Description").



Note

Caution: Select the Looper control instead of the Static control.

2. We are going to modify the top section of the description window. This section is used to:
  - name the looper attribute that will be modified on each row ("ATT\_File" in our case)
  - link this attribute to a control found in the looper ("STC\_File" in our case)
  - specify the property that will be modified on each row ("Caption" in our case).



3. Validate the description window of the looper.

## Programming

To fill the Looper control, we are going to use **fListFile**. This function is used to list the files found in a directory.

We are going to create a button to start the process.

► To create the button:

1. On the "Creation" pane, in the "Usual controls" group, click "Button".
2. Place the button at the requested location in the page (above the looper for example).
3. Modify the caption of the control (by clicking the caption twice for example): "fListFile only".
4. Display the popup menu of the button (right mouse click) and select "Code".
5. Enter the following server click code:

```
sListFile is string
sListFile = fListFile(CompleteDir(fWebDir()+ "res\*.PNG")
LooperDeleteAll(LOOP_File)
FOR EACH STRING sFileName OF sListFile SEPARATED BY CR
  LooperAddLine(LOOP_File, sFileName)
END
```

In this code:

- **fListFile** is used to find the list of "\*.PNG" files found in the "res" sub-directory. The result is a character string (named sListFile) containing the full path of each file. The files are separated by the CR character (Carriage Return).
  - **LooperDeleteAll** is used to clear the looper.
  - The FOR EACH STRING statement is used to browse all the sub-strings separated by CR in the result.
  - For each sub-string, **LooperAddLine** is used to add a row into the looper. The sub-string is assigned to the first looper attribute, which means the caption.
- Close the code editor (click the cross in the top right corner).
- Save your page and run its test (by clicking  then  among the quick access buttons).

## Programming with callback

We are now going to create the same process by using a "CallBack" procedure. A "callback" procedure is a procedure that will be called at regular time intervals by a WLanguage function.

In our case, **fListFile** will call a procedure for each file found in the directory. Therefore, it will be possible to add into the procedure the name of the file in the looper.

Let's create a new button.

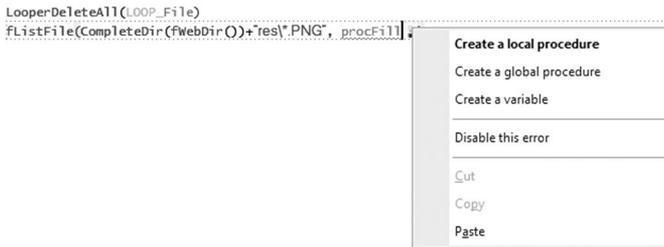
► To create the button:

1. On the "Creation" pane, in the "Usual controls" group, click **OK**.
2. Place the button at the requested location in the page (above the looper for example).
3. Modify the caption of the control (by clicking the caption twice for example): "fListFile with procedure".
4. Display the popup menu of the button (right mouse click) and select "Code".
5. Enter the following server click code:

```
LooperDeleteAll(LOOP_File)
fListFile(CompleteDir(fWebDir())+"res\*.PNG", procFill)
```

6. Save the code (CTRL + S): the "procFill" word appears in red and a compilation error occurs in the "Compilation errors" pane at the bottom of the editor. This name is the name of the "callback" procedure that will be used. WebDev does not recognize it and it triggers an error. We are going to create this procedure

7. Right-click the small bulb displayed beside "procFill" and select "Create a local procedure".



8. A window proposing to create the "procFill" local procedure is opened. Validate this window ("Add" button). The procedure is automatically created and displayed in the code editor.

9. Enter the following code:

```
PROCEDURE procFill(Directory, File)
LooperAddLine(LOOP_File, CompleteDir(Directory)+ File)
RESULT True
```

This code requires some explanations:

- First, the declaration of the procedure. The callback procedure called by **fListFile** expects 2 parameters: the name of the directory and the name of the file currently read. These parameters are automatically filled when **fListFile** calls the procedure.
- **LooperAddLine** is used to add a new row to the looper. The content of the attribute corresponds to the name of the directory concatenated with the name of the file.
- The "RESULT" keyword is used to signal to **fListFile** that the procedure is ended. In this case, it was successfully run: the procedure returns True.

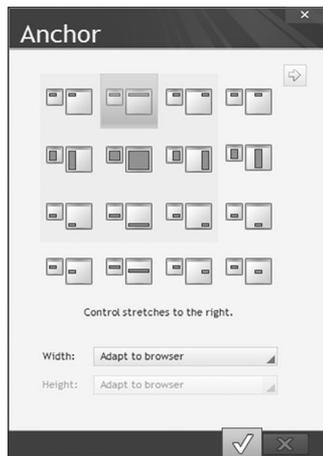
- ▶ Close the code editor (click the cross in the top right corner).
- ▶ Save your page and run its test (by clicking  then  among the quick access buttons).
- ▶ Increase the size of your browser: you have noticed that some paths are displayed over 2 rows in the looper. The looper has a fixed size. We are going to see how an anchoring can be implemented in order for the looper to occupy the entire space the browser.

## Anchoring the page and the controls

WebDev allows you to easily anchor the controls and pages. To do so, select "Anchor" from the popup menu.

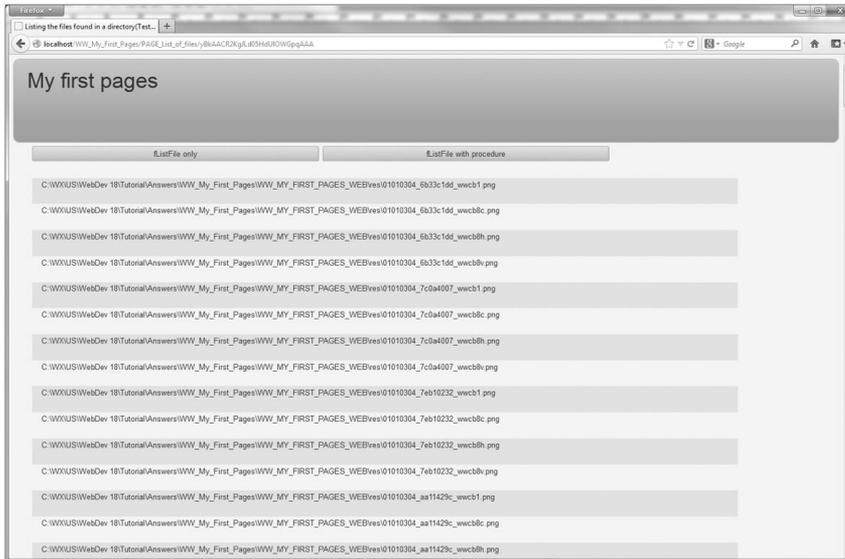
We are going to modify the anchoring of the page; it must be anchored in width to occupy the entire space in the browser.

- ▶ To anchor the page:
  1. Right-click the page to display the popup menu of the page.
  2. Select "Anchor". In the window for defining the anchor, select an anchor in width:



3. Validate.
  4. The anchor marks (red arrows) appear in the page.
- ▶ We are now going to anchor the looper in width as well as the Static control displayed in this looper.
    1. Select the looper and press F4: the last action performed in the page editor is applied to the selected control. In our case, the anchoring that was defined for the page is automatically applied to the looper.
    2. Select the Static control found in the looper and press F4. The Static control is also anchored in width.

- ▶ Save your page and run its test (by clicking  then  among the quick access buttons). When the browser is enlarged, the name of the file is displayed on a single row by occupying the entire available space in the browser.



- ▶ Close the browser.

# LESSON 1.4. CONCEPTS

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

---

- The WebDev concepts and the terminology used
- The different types of pages
- The different types of sites
- Operating mode of a site



Estimated time: 10 min

## Main concepts and terminology

After these exercises, let's talk about the main concepts of WebDev and about the terminology specific to WebDev.

### Main concepts

WebDev allows you to easily create a site. But what is a **Site**?

An Internet site (or an Intranet site or even an Extranet site) is a set of HTML pages stored on a Web server (Web stands for "World Wide Web").

To create a site, WebDev proposes to create a **project**. A project links and organizes the different elements of the site. The site will be created and deployed from the project.

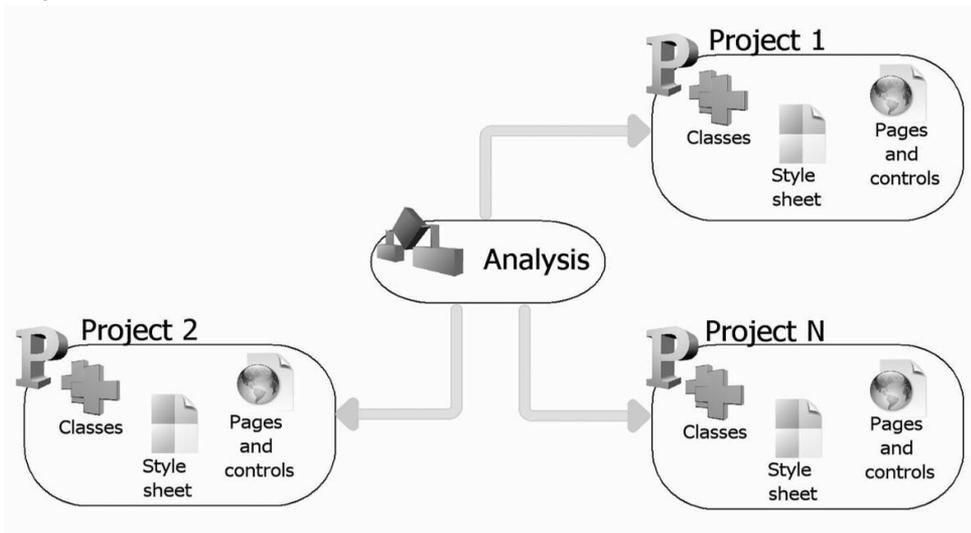
If your site handles data, WebDev allows you to define the structure of the database via **the analysis**. The WebDev analysis contains the description of the data files (also called "Tables" in several databases). These files will contain the data of the application.



Notes

Describing the data files in the analysis does not mean that they are created. The data files are physically created when using the site.

One or more WebDev projects can be linked to the same analysis. In this case, we talk of shared analysis.



## Terminology

As already seen, a WebDev project (linked to an analysis if necessary) is used to create a site. Before we actually start working with WebDev, let's go back to the terms used in WebDev. Indeed, several terms are specific to WebDev and they may differ from the ones used in other tools.

**In the analysis**, the following terms are used:

- **File:** The analysis is used to describe the structure of the database files. A "File" may correspond to a "table" in other databases.
- In WebDev, "**Table**" represents a graphic object used to view the content of a data file in table format an/or to enter rows. A table can be used to enter the details of an order for example.
- **Record:** A record is also called row. A file record corresponds to all the items defined for the file.
- **Item:** In the analysis, an item represents a section of a data file. All the items found in a data file are used to define the structure of a record.
- **Key/Index:** With WebDev and its HFSQL database, the concept of index is linked to the concept of key. The concept of key is part of the characteristics of an item. The keys are used to improve the speed for accessing the data and to simplify the browse operations performed on the data files. In WebDev, if a HFSQL data file includes several key items, a single index file will be created at run time.

**In the pages and reports**, the following terms are used:

- **Page:** The pages are used to display or enter information. The Web user can directly act on the pages via controls, buttons, ...
- **Report:** The reports are used to get a custom view of information. This information can come from the database, from text files, from controls found in the pages, .. The reports can be generated in PDF, HTML, ...
- **Control:** The notion of "control" is used to represent the different graphic objects displayed in a page or in a report.
- **Template:** The templates are used to define the "appearance" of different elements used in the site. There are:
  - The page template: it is used to define the "appearance" of the site: appearance of the pages, buttons, controls, ...
  - The control template: it is used to group a set of controls to define a feature.
  - The report template: it is used to define the "appearance" of the reports that can be printed from the site: all the reports will use the same style book (logos, margins, ...).
- **Style:** The style groups the graphic characteristics of an element: background image, border, font, ... The styles of the different elements found in the interface of a WebDev site are grouped in a style sheet.

- **Skin:** The skins are used to standardize the visual aspect of a site and they allow you to easily change style. A skin includes:
  - a style sheet containing WebDev styles.
  - an image directory.
  - a default color palette.
  - a set of reports used as base to build the new reports of the project.



Important

In a project, the "CustomerName" entity can correspond to:

- the name of a page control
- the name of a report control
- the item of a data file
- a variable defined by the developer

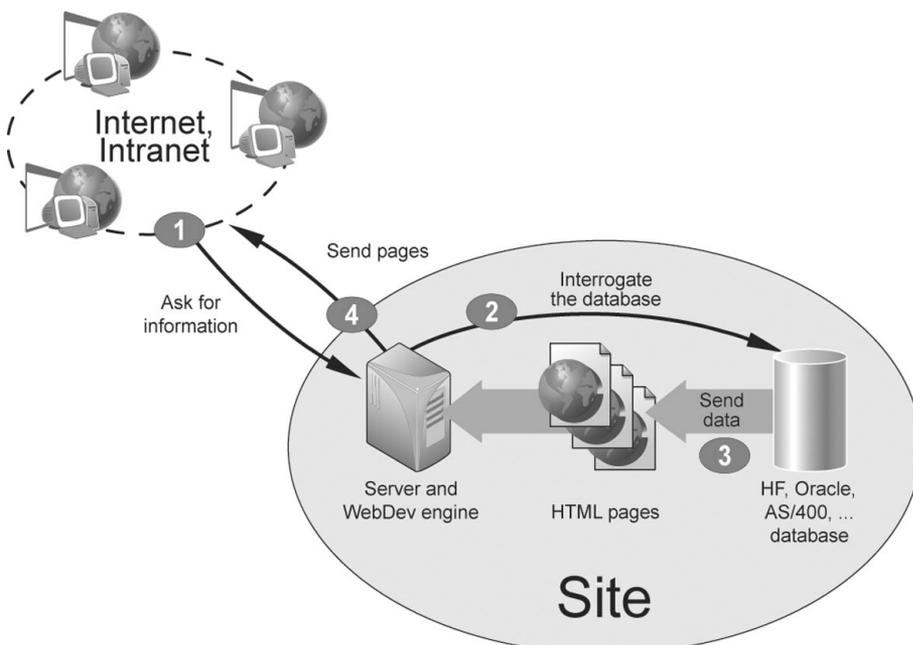
## The different types of pages proposed by WebDev

WebDev allows you to easily create Internet/Intranet sites that manage the data. WebDev can also be used to create "static" sites that manage no data.

### The different types of pages

#### Dynamic pages

When your site needs to manage real-time data and Web users, link your company to Internet (online orders, ...), ... choose dynamic pages.



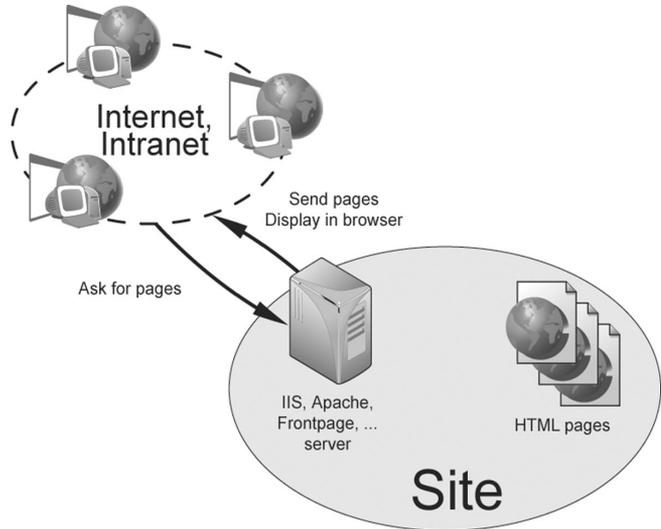
A dynamic page is used to display data coming from a database, as well as images and interactive text.

For example, to display information about several products, a single dynamic page is all you need: when a product is selected, the page will display the information regarding this product.

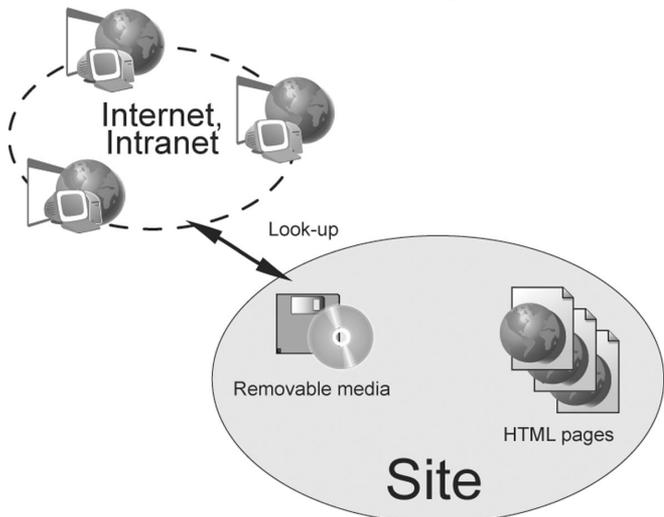
**Static pages**

When your site displays fixed text and images, choose static pages. These static pages can be:

**1. Hosted on a server**



**2. Distributed on removable media**



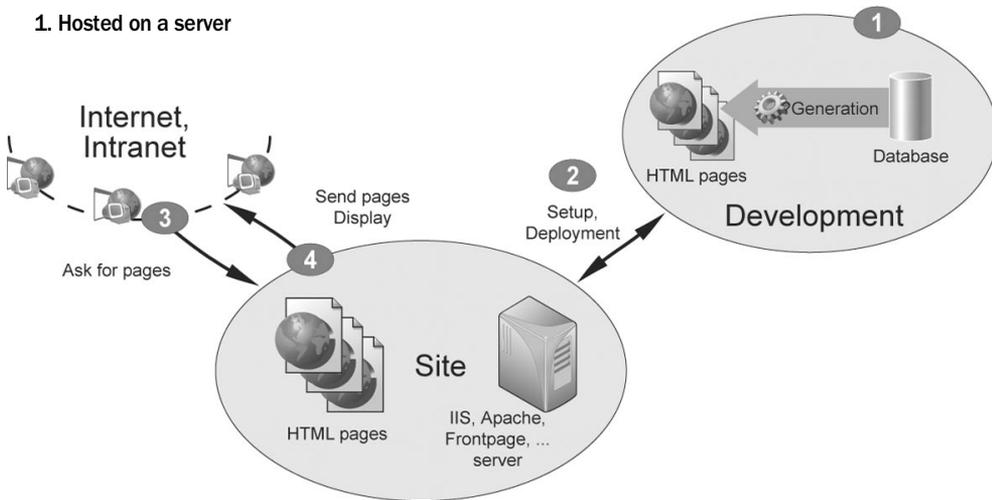
A static page is used to display text and images but the Web user cannot "interact" with them. To display information about several products, you will have to create as many static pages as the number of products.

In most cases, this type of page is used to present a company and to propose "standard" information that will not change over time.

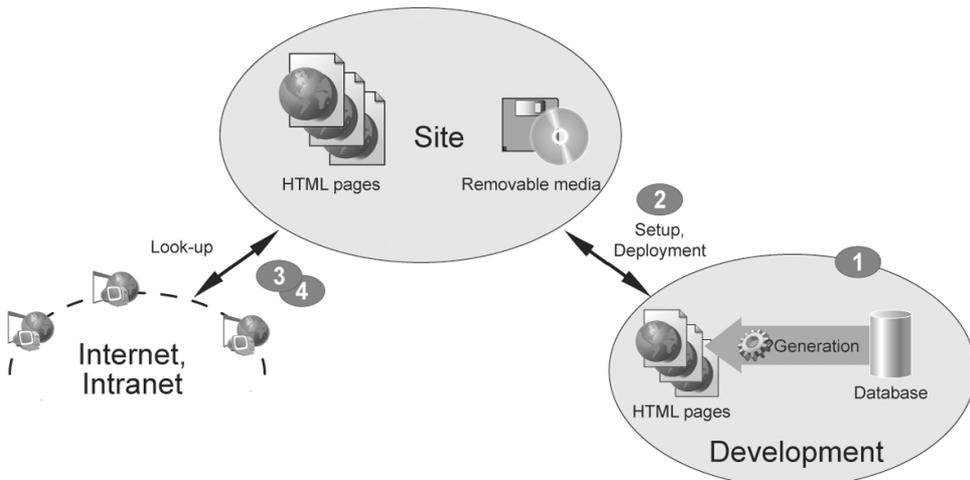
**Semi-dynamic pages**

Many companies propose their product catalog on CD. It may be interesting to choose an Internet site. Indeed, an Internet site on CD only requires an Internet browser, regardless of the operating system installed on the user computer.

**1. Hosted on a server**



**2. Distributed on removable media**



The semi-dynamic pages are perfectly suited for this choice. These are "static" pages automatically created by WebDev. They are generated from a database. This enables you to professionally present your products, their characteristics, ... without having to write a single code line.

## The different types of sites

### The dynamic sites (WebDev or PHP)

A dynamic site is used to manage data. It can contain dynamic pages, static pages or semi-dynamic pages for a WebDev site. A PHP site can contain PHP pages or static pages.

A dynamic site can also be used to perform operations on the server (generating orders, creating images, ...). A dynamic site operates by using the runtime engine (installed on the server):

- WebDev engine for a WebDev site
- PHP engine for a PHP site

### The static sites

A static site is often used as a "showroom". It can contain static pages or semi-dynamic pages. A static site cannot be used to perform operations on the server.

The static sites require no specific setup on the server. The static sites can also be distributed on CD.

### The semi-dynamic sites

A semi-dynamic site is a site that displays static pages and semi-dynamic pages. The semi-dynamic pages are used to display data (coming from a database). The content of these pages was generated when developing the site. A semi-dynamic site cannot be used to perform operations on the server.

The semi-dynamic sites require no specific setup on the server. The semi-dynamic sites can also be distributed on CD.

## A Web server and browsers ...

---

### The different browsers

A site is intended to be used by the Web users. The Web users use a browser to access the site.

Several browsers allow you to access the Internet sites according to the operating system. The most common ones are:

- Internet Explorer
- Firefox
- Chrome
- Safari
- Opera
- as well as several browsers on the cell phones that can access Internet.

Each browser can have its own features and can be more or less compliant with the Internet standard.

WebDev allows you to create Web sites that support ALL the browsers.

## The different Web servers

A Web server is a computer running an Internet service software. It is also called "HTTP server". To simplify things, we will be using "Web server" for the Internet service software.

The following Web servers are automatically recognized by WebDev:

- Microsoft Internet Information Server (IIS, from version 2.0),
- Apache (versions 1.3) for Windows,
- Apache2 (versions 2.0, 2.2, 2.4) for Windows.

Other Web servers can be configured in order to operate with WebDev. Regardless of the browser used by the Web user, your sites will be accessible anywhere in the world!

## What's happening on the browser side ...

On the browser side, you will find a bright-eyed Web user!

More seriously, the browser is used to display:

- the pages with the data sent by the server in case of dynamic pages.
- an HTML page stored on the server in case of static pages or semi-dynamic pages.

From the browser, you can:

- perform input checks (to check that no mistake was made by the Web user when entering a date for example),
- display a message,
- open another browser.

These "browser" actions can be programmed via WebDev WLanguage when developing your site. These processes will be run on the computer of the Web user, in the browser.

You also have the ability to write these processes in Javascript. We advise you to program in WLanguage for better legibility.

For the static sites, no browser action can trigger a "server" action.

For the dynamic sites, each page "validation" (click on a button, validation of input, ...) triggers a "browser" action then a "server" action.

## What's happening on the server side ...

The "server" side exists only if your site uses dynamic pages.

Each action performed by the server was triggered by an action performed by the Web user in his browser.

From the server, you can:

- check the information entered in the browser by the Web user
- return images, files or pages to the browser
- manage the data files
- generate documents in different formats: PDF, HTML, Word, Excel, XML, ... and send them to the browser.
- draw images, charts, ...

These "server" actions are programmed in WLanguage when developing your site (orange bars). The server actions require the setup of a WebDev runtime engine on the server.

This engine is automatically installed on your development computer (the WebDev Administrator allows you to configure this engine). When deploying your site, this engine is installed at the host company (with the WebDev application server).

Those are the main notions required to create a WebDev site. We will now start programming by creating our first pages.

# LESSON 1.5. PROGRAMMING CONCEPTS

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

---

- Server code/Browser code
- The different types of variables
- Main statements of WLanguage
- Procedures and functions
- Processing the strings, numeric values and currencies
- Processing the dates and times



Estimated time: 30 min

## Introduction

The code editor of WebDev and its language (WLanguage) have been presented in the previous lesson. This lesson allows you to discover the programming concepts of WLanguage by presenting the following topics:

- Server code and browser code
- Declaring the different types of variables
- Main statements of WLanguage
- Procedures and functions
- Processing the strings
- Processing the numeric values
- Processing the dates and times

## Server code and browser code

In the previous lesson, you have noticed that two types of code could exist:

- the browser code, run on the computer of the Web user.
- the server code, run on the Web server. This code is available for the sites that require a runtime engine on the server.



Notes

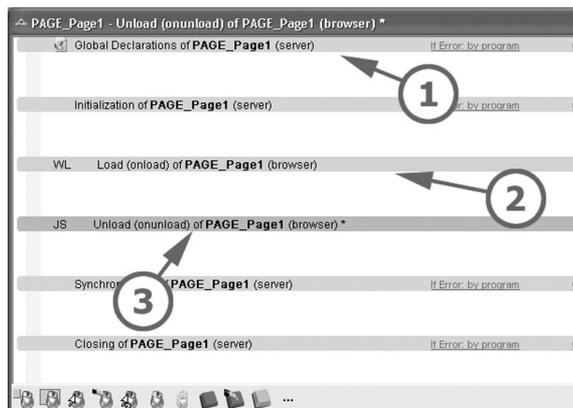
WebDev allows you to create several types of sites: static site, semi-dynamic site, dynamic site.

The pages found in the dynamic sites include a server code and a browser code. The actions can be performed locally (on the computer of the Web user) and on the server (displaying a page that contains information read in a database for example).

The pages found in the static sites include a browser code only.

In the code editor:

1. The server code is represented by an orange bar.
2. and 3. The browser code is represented by a green bar (WLanguage mode) or by a blue bar (Javascript mode).



## Server code

The server code is run on the server. When the Web user performs an action in a page, the browser code is run and the page is sent to the server. The server code is run before sending a response page to the browser of the Web user.

This server code is used to perform all the possible actions on the server:

- management of database,
- management of external files,
- creation of charts, ...

The WebDev engine is required to run this code (PHP engine for a PHP site). This code is available in the dynamic sites only.

For the dynamic PHP sites, you also have the ability to enter PHP code in the code editor directly.

- To switch from WLanguage code to PHP code, all you have to do is click the "WL" logo found in front of the code caption.



- To switch from PHP code to WLanguage code, all you have to do is click the "PHP" logo found in front of the code caption.



## Browser code

A browser code is run on the computer of the Web user. The browser code is run when the Web user performs an action in a page. This code requires no action on the server side.

When the page is saved in the editor, the browser code entered in WLanguage is automatically converted into Javascript and included in the HTML pages created by WebDev.

The browser code can be written in WLanguage or in Javascript:

- To switch from WLanguage code to Javascript code, all you have to do is click the "WL" logo found in front of the code caption.



- To switch from Javascript code to WLanguage code, all you have to do is click the "JS" logo found in front of the code caption.



## Declaring the different types of variables

WebDev proposes several types of variables (boolean, integer, real, currency, string, date, time, duration, datetime, variant, array, structure, ...).

The syntax used to declare a variable is as follows:

```
VariableName IS VariableType
```

### Some examples:

```
Subscript is int
SupplierName is string
ArrayPrice is array of 10 currencies
I, J, K are int
// Declaration and assignment
Counter is int = 120
B1 is boolean = False
```

See the online help (keyword: "Type of data") for more details.

### Declaring the variables and their scope

Two types of variables can be handled in WLanguage:

- The global variables
- The local variables

The global variables can be global to the project, to a page or to a set of procedures. These variables must be declared:

- in the server code for initializing the project (server variables global to the project). These variables can be used in the server code of all the processes of the project and project elements (pages, reports, ...).
- in the server code for declaring the global variables of a page (server variables global to a page). These variables can be used in the server code of all the processes of the page and page elements.
- in the server code for declaring the sets of procedures (server variables global to a set of procedures).



Caution!

The global variables declared in server code are available in the browser code of the page for the following types only: boolean, integer, real, string or associative array.

The modifications made in browser code to a global variable (assignment of value for example) are not carried over in server code.

- in the browser code for loading the page (browser variables global to a page). These variables can be used in the browser code of all the processes of the page and page elements.



Caution!

The browser variables global to a page are not available in the server codes of the page.

All the variables declared elsewhere are local variables.



Caution!

**Local variable: specific feature in browser code**

In browser code, only the following types can be used to declare the local variables: boolean, integer, real, string, simple array or associative array of one of the previous types.



Caution!

Most developers are tempted to declare all their variables as "global" variables in a project. This type of programming is quite "easy". All the variables being global variables, they can be handled from any process.

But this type of programming often causes overwritten variables and exception problems.

Therefore, we advise you to declare a limited number of global variables and to declare local variables when necessary.

If you want to share values between 2 pages, we highly recommend that you use the method for "passing parameters" (see the online help for details).

## Main statements of WLanguage

---

WLanguage is a 5GL made of:

- WLanguage functions
- WLanguage properties
- WLanguage keywords
- WLanguage statements

You will discover all these statements as you progress through this tutorial.

In this lesson, we will only present the main statements, used to perform conditions and loops or to enter comments.

## Conditional statements (server code and browser code)

WLanguage allows you to manage the conditional statements such as :

- **IF, ELSE, END** to run a test on a condition.
- **SWITCH, CASE, END** to run one or more actions according to the results of a test on a condition.

### Some examples

```
IF Maximum > Random(1, 999) THEN
    Info("Congratulations, you've won!")
ELSE
    Info("Back luck, you've lost!")
END
```

```
SWITCH Day
CASE "Monday"
    // First day of the week
CASE "Wednesday"
    // Third day of the week
CASE "Friday"
    // Fifth day of the week
CASE "Saturday", "Sunday"
    // It's the weekend
CASE ELSE
    // it's another day
END
```

## Loop statements (server code and browser code)

WLanguage proposes several methods for managing the loops :

- **FOR, END** for a specified number of iterations.
- **FOR EACH, END** to perform different types of browses: browsing a HFSQL file, browsing the array elements, browsing the sub-strings of a string, browsing the control elements, ...
- **WHILE, END** for a specified number of iterations whose exit condition is checked at the beginning of the loop.
- **LOOP, END** for an undefined number of iterations whose exit condition is checked inside the loop. The **BREAK** statement is used to exit from this loop.



Notes

WLanguage also proposes FOR EACH and END loops used to browse the control elements, the character strings, the records of a data file, ... These advanced loops will be presented later in this tutorial.

## Some examples

```
FOR Sub = 1 to 100 // Sub does not have to be declared
  Cnt = Cnt + 1
END
```

```
Sub is int = 0
LOOP
  Sub ++
  IF Sub>Maximum THEN BREAK
END
```

```
Sub is int = 0
WHILE Sub<Maximum
  Sub ++
END
```

**Tip**

### Server code

We recommend that you use *Multitask* in a long loop in order to give control back to the other server programs or to the operating system while the loop is run.

## The comments

To enter comments in the code, the code line must start with // (two "/" characters). These comments can be written in server code or in browser code.

Example :

```
// This is a comment line
```

**Tip**

Several code lines can be set in comment; to do so, select the code lines with the mouse or with the keyboard and press [CTRL] + [/] (on the numeric keypad).

To perform the reverse operation(to remove the comments), select the code lines with the mouse or with the keyboard and press [CTRL] + [SHIFT] + [/] (on the numeric keypad).

## The MySelf keyword (server code and browser code)

**MySelf** is a keyword that returns the name of the current control. **MySelf** is used to make a local code (process of a control, ...) or a global code (global procedure, class, ...) independent of the current control.

```
MySelf..Value = "New value"
```

## The MyPage keyword (server and browser code)

**MyPage** is a keyword that returns the name of the current page. **MyPage** is used to make a global code (global procedure, ...) independent of the current page:

```
TitleWelcome = "You are here > " + MyPage..Caption
```

## Practical example on the different main statements

Now that you are familiar with the main statements, let's try to run a test!

- ▶ Start WebDev 19 if not already done. Close the current project: on the "Home" pane, in the "General" group, expand "Close" and select "Close the project".
- ▶ In the home window, click "Tutorial" and select the "Programming concepts" project. The project is loaded.  
Tip: you also have the ability, on the "Home" pane, in the "Online help" group, to expand "Tutorial" and to select "Programming concepts".
- ▶ Open the "PAGE\_Code.WWH" page: in the "Project explorer" pane, in the "Pages" folder, double-click the name of the page. This page presents several examples.



Tip

To quickly find a page in the current project, press [CTRL] + [E]. A window is displayed, allowing you to perform a search on all the pages containing the letters typed in the search control.

- ▶ Run the test of this page (  among the quick access buttons).
  1. Click the "Loop" tab.
  2. Check the different examples.
  3. Click the "IF/Switch" tab.
  4. Check the different examples.
  5. Close the browser and go back to the editor.

## Main WLanguage operators

WLanguage includes several types of operators:

- the logical operators (AND, OR, NOT, ...)
- the arithmetic operators that are used to add, subtract, ...
- the comparison operators that are used to compare values. Some of them have been used in the previous paragraph (>, <, =, ...)
- the binary operators that are used to perform operations on the binary values,
- the assignment and exchange operators (=, <=>)
- the operators on the character strings (they will be presented in "Processing strings (server and browser code)", page 74).
- ...

We are going to present the use of the logical operators, comparison operators and indirection operators. The other types of operators will be presented when necessary in this tutorial. See the online help (keyword: "Operators") for more details.

## Logical operators

WLanguage includes 5 logical operators that are used to perform logical operations and to build conditions:

- AND and `_AND_` that correspond to the logical multiplication.
- OR and `_OR_` that correspond to the logical addition.
- NOT that corresponds to the logical negation.

### Practical example

Nothing is better than an example to understand the subtlety of these operators.

- ▶ In the "WW\_Programming\_Concepts" project, open the "PAGE\_Operator" page.
- ▶ Run the test of this page ( among the quick access buttons) and check the logical operators.

## Comparison operators

WLanguage includes several comparison operators that are used to perform several processes:

- equality operators: equality (`=`), flexible equality (`~=`) or very flexible equality (`~~`).
- comparison operators: different (`<>`), less and greater than (`<`, `<=`, `>`, `>=`) or starts with (`[=`, `[=~`, `[=~~`).

### Practical example

An example page is available for checking the different use cases of the comparison operators.

- ▶ In the "WW\_Programming\_Concepts" project, open the "PAGE\_Operator" page.
- ▶ Run the test of this page ( among the quick access buttons) and check the comparison operators.

## Indirection operators

The "indirection" mechanism is used to build the name of a control, the name of a file item or the name of a variable from a string expression.

This mechanism can be used to create generic processes independent of the names of controls, variables, file items, ...

### The indirection is performed by the operators { }.

To optimize the speed of your applications, the type of the element handled should be specified when using the indirection syntax.

Therefore, if you handle a control, the corresponding type will be *IndControl* (see the example below).

Some indirection examples:

- Example of simple indirections:

```
{ "NAME" } = CustName
//is equivalent to NAME=CustName
{ "NAME" } = { "CU.CUSTNAME" }
//is equivalent to NAME=CU.CUSTNAME
{ "CUSTPAGE.NAME" } = CustName
//is equivalent to CUSTPAGE.NAME=CustName
{ "CUSTPAGE"+"NAME" } = CustName
//is equivalent to CUSTPAGE.NAME=CustName
```

- Example of indirection with a procedure:

```
ControlName is string
ControlName = "EDIT1" //EDIT1 is the name of the control
//call a procedure used to make a control invisible
INVISIBLE(ControlName)

PROCEDURE INVISIBLE(NControl)
{NControl,indControl}..Visible = False
```

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

## Procedures and functions

---

### Definition

When a process is called several times in a project or in a page, it may be useful to create a procedure containing this process. Then, all you have to do is call the procedure or the function whenever required.

The procedures and the functions are available in programming:

- The **functions** return a result
- The **procedures** are used to run a specific process.

WebDev allows you easily manage the two types of processes: in WebDev, there is no difference between a procedure and a function. This is the reason why we shall talk about "procedure" in the rest of this lesson.

You have the ability to create "local" procedures and "global" procedures.



Caution!

The procedures in server code are not accessible in browser code and the procedures in browser code are not accessible in server code. However, a server procedure can be run from a browser code via the validation of a button.

### Local procedure

A "local" procedure is linked to a single page.

A local procedure can only be used in the processes of the page in which it was declared (and in the processes of the controls found in this page). It is part of the page.

### Global procedure and set of procedures

The "global" procedures are stored in "sets of procedures". Each "set of procedures" is a file containing all the global procedures that are associated with it.

In a project corresponding to a dynamic site, two types of sets of procedures can be used:

- A set of server procedures, file whose extension is ".WDG".
- A set of browser procedures, file whose extension is ".WWN".

You can create as many sets of procedures as necessary in a project.

A set of procedures is used to:

- share the global procedures between several developers, for the same project.

- share the global procedures between several projects. Indeed, the same set of procedures can be used in several projects.

### How do I choose whether a procedure must be global or local?

To choose whether a procedure must be global or local, ask yourself the following question: "Will the procedure be used in this page only or can it be called from another page?"

- If the procedure is called "from this page only", the procedure can be "local".
- If the procedure can be called "from other pages", the procedure must be "global".

### About passing parameters

In the "WW\_My\_First\_Pages" project, you have noticed that a procedure was managing parameters. The parameters can be mandatory or optional parameters.

The "mandatory" parameters are always defined before the "optional" parameters. The declaration of an "optional" parameter is performed by assigning a default value when declaring the parameter. Example:

```
PROCEDURE MyProcedure(Param1, OptionParam = "Default value")
```

### Calling a procedure

To call a procedure in a process, all you have to do is write its name in the code editor and specify parameters if necessary.

**Reminder:** a server procedure can only be called from a server process and a browser procedure can only be called from a browser process.

WebDev manages the calls to nested procedures.



Tip

In the code editor, when you are positioned on a procedure name, press the [F2] key if you want to view the code of this procedure. Press [CTRL]+[F2] to go back to the name of the procedure in the previous code.

### Creating a procedure

A procedure can be created from the main editor of WebDev. All you have to do is use the "Project explorer" pane.

- ▶ To create a set of global procedures:
  1. Select "Procedures" in the "Project explorer" pane.
  2. Right-click and select:
    - "New set of procedures" to create a set of server procedures.
    - "New set of browser procedures" to create a set of browser procedures.
  3. Specify the name of the set of procedures and validate. The new set of procedure is displayed in the project explorer.
  4. Save the set of procedures by clicking  among the quick access buttons)
- ▶ To create a global procedure:
  1. Select the set of procedures in which the global procedure must be created.
  2. Right-click and select "New global procedure".
  3. Specify the name of the global procedure and validate.
  4. The code editor is displayed with the new procedure that was just created. You can write the code of the procedure.

- ▶ To create a local procedure:
  1. In the "Project explorer" pane, select the page associated with the procedure.
  2. Display the elements of the page (click the small arrow in front of the name of the page).
  3. Select "Local procedures" in the elements of the page.
  4. Right-click and select:
    - "New local procedure" to create a server local procedure.
    - "New local browser procedure" to create a browser local procedure.
  5. Specify the name of the procedure to create and validate.
  6. The code editor is displayed with the new procedure that was just created. You can write the code of the procedure.

**Note:** A procedure can also be directly created from the code selected in the code editor ("New .. Create a procedure containing the selected code" from the popup menu).

## A practical example about the procedures and functions

Enough theory, let's get down to work!

- ▶ Open the "WW\_Programming\_Concepts.WWP" project if not already done.
  1. Close the current project if necessary.
  2. In the home window, click "Tutorial" and select the "Programming concepts" project.
- ▶ Open the "PAGE\_Code.WWH" page (double-click its name in the project explorer).
- ▶ Run the test of the page (  among the quick access buttons).
  1. Click the "Procedures/Functions" tab.
  2. Click the "Call with 1 parameter" and "Call with 2 parameters" buttons to run a procedure.
  3. Click the "Call to function" button to run a function.The code run is displayed in any case.
- ▶ Go back to the WebDev editor.
- ▶ Display the list of local procedures (accessible from the "Project explorer" pane). When you double-click the name of the procedure, the code of this procedure is displayed in the code editor.

## Processing strings (server and browser code)

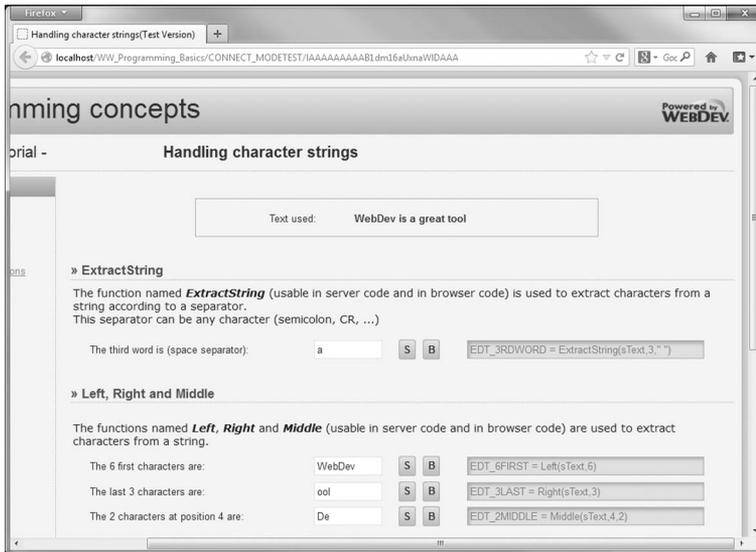
The ability to handle character strings is one of the most important features of a programming language.

Several WLanguage features can be used to handle the character strings: WLanguage functions, extraction operators, concatenation operators, ...

The most common functions for handling the character strings will be presented in this tutorial. See the online help (keyword: "Character string") for more details.

### Practical example

- ▶ Open the "WW\_Programming\_Concepts.WWP" project if necessary.
  1. Close the current project if necessary.
  2. In the home window, click "Tutorial" and select the "Programming concepts" project.
- ▶ Open the "PAGE\_String.WWH" page (double-click its name in the "Project explorer" pane). This page presents the different operations that will be explained in this lesson.



- ▶ Run the test of the page by clicking  (among the quick access buttons).

### Details (server and browser code)

A text control (a static control for example) can be initialized:

- with the string directly:

```
STC_Text = "WebDev is a great tool"
```

- with a string variable :

```
MyString is string = "I'm learning to work with WebDev"
STC_Text2 = MyString
```

A string can be built from several other strings. We talk of **string concatenation**.

To concatenate two strings, all you have to do is use the "+" operator:

```
// Info is used to display the result on the screen
Info(STC_Text + STC_Text2)
```

A section can be extracted from a string by:

- the `[[` and `]]` operators (caution: no space must be found between the `[[` and `]]` brackets).

```
Info(STC_Text[[1 à 6]]) // Displays "WebDev"
```

- **ExtractString** that extracts a sub-string from a string:

```
//Extracts the first word from STC_Text
Info(ExtractString(STC_Text,1," ")) // Displays "WebDev"
```

- **Middle**, that extracts a section of string from a string:

```
Info(Middle(STC_Text2,30,6)) // Displays "WebDev"
```

- **Left**, that returns the left section of a string:

```
Info(Left(STC_Text2,12)) // Displays "I'm learning"
```

- **Right**, that returns the right section of a string:

```
Info(Right(STC_Text,10)) // Displays "great tool"
```

The size of a string is returned by **Length**:

```
Info(Length(STC_Text2)) // Displays 32
```

A string can be converted into uppercase characters by **Upper** or into lowercase characters by **Lower**:

```
Info(Upper(STC_Text2))
// "I'M LEARNING TO WORK WITH WEBDEV"
Info(Lower(STC_Text2))
// "I'm learning to work with webdev"
```

A string can be sought in another string by **Position**:

```
SoughtString is string = "WebDev"
Pos is int
// Retrieves the position of "WebDev" in STC_Text2
Pos = Position(STC_Text2, SoughtString)
IF Pos = 0 THEN
// "WebDev" was not found in STC_Text2
Info(SoughtString + " was not found in text 2")
ELSE
// "WebDev" was found in STC_Text2
Info(SoughtString + " was found in text 2")
END
```



Tip

**Server code**

You also have the ability to find the position of a character string inside another string while ignoring the case. To do so, use **Position** associated with the **IgnoreCase** constant.

Example :

```
Pos = Position(STC_Text2, SoughtString, 1, IgnoreCase)
```

To find out the number of occurrences of a given character string in another character string, use **StringCount**:

```
NbOccurrences is int  
NbOccurrences = StringCount("anastasia", "a") // Returns 4
```

## Processing the numeric values (server and browser code)

The calculations on numeric values can be performed from numeric edit controls or by using the typed variables directly (integer, real, numeric, currency, ...)

### Practical example

- ▶ Open the "WW\_Programming\_Concepts.WWP" project if necessary.
  1. Close the current project if necessary.
  2. In the home window, click "Tutorial" and select the "Programming concepts" project.
- ▶ Open the "PAGE\_Numeric.WWH" page (double-click the name of the page in the "Project explorer" pane).
- ▶ Run the test of the page. This page gives an overview of the operations that can be performed on the "numeric values".

### Details

A numeric edit control can be initialized:

- with the numeric value directly:

```
EDT_NUM1 = 3.14
```

- with a numeric variable:

```
int1 is int = 1234  
EDT_NUM1 = int1
```

The "+" operator is used to concatenate a string and a numeric value:

```
Info("The integer value is " + EDT_NUM1)
```



Tip

If you run the following code:

```
Info("A calculation: " + 1 + 2)
```

The dialog box will display "A calculation: 12".

To display the result of the calculation, you must use the following code:

```
Info("A calculation: " + (1 + 2))
```

The dialog box will display "A calculation: 3".

### Some examples of numeric operations performed in WLanguage:

- The integer part of a number is returned by **IntegerPart** and the decimal part of a number is returned by **DecimalPart** :

```
// Displays the integer part and the decimal part of EDT_Real
Info("Integer part of "+EDT_Real+": " ...
IntegerPart(EDT_Real), ...
"Decimal part of "+EDT_Real+": " ...
DecimalPart(EDT_Real))
```

- The absolute value of a number is returned by **Abs**:

```
// Displays the absolute value of EDT_SignedReal
Info("Absolute value of " + EDT_SignedReal + ...
" : " + Abs(EDT_SignedReal))
```

- The rounded value of a number is returned by **Round**:

```
// Displays the value of EDT_SignedReal rounded to 1 decimal
// place
Info("Rounded value of " + EDT_SignedReal + ": " + ...
Round(EDT_SignedReal,1))
```

- **Root** is used to calculate the Nth root of a number:

```
// Calculates the square root of EDT_Number
EDT_Square_Root = Root(EDT_Number, 2)
```

- **Power** is used to raise a number to the power of N:

```
// Calculates the power of 2 for EDT_Number
EDT_Power = Power(EDT_Number, 2)
```

### Note: The type of the numeric edit controls

When a control is defined as a numeric control, its type may be undefined (integer, real, double real, ...). Its type is automatically defined according to the mask selected for the control.

To force the type of a control, all you have to do is use a typed variable. For example:

```
Val1 is int
Val1 = 123456789 // Assign the control
EDT_NUM1 = Val1
Val1 = EDT_NUM1 // Retrieve the control
```

## Processing the currencies (server code only)

---

### Practical example

- ▶ Open the "WW\_Programming\_Concepts.WWP" project if necessary.
  1. Close the current project if necessary.
  2. In the home window, click "Tutorial" and select the "Programming concepts" project.
- ▶ Open the "PAGE\_RealCurrency.WWH" page (double-click the name of the page in the "Project explorer" pane).
- ▶ Run the test of the page. This page is used to run the test of the different code sections presented in this paragraph.

### Details

The "Currency" type is a real coded on 10 bytes. It is used to give the solution to the two problems not solved by the reals:

- more than 15 significant digits (the double reals support up to 15 significant digits), the "Currency" type supports 23 significant digits.
- avoid an error resulting from the binary coding of reals:
  - A real supports up to 15 significant digits. In fact, the binary coding of the reals does not allow to code all reals with 15 significant digits.
  - The "Currency" type uses a different system for coding reals that causes no rounding error. For example, the following code returns a false result:

```
// The calculation is false with a Real variable
x is Real
x = 18.6 - 8.6 - 10
Error ("18.6-8.6-10="+x)
```

On the contrary, the result is correct with the following code:

```
x is Currency
x = 18.6 - 8.6 - 10
Info(x)
```

To perform divisions on currencies, we recommend that you use intermediate variables of "Currency" type.

```
x, y are Currencies
x = 12345678901234567.123456
y = 12345678901234567.123456
x = x/y
Info ("12345678901234567.123456", ...
      "/" , "12345678901234567.123456", "= "+x)
```



Note

The **Numeric type** is used to perform advanced calculations. By default, the Numeric type corresponds to 32 digits for the integer part and to 6 digits for the decimal part (like the currency type). The numeric type allows you to configure the number of digits that will be used for the integer part and the number of digits that will be used for the decimal part.

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

### Mixing strings and numeric values

The numeric values and the strings can be mixed together. WebDev is very flexible for assigning variables. For example, a string of digits can be assigned in a numeric variable (and conversely). For example:

```
i is int
c is string
i = 123
c = i // The c variable contains the "123" string
c = "456"
i = c // The i variable contains the value 456
```

To transform a number into a character string while respecting a specific format, all you have to do is use **NumToString**. For example:

```
NumToString(1234.567, "012,3f") // Returns "00001234,567"
```

- ▶ Open the "PAGE\_NumToString.WWH" page and run its test. Double-click its name in the project explorer.



Notes

#### Server and browser code

**Val** is the "reverse" function of **NumToString**. This function is used to convert a string into a numeric value.

## Processing the dates and times (server and browser code)

To easily manage the dates and times in your applications, WebDev includes:

- a **Date, Time or Duration edit control**. With this control, no more hassle to enter a valid date or a valid time.
- a **formatted Date, Time or Duration display control**. With this control, no more problem to display a valid date or time.
- **Date, Time, DateTime and Duration variables**. These variables allow you to easily handle the dates and times by programming and to perform multiple calculations.

## The dates (server code and browser code)

### Practical example

- ▶ Open the "WW\_Programming\_Concepts.WWP" project if necessary.
  1. Close the current project if necessary.
  2. In the home window, click "Tutorial" and select the "Programming concepts" project.
- ▶ Open the "PAGE\_Date.WWH" page. This page illustrates the explanations given in the rest of this lesson.

### Input mask and returned value

The dates (like the times) are strings with a preset format.

In a "Date" (or "Time") edit control, you must distinguish between:

- The **input mask**: this format will be used to display the date and the time.
- The **returned value** (or stored value): this is the value returned by the control to the program.

This information is entered in the description window of the edit control.

- ▶ In the "PAGE\_Date.WWH" page, display the description window of the "Enter a date" control ("Description" from the popup menu). Select the "General" tab in this window. The following information is displayed:

For example, for a "Date" control:

- the input mask will be "DD/MM/YYYY". The date entered by the user will have the following format: "01/11/2011".
- the returned value will be "YYYYMMDD" (Example: the value entered in "23/04/2011" format will return "20110423" to the program).



Note

You have the ability to choose a "System date" input mask. In this case, the format displayed will be the one defined in the "Regional settings of the control panel of Windows" of the computer currently run.

Let's take a look at the following examples to understand the relationships between the input mask and the returned value:

- If the input mask of a date control is "DD/MM/YY" and the returned value is "YYYYMMDD":

```
TDATE="19981225" //displays the date in "25/12/98" format
TDATE="981225" //displays the date in an incorrect format
```

- If the input mask of a date control is "MM/DD/YYYY" and the returned value is "YYMMDD":

```
TDATE="981225" //displays the date in "12/25/2098" format
TDATE="19981225" //displays the date in an incorrect format
```

Therefore, pay great attention to the format of the returned value in a date or time control.

The WLanguage functions that handle:

- the dates use the "YYYYMMDD" format.
- the times use the "HHMMSSCC" format (some of them use the "HHMMSSCCMMM" format to manage the milliseconds).



Notes

**Reminder:** By default, the format (input mask) of the control corresponds to the date mask defined in the "Language" tab of the description window of the project (on the "Project" pane, in the "Project" group, click "Description"). Therefore, the same mask is automatically used in all the date controls of the application.

This feature is also very useful in the multilingual applications.

- ▶ Now let's see how to manage the dates by programming.  
Close (if necessary) the description window of the control and run the test of the "PAGE\_Date" page.

### What is today's date?

To find out today's date, all you have to do is use **Today** (or **DateSys**). **Today** returns the system date of your computer as a character string in "YYYYMMDD" format. Example:

```
Info("Today is " + Today())
```

- ▶ To check this feature in our example, click the "Today" button. You will notice that the date is displayed in "YYYYMMDD" format.

To display the date in a more explicit format, all you have to do is use **DateToString**:

```
Info("Today is " + DateToString(Today(), maskSystemDate))
```

**DateToString** transforms a string in "YYYYMMDD" format into a string in the selected format. The **maskSystemDate** constant allows you to use the date format defined in the "Languages" tab of the project characteristics. Reminder: To display the characteristics of the project, on the "Project" pane, in the "Project" group, click "Description".

► To check this feature in our example, click the "DateToString" button.

### What is today's date, with the day and the month in letters?

To display the date in letters, you must use **DateToString** with a specific date format:

```
Info("Today is " + DateToString(Today(), "DDDD DD MMMM YYYY"))
```

In this code:

- the "DDDD DD" string is used to get the day in letters. For example: Monday 17.
- the "MMMM" string is used to get the month in letters. For example: September.
- the "YYYY" string is used to get the year. For example: 2012

► To check this feature in our example, click the "In letters" button.

### How many days between 2 dates?

You want to find out how many days have passed between two dates? It's quite simple: use **Date-Difference**:

```
NumberDays is int  
NumberDays = DateDifference("20100101", Today())  
InfoBuild("Number of days between %1 and %2: %3", ...  
    DateToString("20100101", maskSystemDate), ...  
    DateToString(Today(), maskSystemDate), NumberDays)
```

In this code, **InfoBuild** is used to display the result. This function is used to build the text displayed according to parameters.

These parameters appear in the string as %1, %2, ...



Notes

### Why not directly build a string with the result of functions?

You also had the ability to build the message content as follows:

```
Info("The number of days between " + ...
    DateToString("20010101", maskSystemdate) + " and " + ...
    DateToString(Today(), maskSystemDate), ...
    NumberDays)
```

This notation presents several drawbacks:

- the readability of the code: a string is easier to read.
- the translation: a string can be entirely translated. The words and the parameters can be reversed.

The translator has a single obligation: include %1, %2, %3.

According to the same principle, WLanguage also proposes **StringBuild** and **ErrorBuild**, ...

- ▶ To check this feature in our example, click the "DateDifference" button.

### What is the day of the week corresponding to a given date?

To find out the day corresponding to a given date, all you have to do is use **DateToDayInAlpha**:

```
Aday is string
Aday = DateToDayInAlpha("17890714")
Info("July 14th, 1789 was a " + Aday)
```

- ▶ To check this feature in our example, click the "DateToDayIn Alpha" button.

### Practical exercise

Now that you are a bit more familiar with the dates: what is the purpose of the following code?

```
DateToString(Today(), "DDDD DD MMM YYYY")
```

Answer: Displays the literal date (the day and the date in everyday's language).

## Handling Date variables (server code)

Specific types of variables are available for the Date, DateTime and Duration values.

**These types of variables are available in server code only.**

These variables can be handled like "strings". Each variable contains a value according to the following table:

| Type     | Default format   |
|----------|------------------|
| Date     | YYYYMMDD         |
| DateTime | YYYYMMDDHHMMSSCC |
| Duration | YYYYMMDDHHMMSSCC |

The "Date" type supports the dates from 01/01/0001 to 31/12/9999 (that should keep us going for a while!).

More seriously, to handle the value of these variables, you have the ability to use the following syntaxes:

```
MyDate is Date = "20111021"  
// Displays the formatted date  
Info(DateToString(MyDate)) //Displays "21/10/2011"  
// Adds a year to MyDate  
MyDate..Year = MyDate..Year + 1  
// Adds a month to MyDate  
MyDate..Month = MyDate..Month + 1  
// Adds a day to MyDate  
MyDate..Day = MyDate..Day + 1  
// Displays the formatted date  
Info(DateToString(MyDate)) //Displays "22/11/2012"
```

In this code, **Year**, **Month** and **Day** are WLanguage properties.

The following syntax allows you to use the WLanguage properties on WebDev elements:

```
<Element Name>..PropertyName
```

See the online help for more details.

- ▶ In the "PAGE\_Date" page, the "Code example" button is used to calculate the date of the next January 1st by using a Date variable. The following code is used:

```
MyDate is Date
// MyDate is automatically initialized to today's date

// Calculate the date of the next January 1st
NewYear is Date
NewDate..Year++
NewYear..Day = 1
NewYear..Month = 1
MyDuration is Duration = NewYear - MyDate

InfoBuild("There are %1 days left before the next midnight party
on New Year's Eve", on %2.", MyDuration..InDays, ...
DateToString(NewYear))
```

## The times (server code and browser code)

### Practical example

- ▶ Open the "WW\_Programming\_Concepts.WWP" project if necessary.
  1. Close the current project if necessary.
  2. In the home window, click "Tutorial" and select the "Programming concepts" project.
- ▶ Open the "PAGE\_Time.WWH" page. This page illustrates the explanations given in the rest of this lesson.
- ▶ Run the test of this page.

### What time is it?

To find out the time, all you have to do is use **Now** (or **TimeSys**). **Now** returns the current time of your system as a character string in "HHMMSSCC" format:

```
Info("It is " + Now())
```

- ▶ To check this feature in our example, click the "Now" button.
- ▶ You want to display the time in "HH:MM:SS:CC" format? The code becomes:

```
Info("It is " + TimeToString(Now()))
```

**TimeToString** transforms a time in "HHMMSSCC" format into a string in "HH:MM:SS:CC" format.

- ▶ To check this feature in our example, click the "TimeToString" button.

If you do not want to display the hundredths of a second, the code is:

```
MyTime is string = TimeToString(Now()), "HH:MM:SS")
Info("It is "+ MyTime)
```

All you have to do is specify the display format with **TimeToString**.

- To check this feature in our example, click the "Without the hundredths" button.

### How much time has passed between two given times?

It is 17:25. How much time has passed since 12:15? The code is as follows:

```
// Displays the time passed since 12:15
Diff is int = TimeDifference("1215", Now())
Result_Time is Time = IntegerToTime(Abs(Diff))

IF SysTime() >= "12150000" THEN
    STC_Result4 = StringBuild(...
        "Time passed since 12:15 -> %1 h %2 min %3 s ",...
        Result_Time..Hour, ...
        Result_Time..Minute, Result_Time..Second)
ELSE
    STC_Result4 = StringBuild(...
        "Time left until 12:15 -> %1 h %2 min %3 s ",...
        Result_Time..Hour, ...
        Result_Time..Minute, Result_Time..Second)
END
```

**TimeDifference** calculates the difference between the current time and 12:15.

**IntegerToTime** transforms an integer (that corresponds to the number of hundredths of a second since midnight (or 00:00)) into a time in "HHMMSSCC" format.

**TimeToInteger** performs the reverse operation.



Caution !

- No ":" must be found in the time passed in parameter to **TimeToInteger** otherwise the result would be incorrect.
- To calculate durations exceeding 24 hours, use the Date type, Time type, ...

- To check this feature in our example, click the "1st example" button.

The same calculation can be performed by using the Time and Duration variables.

Indeed, like for the dates, WLanguage proposes specific variables for the Time, DateTime and Duration values.

**These types of variables are available in server code only.**

These variables can be handled like "strings". Each variable contains a value according to the following table:

| Type     | Default format   |
|----------|------------------|
| Time     | HHMMSSCC         |
| DateTime | YYYYMMDDHHMMSSCC |
| Duration | YYYYMMDDHHMMSSCC |

The WLanguage code becomes:

```
In12H is Hour
In12H..Hour+= 12
In12H..Minute+= 30
InfoBuild("In 12 hours and 30 minutes, it will be %1" , ...
          TimeToString(hIn12H, "HH:MM"))
```

► To check this feature in our example, click the "2nd example" button.

## Calculations with dates and times

---

Several WLanguage functions can be used to manage the dates and times and to perform different calculations.

See the online help (keyword: "Date, WLanguage functions") for more details.



**PART 2**

**Site with data**

**10**

**DEVELOP 10 TIMES FASTER**





# LESSON 2.1. OVERVIEW

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

---

- Overview of the site created in this part



Estimated time: 5 min

## Overview of the site created in this part

---

In this part, we will present the databases that can be handled by WebDev. For educational purpose, we are going to develop an Intranet site from A to Z, from the creation of the analysis to the test of the site. The deployment of the site will be explained in a specific section.

You will see the main points for developing a WebDev site.

The Intranet site that we are going to develop is an online management of a library. You will have the ability to find a book directly from your computer.

The database used is HFSQL Classic, the free database supplied with WebDev. The HFSQL Client/Server database will be presented later in this tutorial.

Let's see what you are going to do first.

The full project of the application is supplied with this tutorial:

1. On the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Tutorial".
2. Select "Site with data (Answer)".
3. Run the test of the project ( among the quick access buttons).
4. Use the application. This application was created by the automatic application generator of WebDev (RAD).
5. Close the browser.

## LESSON 2.2. PROJECT AND ANALYSIS

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

---

- Creating a project
- Creating an analysis



Estimated time: 50 min

## Overview

To create a site with a database, you must:

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

Then, we will create the site via the RAD module (Rapid Application Development).

## Creating the project

► To create the project:

1. Start WebDev (if not already done). Close the current project if necessary.
2. In the welcome window, click the "Create a project" button and double-click "Internet site". The wizard for project creation starts. The different wizard planes help you create your project. The information specified in this wizard can be modified later.

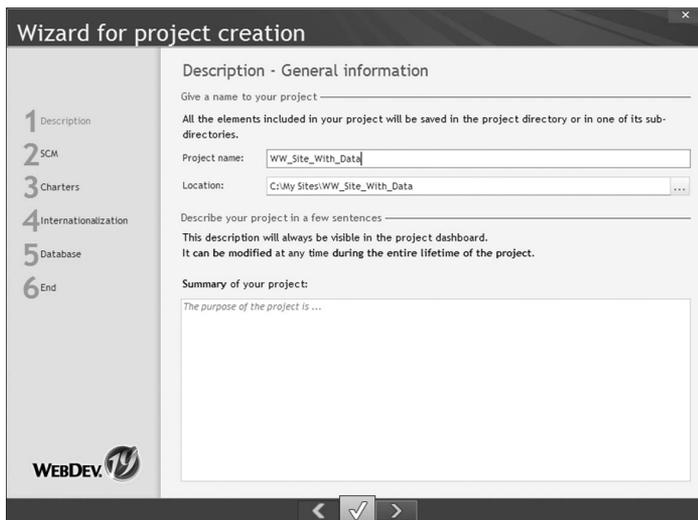


Notes

Tip: To create a project, you can also:

1. Click  among the quick access buttons of the WebDev menu.
2. Click "Project" in the wheel that is displayed.

3. The first wizard screen is used to enter the name of the project, its location and its description. In our case, this project will be named "WW\_Site\_With\_Data".



By default, WebDev proposes to create this project in the "\\My Sites\WW\_Site\_With\_Data" directory. You can keep this location or modify it via the [...] button.



Notes

If you are using an Apache server, no accented characters should be used in the project directory. Indeed, the use of accented characters may prevent the Apache server from restarting after its configuration for the new project.

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

**4.** The different wizard steps are specified on the left of the wizard. These steps can be clicked directly. The other screens of step 1 ("Description") are not fundamental, so click "3 Charters" directly.

**5.** We will be using the standard programming charter. Go to the next wizard screen (click the yellow arrow oriented to the right).

**6.** The style book is used to define the "look" of the site. Choose the "iStyle" skin. Expand the "Color palette" button and select the "Gentle & Light" palette.



Notes

The skins are used to standardize the visual aspect of a site and they allow you to easily and quickly change style.

Go to the next wizard screen.

**7.** We are now going to specify the information regarding the database. Click "5 Database".

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



Notes

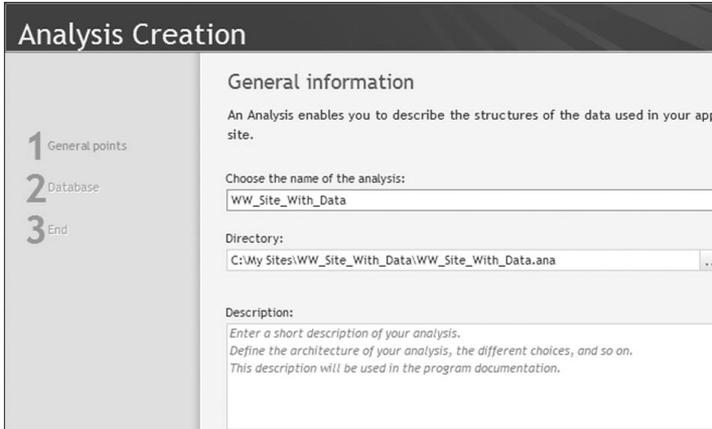
To better understand the lessons found in this section and to optimize your training, we advise you to create the "WW\_Site\_With\_Data" project.

A corrected example can be access at any time to check the validity of the operations performed. This corrected example is available from the welcome window of WebDev.

In the home window, click "Tutorial" and select the "Site with data (Answer)" project.

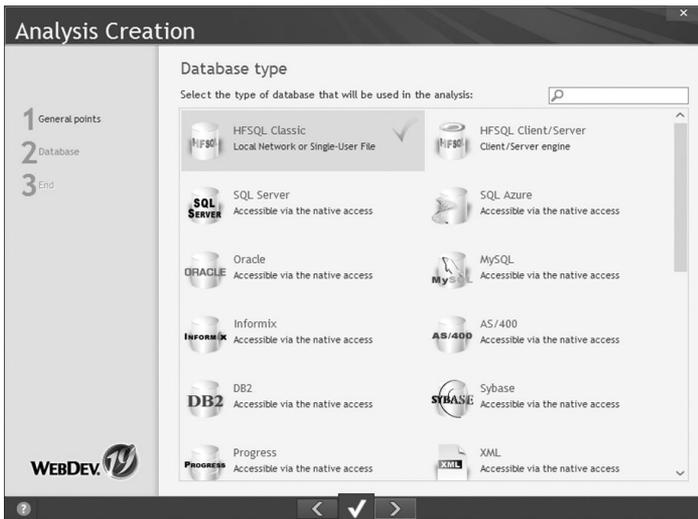
## Creating the analysis

- The steps of the wizard for analysis creation are as follows:
1. Specify the name and directory of the analysis.



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

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



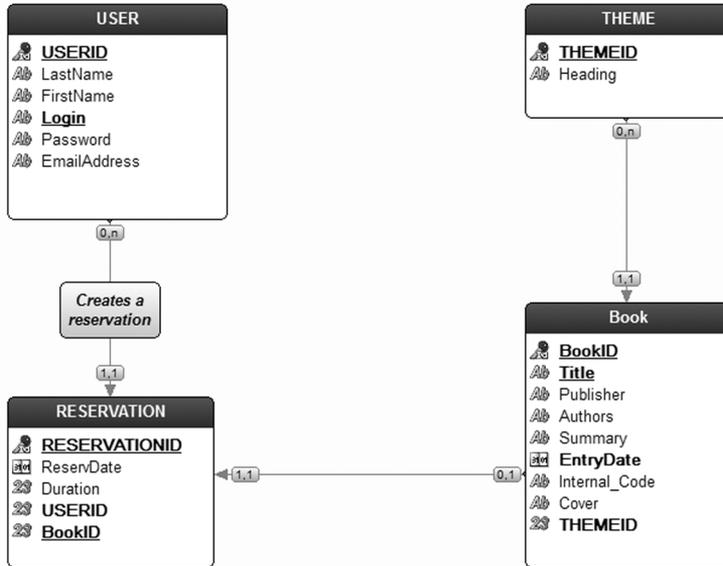
Go to the next wizard screen.

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

## Creating the description of the data files

Our project for library management will be associated with an analysis. This analysis includes the following data files:

- BOOK
- RESERVATION
- THEME
- USER



To create the data files of this analysis, we are going to use the different methods proposed by WebDev.

## Creating a data file and its items in the editor

- The steps of the wizard for creating a data file are as follows:
1. In the wizard, select "Create a new description of data file". Go to the next wizard screen.
  2. We are going to create the data file named "USER". Its name is "USER"

This name will be used:

- to handle the data file by programming. The variable associated with the data file will be "User".

- to build the name of the associated physical data file (USER.fic file).

The caption and the representation of the record are automatically displayed.



Notes

In the wizard, the "A record represents" control indicates the information that will be used to describe the links between the data files. This information must be specified with great care!

3. Keep "Automatic identifier on 8 bytes" (mandatory for the server replication).

This option indicates whether the data file must include a unique key, automatically managed by WebDev.



Notes

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

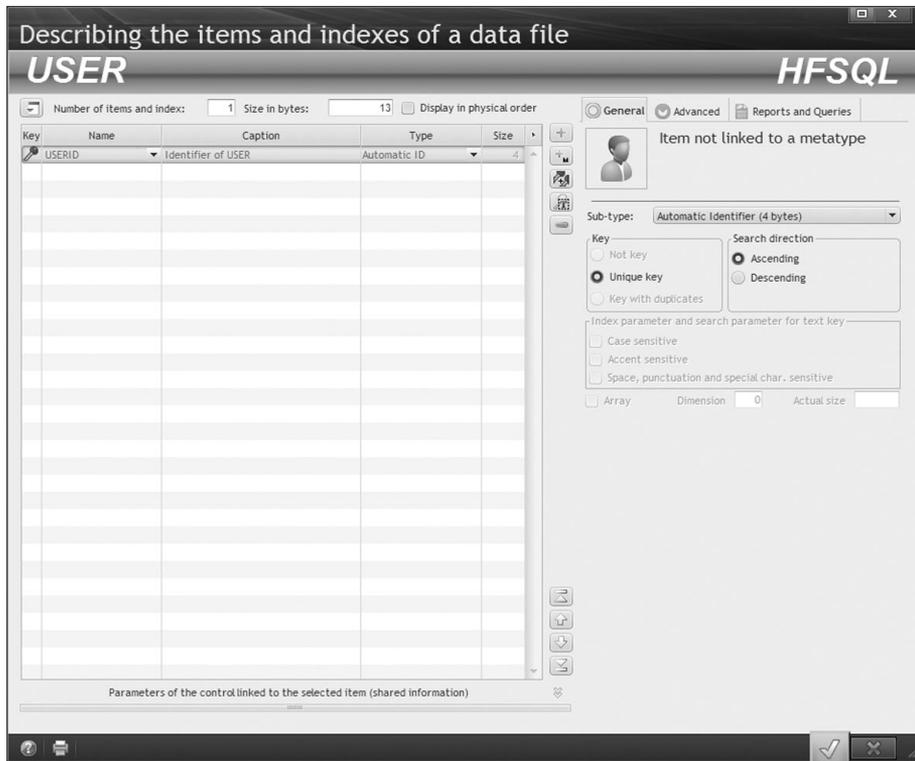
This identifier is automatically managed by WebDev. Whenever a record is added into the data file, WebDev automatically assigns a value to the identifier of the file. This value is unique.

You can select "None" if no automatic identifier is required (if no unique key is required or if a unique key already exists in the data file).

4. Go to the next screen and select the type of database associated with the data file. We are going to work on HFSQL Classic data files. Select "HFSQL Classic" and go to the next screen.

5. Click the green button to validate. The data file is automatically created in the analysis.

The description window of items is opened.



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

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

- ▶ First, we are going to create the "LastName" item. This item will contain the last name of the user.
  1. In the description window of items, click the "Name" column of the first empty row. This column automatically becomes editable. Enter "LastName".
  2. Click the "Caption" column. The name of the item is automatically displayed. We are going to modify the caption of the item by typing "Last name of the user". In the "Type" column, the "Text" type is automatically selected. Don't change anything.

- ▶ Position on the next empty row and create the "FirstName" item. This item will contain the first name of the user.
- ▶ We are now going to create the "Login" item. This item will contain the identifier of the user.
  1. In the description window of items, click the "Name" column of the first empty row. This column automatically becomes editable. Enter "Login".
  2. We are going to modify the size of the item. Click the "50" box and replace "50" by "10". Click the next line. The values are automatically updated.

| Key | Name      | Caption                | Type         | Size |
|-----|-----------|------------------------|--------------|------|
|     | USERID    | Identifier of USER     | Automatic ID | 4    |
|     | LastName  | Last name of the user  | Text         | 50   |
|     | FirstName | First name of the user | Text         | 50   |
|     | Login     | Login                  | Text         | 10   |
|     |           |                        |              |      |

3. This item will be a key item in our data file: the keys are used to improve the speed for accessing the data and to simplify the browse operations performed on the data files. You will have the ability to perform searches or browses on this item.



Notes

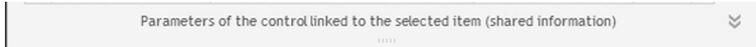
The concept of key is part of the characteristics of an item. When creating an item, you have the ability to specify whether it is:

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

4. To define the key, you must: reselect the row corresponding to the "Login" item in order to enable the description controls found on the right of the screen. Then, all you have to do is specify the type of key used. In our case, the login is a unique key.

|  |  |
|--|--|
| .Key<br><input type="radio"/> Not key<br><input checked="" type="radio"/> Unique key<br><input type="radio"/> Key with duplicates  | Search direction<br><input checked="" type="radio"/> Ascending<br><input type="radio"/> Descending |
| Index parameter and search parameter for text key<br><input type="checkbox"/> Case sensitive<br><input type="checkbox"/> Accent sensitive<br><input type="checkbox"/> Space, punctuation and special char. sensitive |  |

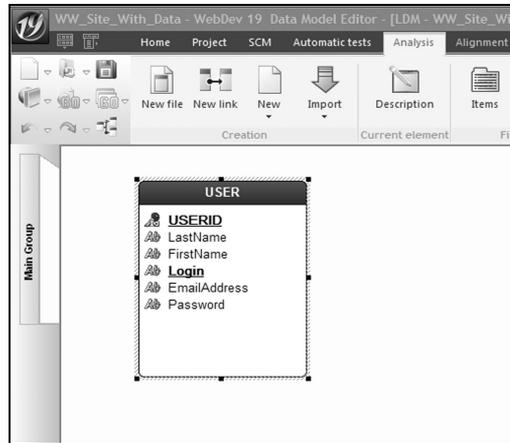
- ▶ We are now going to create the "Password" item.
  1. Position yourself on a new table row. Enter:
    - the name: Password
    - the caption: Password associated with the login
    - the type: Text
    - the size: 10
  2. In the lower section of the screen, click the double arrow to display the parameters of the control linked to the selected item.



The information entered here will be automatically used when creating the pages linked to the data file. You will find here the type of the control and the caption. We are going to modify the type of the control by choosing the "Password" type. Therefore, when the password is typed in the site, the characters will be automatically hidden.

3. Click the double arrow again.
  4. The password of a user being a sensitive information, it will only be displayed in the form pages (the page for editing the user for example). To do so, select the "Advanced" tab on the right of the screen and specify that the RAD must generate a control in form mode for this item only (uncheck "a column in table mode" and "a control for the reports").
- ▶ We are going to create an email item. Easy!
    1. Position yourself on a new table row. Enter:
      - the name: EmailAddress
      - the caption: Email address
      - the type: Text
      - the size: 260
    2. That's it, the USER file is successfully described.
  - ▶ Click the green button to close the description window of items.
  - ▶ Select "Go back to the WebDev editor" and validate. Close the window that is displayed.

The USER file is displayed in the data model editor.

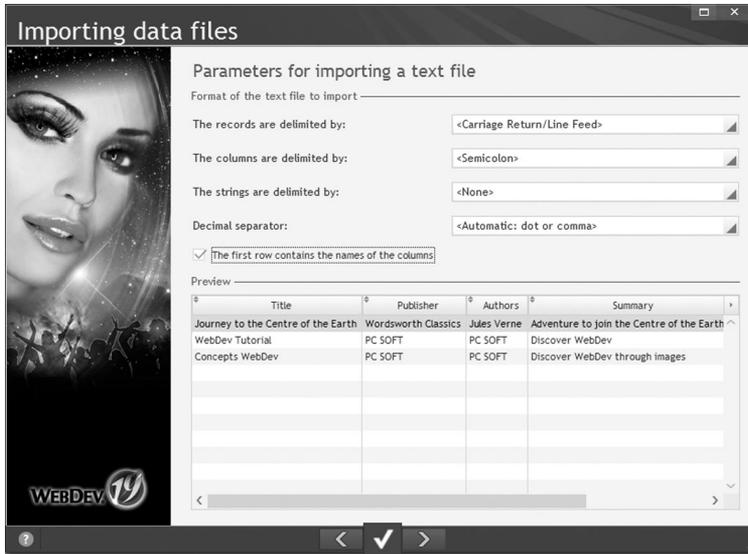


### Importing a CSV file

To create the BOOK data file (containing the characteristics of the books), we will be using another method: importing a CSV file. \xcb from the CSV file containing the data, WebDev will create the description of the data file in the analysis as well as the corresponding HFSQL data file with the data found in the CSV file.

- ▶ To import a CSV file into the analysis:
  1. In the file explorer of Windows, open the following WebDev sub-directory: "\Tutorial\Exercises\WW\_Site\_With\_Data".
  2. Select the "Book.csv" file.
  3. Drag and Drop the "Book.csv" file to the data model editor of WebDev. The wizard for importing the data files starts.
  4. We are going to convert the content of the CSV file to the HFSQL format. Go to the next screen.
  5. Select the format of the files to import. Select "Text file". Go to the next wizard screen.
  6. WebDev indicates the path of the file to import. Go to the next wizard screen.

7. Specify the following import parameters:



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

8. Go to the next screen.

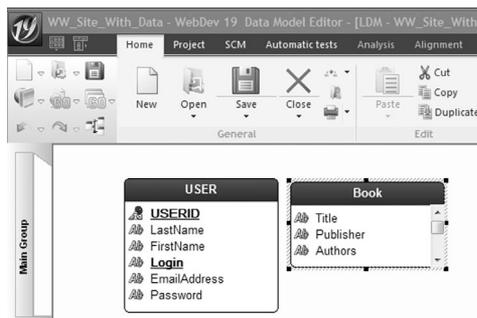
9. The structure of the data file that will be created is displayed. We are going to perform some modifications regarding the sizes of the created items:

| Item          | Type | Size |
|---------------|------|------|
| Title         | Text | 50   |
| Publisher     | Text | 50   |
| Authors       | Text | 50   |
| Summary       | Text | 50   |
| EntryDate     | Date | 8    |
| Internal_Code | Text | 10   |

10. Go to the next screen.

11. The wizard indicates the creation directory of the HFSQL file created with the data found in the imported CSV file. Go to the next wizard screen.

12. Validate the wizard. The data file appears in the analysis.



- ▶ Let's see the description of the imported data file:
  1. Select the "Book" data file and select "Description of data file" from the popup menu.
  2. Modify the caption of the file: delete "(Imported)".
  3. Click the icon  to display the description of the items found in the data file.
  4. This data file contains no automatic identifier. We are going to create it:
    - Create a new item in the "Book" file.
    - This item is named "BookID" and its caption is "Identifier of Book". The type of this item is "Automatic ID".
    - Move this item to first position in the list of items via the arrow buttons (found at the bottom of the list of items).
  5. We are going to modify some information:

| Item      | Characteristic to modify                                       |
|-----------|--|
| Title     | Unique key   |
| Summary   | Name and caption: Summary<br>Sub-type: Text memo               |
| EntryDate | Type: Date<br>Sub-type: Date (yyyymmdd)<br>Key with duplicates |

We get the following data:

Describing the items and indexes of a data file

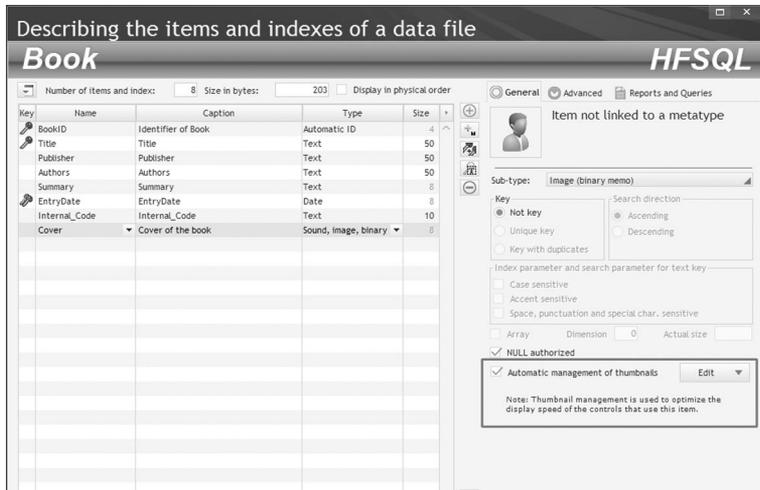
## Book

Number of items and index: 7 Size in bytes: 194  Display in physical order

| Key   | Name          | Caption            | Type         | Size |
|---|---------------|--------------------|--------------|------|
|  | BookID        | Identifier of Book | Automatic ID | 4    |
|  | Title         | Title              | Text         | 50   |
|   | Publisher     | Publisher          | Text         | 50   |
|   | Authors       | Authors            | Text         | 50   |
|   | Summary       | Summary            | Text         | 8    |
|  | EntryDate     | EntryDate          | Date         | 8    |
|   | Internal_Code | Internal_Code      | Text         | 10   |

- ▶ To allow the Web user to get additional information about the book, we are going to create a "Cover" item. This item will contain the image found on the cover of the book.
  1. Create a new item in the "Book" file.
  2. This item is named "Cover" and its caption is "Cover of the book". This item is a "Sound, Image, Binary" item.

### 3. Check "Automatic management of thumbnails".



#### Notes

The management of thumbnails is used to optimize the display of controls linked to the item.

4. Validate the description of items.
5. Validate the description of the data file.

## Importing files from the dictionary

To simplify the creation of the data files in the analysis, a file description was prepared and included in a dictionary beforehand. Let's see how to import this description. A specific WebDev pane, the "Dictionary" pane, allows you to easily handle the dictionaries.

- ▶ To import elements from a dictionary:
  1. Display the "Dictionary" pane if necessary: on the "Home" pane, in the "Environment" group, expand "Panels" and select "Dictionary".
  2. Click the pane and select "Open" from the popup menu.
  3. Select the "Dictio\_TUT.dic" file found in the following WebDev sub-directory: "\\Tutorial\Exercises\WW\_Site\_With\_Data". The dictionary appears in the pane. This dictionary contains the "RESERVATION" file.
  4. Drag and Drop the "RESERVATION" file to the data model editor.
  5. The editor asks whether a subscription is required. Answer "No".
  6. Validate. The description of the "RESERVATION" data file is included in the analysis.

## Direct import of existing data files

The last method for creating data files consists in importing the existing HFSQL data files. The last file was prepared in this format.

- ▶ To import HFSQL data files:
  1. In the file explorer of Windows, open the following WebDev sub-directory: "\\Tutorial\\Exercises\\WW\_Site\_With\_Data".
  2. Select the "THEME.fic" file.
  3. Drag and Drop the "THEME" file to the data model editor of WebDev.
  4. The import wizard starts. Validate the different planes. The data file appears in the data model editor.
  5. To handle the data of the THEME file, copy the THEME.fic file and the THEME.ndx file (found in the "\\Tutorial\\Exercises\\WW\_Site\_With\_Data" directory) into the EXE sub-directory of your project.

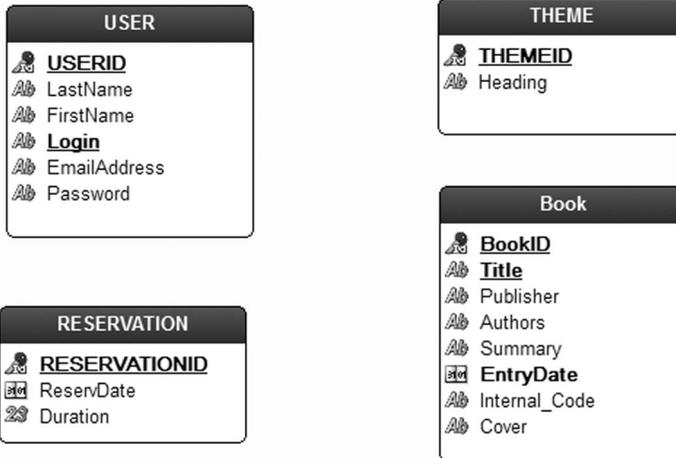
**Notes**

To open the directory of the current project in WebDev, click  found on the "Home" pane, in the "General" group.

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

## Creating the links

All the descriptions of data files required by the site for library management have been created.



We are now going to create the links between the data files.

- ▶ Let's create the link between the "USER" file and the "RESERVATION" file.
  1. On the "Analysis" pane, in the "Creation" group, click "New link". The mouse cursor turns into a pen.
  2. Click the "USER" file then click the "RESERVATION" file.
  3. The wizard for link creation starts.
  4. Answer the questions asked by the wizard:

The screenshot shows the "Describing a link" wizard window. It displays a relationship diagram between the "USER" and "RESERVATION" data files. The relationship is defined as 1 to n (one user has many reservations). Below the diagram, there are four questions with radio button options:

- Each "user" has at least one "reservation":  Yes,  No
- Each "reservation" has at least one "user":  Yes,  No
- Each "user" can have several "reservations":  Yes,  No
- Each "reservation" can have several "users":  Yes,  No

There is also a checkbox for "Display the advanced cardinalities" which is unchecked. At the bottom, there is a "Caption:" field and navigation buttons (back, confirm, forward).

- Each user has at least one reservation: No
- Each user can have several reservations: Yes
- Each reservation has at least one user: Yes

- Each reservation can have several users: No

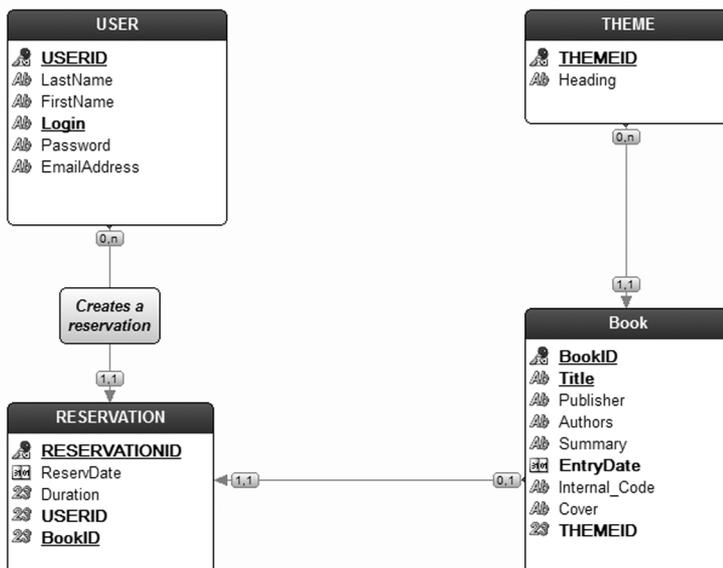


**Notes** You also have the ability to directly enter the cardinalities of the link in the wizard.

5. Specify a caption: "Creates a reservation".
6. Go to the next screen. The wizard automatically proposes the key used by the link (USERID). Display the next wizard screen.
7. The wizard proposes to create a new key in the "RESERVATION" file. Accept this option by going to the next screen.
8. Validate the integrity rules by going to the next screen.
9. Click the green arrow. The link is automatically created in the data model editor.

- ▶ Similarly, create a link between the "THEME" file and the "Book" file. The characteristics of this link are as follows:
  - Source file: THEME
  - Linked file: Book
  - Cardinality: (0, n) - (1,1)
- ▶ Then, create a link between the "Book" file and the "RESERVATION" file. The characteristics of this link are as follows:
  - Source file: Book
  - Linked file: RESERVATION
  - Cardinality: (0, 1) - (1,1)

The analysis is as follows:



## Generation of the analysis

All the data files of our application have been described. We are now going to prepare the development of our site.

The first step consists in generating the analysis.

Generating the analysis is used to make the information about data files available to the other modules of the project. These data files can be handled in the programs, the editors, ...

Without generation, even though the description of the data file exists, you would not be able to use the data file in your programs.

When generating the analysis, all the modifications performed on the analysis and on the data files will be automatically applied to the entire project (pages, linked controls, reports, ...).



Caution!

The generation must be performed whenever you want the modifications performed in the analysis to be taken into account in the programs that use this analysis.

If the analysis is modified several times without any programming between each modification, there is no need to generate the analysis at each modification. Generate the analysis when all the modifications have been made and when you start programming.

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

In this part, we have modified the descriptions of the data files found in the analysis (Book file, User file, ...).

To update the data files of the project, WebDev automatically starts the procedure for modifying the data files. This operation is used to update the data files (".fic" files) according to their description in the analysis.

- ▶ The wizard for automatic modification starts.
  1. Keep the selected option and go to the next screen.
  2. Validate the search location: the physical data files are sought in the EXE directory of your project.
  3. Select all the proposed data files (including the THEME file). Go to the next screen to validate.
  4. Validate the different planes until the automatic data modification is performed.

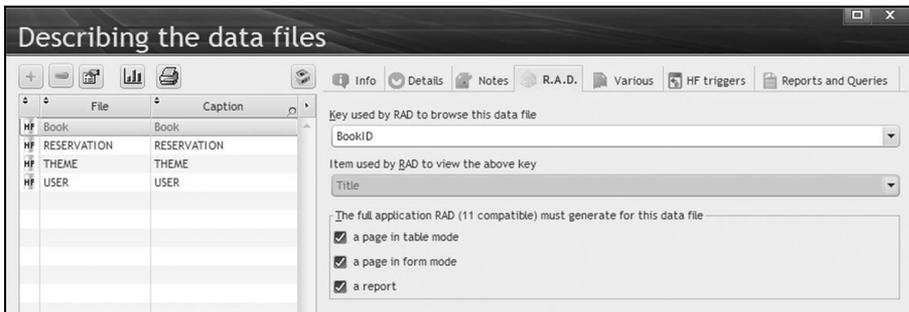
## Configuring the analysis for RAD

When generating the site, the pages and the reports, the RAD module (Rapid Application Development) uses the options specified in the analysis. We recommend that you spend a little bit of time to configure these options in order to get the best possible result. The elements used by RAD are defined:

- in the description of the data files ("RAD" tab).
- in the description of the items ("RAD" tab).
- in the shared information specified for each item.

### Configuring the RAD in the description of the data files

- ▶ To configure the RAD options of the data files:
  1. Display the description of the data files: on the "Analysis" pane, click the group button of the "File" group (  ).
  2. Select the "Book" file in the list on the left and display the "RAD" tab.



The following information must be configured *for each data file*:

- The key used by RAD to browse the data file.
- The item to display for the viewing.
- The elements generated by RAD for the data file.

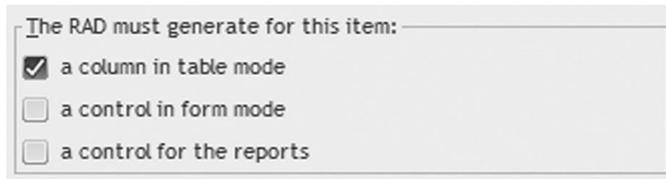
The table below summarizes the configuration to perform:

| <i>Data file</i> | <i>Key used</i> | <i>Item to view</i> | <i>Element to generate</i> |
|------------------|-----------------|---------------------|----------------------------|
| Book             | BookID          | Title               | Table, Form, Report        |
| RESERVATION      | BOOKINGID       |                     | Table, Form, Report        |
| THEME            | THEMEID         | Heading             | Table, Form, Report        |
| USER             | USERID          | LastName            | Table, Form, Report        |

- ▶ To perform these modifications:
  1. Select the file in the table.
  2. Perform the modifications in the RAD tab.
  3. Click the "Validate" button (this step is optional).
  4. Go to the next file.
- ▶ At the end of these modifications, validate the description window of the data files and validate (if necessary) the backup of modifications.

## Configuring the RAD in the description of items

- ▶ To configure the RAD options of items:
  1. For each file, display the description of the items ("Item description" from the popup menu).
  2. Display the "Advanced" tab.
  3. Configure the RAD options.
 For each file, *the identifiers of the files* must have the following characteristics:



The RAD must generate for this item:

- a column in table mode
- a control in form mode
- a control for the reports

Indeed, the identifier must not be displayed in the pages in form mode and in the reports. Keep everything checked for the other items.

4. Validate the description of the file items.

These operations must be performed for:

- The BookID item of Book file
- The RESERVATIONID item of RESERVATION file
- The THEMEID item of THEME file
- The USERID item of USER file

### Special case: BookID in "RESERVATION" file and THEMEID in "Book" file

By default, when the link was created between RESERVATION and BOOK, the identifier of the book was copied into the RESERVATION file.

However, when the Web user wants to enter a reservation, he cannot enter the identifier of the book (that has no significance to him).

- ▶ Therefore, the edit control associated with the BookID item will be replaced by a combo box used to list the different books.
  1. Open the description of the items found in the RESERVATION file.
  2. Display the shared information of the BookID item (button  at the bottom of the screen). This item is linked to an edit control by default.
  3. Select a "Combo box" control.
  4. In the "Content" tab of the shared information, we are going to specify the mode used to fill the combo box:
    - Source for the fill operation: select "File" then "Loaded in memory".
    - Browsed file: Book
    - Displayed item: Title
    - Stored item and search item: BookID
  5. Validate the description of the data file.

- ▶ The same problem occurs between the "THEME" file and the "Book" file. Therefore, the edit control associated with the THEMEID item will be replaced by a combo box used to list the different themes.
  1. Open the description of the items found in the Book file.
  2. Display the shared information of the THEMEID item (  at the bottom of the screen). By default, this item is linked to an edit control.
  3. Select a "Combo box" control.
  4. In the "Content" tab of the shared information, we are going to specify the mode used to fill the combo box:
    - Source for the fill operation: select "File" then "Loaded in memory".
    - Browsed file: THEME
    - Displayed item: Heading
    - Stored item and search item: THEMEID
  5. Validate the description of the data file.
  
- ▶ To take these modifications into account in our project, we are going to regenerate the analysis.

On the "Analysis" pane, in the "Analysis" group, click "Generation".

You can cancel the wizard for automatic data modification that was started at the end of generation.
  
- ▶ Close the data model editor. Let's start programming now.

## LESSON 2.3. THE FULL RAD

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

---

- What is the RAD?
- Generating RAD
- Test of the project



Estimated time: 20 min

## What is the RAD?

---

R.A.D. stands for "Rapid Application Development".

The RAD is used to automatically build an application, which means all the necessary pages, reports and queries.

As already seen in a previous lesson, to develop a site in WebDev, a project and an analysis (if necessary) must be created beforehand. The analysis contains the definition of the structures of the data files handled in the processes.

The RAD module of WebDev uses this analysis. The RAD module contains a wizard allowing you to specify the parameters required to develop the project.

All the generated pages, reports, queries and source code can be customized. You also have the ability to modify the types of controls, the default values, ...

The RAD can also be used to generate several types of pages, it's the page RAD.

Let's see how to use the Project RAD module.



Note

### RID (Rapid Interface Design)

WebDev can also be used to generate pages containing the controls linked to the analysis items only. The entire code required for these pages to operate must be written by the developer.

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

## Generating RAD

---

► To start generating RAD:

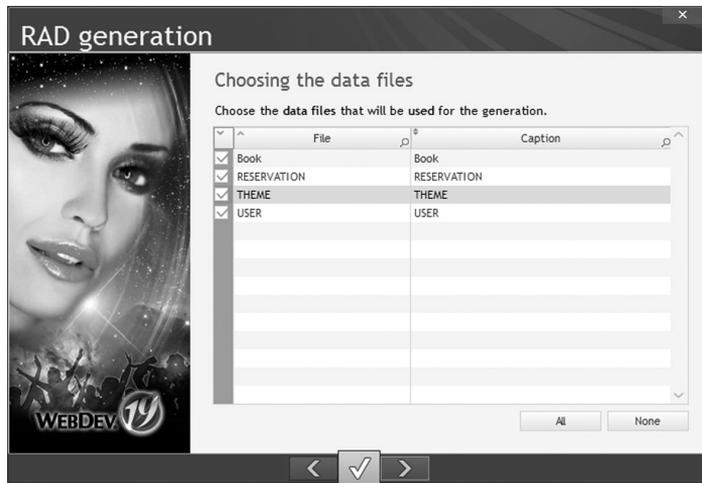
1. On the "Project" pane, in the "Generation" group, click "Full Application RAD". The wizard for generating the RAD application starts.
2. Select the template that will be used for the RAD generation: "Classic RAD (Intranet/Extranet)" for example. Go to the next wizard screen.



Note

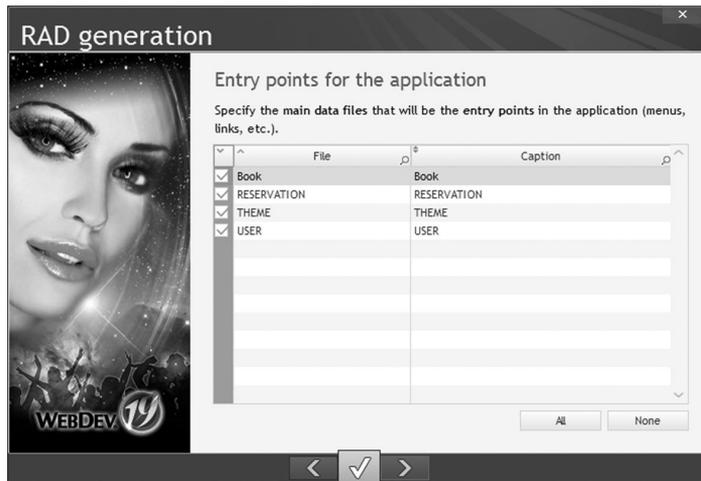
This step is not required if you are using WebDev 64 bits. The "Classic RAD (Intranet/Extranet)" template is automatically selected.

3. All the data files found in the analysis will be taken into account.



Display the next screen.

4. The entry points of the application correspond to the entries available in the menu.



Keep the proposed data files. Display the next screen.

5. Validate. The site is generated and its test can be run.

## Running the test of the site

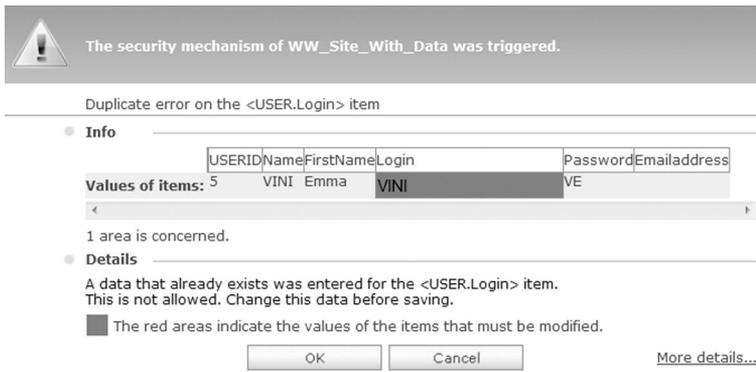
Let's now run the test of the generated site.

► To run the test of the site:

1. Click . The site is started.
2. Select "USER .. Display the list of USER".
3. The list of users is displayed. This list is empty.
4. Click the "New" button. The form for entering a new user is displayed.
5. Specify:
  - the last name of the user: MORGAN
  - the first name of the user: Jack
  - the login: MORGANJ
  - the password: MJ
6. Click the "Validate" button.
7. The user appears in the list.
8. Enter the following users:

| Last name | First name | Login  | Password |
|-----------|------------|--------|----------|
| MORGAN    | Franck     | MORGAN | MF       |
| TASTE     | Flora      | TASTE  | TF       |
| VINI      | Hans       | VINI   | VH       |
| VINI      | Emma       | VINI   | VE       |

9. A specific window is displayed when validating the last record:



This dialog box signals the existence of a duplicate: indeed, the login (that is a unique key) is identical for two persons. This dialog box is used to modify the value of the login: enter "EMMA" and validate.

This page is one of the pages for automatic management of HFSQL errors. We will see later in this tutorial how to manage by programming the different types of errors that may occur on the databases.

10. Select "Book .. Display the list of Book".

11. The list of books is displayed. This list contains the records found in the imported file.

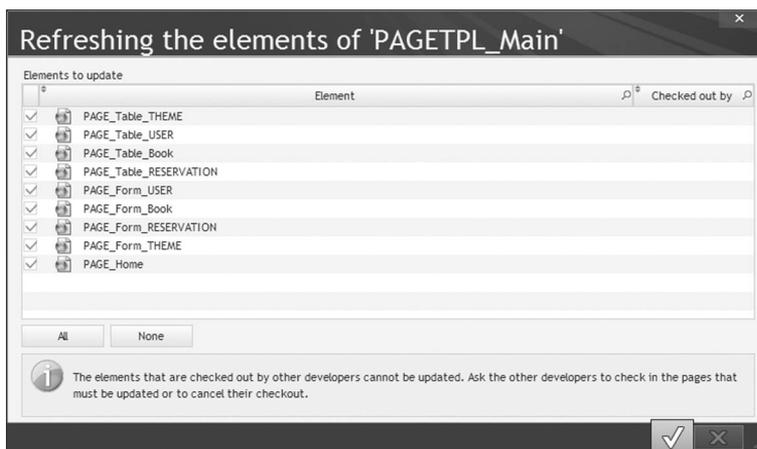
## Customizing the generated site

During the test of the site, you have noticed that the sentence "<Specify the name of the site>" was displayed on all the pages.

We will customize the site by modifying this text. Don't worry, no need to spend hours to perform this operation: all the pages of the site will be modified in a single operation.

The different pages generated by RAD use a page template. This template is used to define the interface common to all the pages.

- ▶ Display a page of the site in the WebDev editor (the PAGE\_Form\_Book page for example). To display the page, all you have to do is double-click its name in the project explorer.
- ▶ You will notice that "<Specify the name of the site>" is displayed on this page. We are now going to modify it.
  1. Display the popup menu (right mouse click) of "<Specify the name of the site>".
  2. Select "Open the template" from the popup menu.
  3. A new page is displayed in the editor. This page contains the controls used for the template of the site pages. These controls are common for all the pages of the site. This page is enclosed by a yellow line (to indicate that it is a template).
  4. Select the static control named "<Specify the name of the site>" and display the control description ("Description" from the popup menu). Replace the text of the static control by "My Library" and validate. The static control is immediately modified.
  5. Save your modifications (CTRL + S or  among the quick access buttons).
  6. To update the pages that use this template, on the "Page" pane, expand "Templates" and select "Update the associated pages".
  7. WebDev displays the list of pages to update.



Validate. The pages are updated.

8. Close the page template displayed in the editor.
9. Run the test of the site. The modification was performed on all the pages.

## LESSON 2.4. EDIT PAGE

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

---

- Creating an edit page
- Managing the addition of a record
- Managing the search and the modification

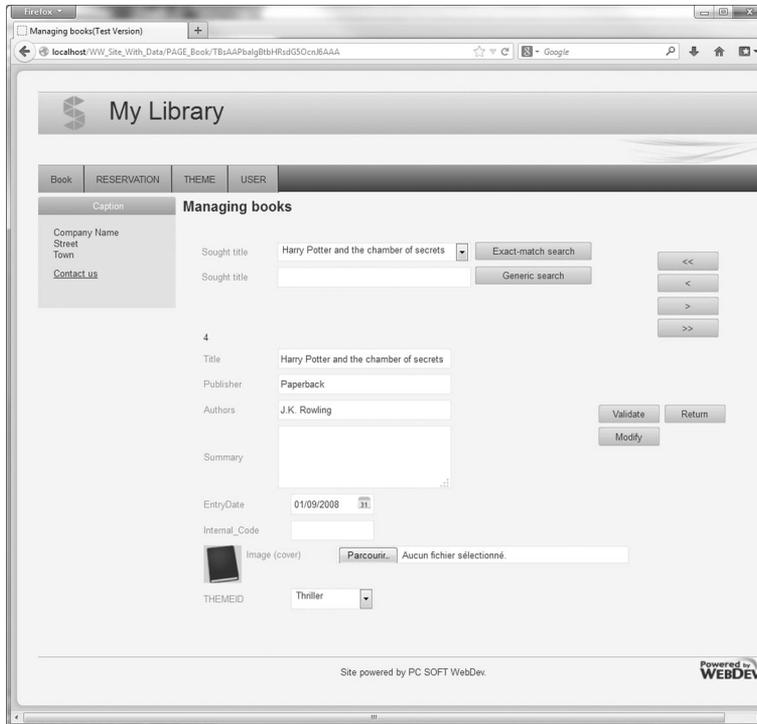


Estimated time: 50 min

## Overview

We have seen the different pages automatically created by RAD. We are now going to create the pages used to manage the addition and the modification of records. These operations will allow you to discover several topics regarding the management of data files and they will also enable you to use some WebDev features.

We are going to create a page used to manage the input, the search and the modification of books. This page is as follows:



## Creating an edit page

To create the edit page on the "Book" file, we could use the wizard for creating RAD pages and create a "Form" page.

But we want to create this page from a blank page: you will see all the steps required to program such page.

► To create an edit page:

1. Open (if necessary) the "WW\_Site\_With\_Data" project that was created in the previous lessons.
2. Create a new blank page. Click  among the quick access buttons. Click "Page" in the wheel that is displayed.

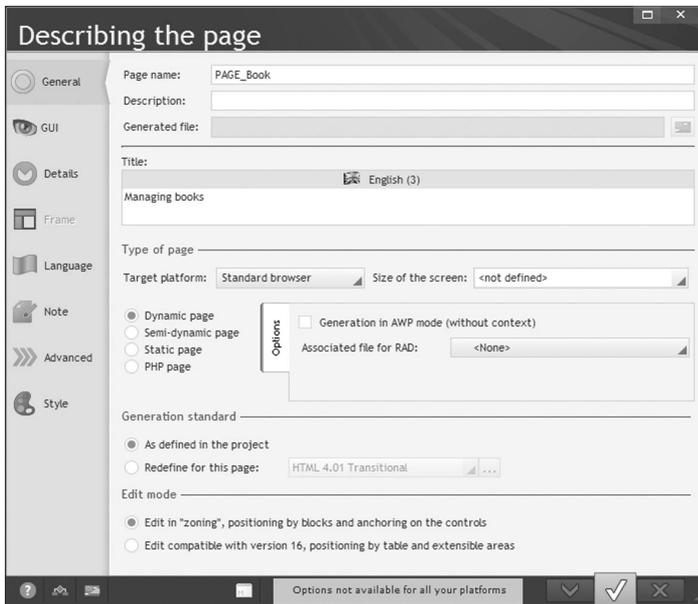
3. In the wizard for page creation, select "Page" on the left of the window then select the "PAGETPL\_Main" template.

4. Validate.

The new page is displayed in the editor.

5. Display the description window of the page. To do so, click the page, display the popup menu (right mouse click) and select "Description" from the popup menu. In the description window, specify:

- the name of the page: "PAGE\_Book".
- the title and description of the page: "Managing books".



6. Validate the description window and save the page.

7. The different data files described in the analysis are listed in the project explorer (last item named "WW\_Site\_With\_Data").

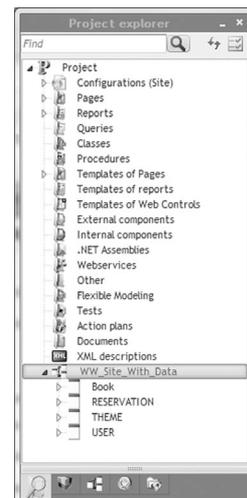
8. Click the "Book" file: the items found in the data file are listed in the bottom section of the project explorer.



Note

If the lower part of the project explorer is not displayed, select the splitter to display it (- - -).

9. Click one of the items found in the bottom section of the project explorer and press [CTRL] + A: all the items are selected in the project explorer.



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

11. Different controls are automatically created in the page. These controls are automatically linked to the item in the data file.

To check this, all you have to do is display the "Link" tab found in the description of one of the controls ("Description" from the popup menu).



12. Close the description window of the control.

13. Save the page.

► We are going to display a title in this page. By default, the title is "Page title". This title being used by the template of the site pages, it cannot be modified directly. To modify this title in the current page, we must overload this control in our page to display "Managing books" instead.

1. Click the "Page title" static control and display the popup menu.

2. Select "Overload the control".

3. Modify the caption of the control: the new caption is "Managing books". This caption can be modified in the page or in the description window of the control.



Note

The controls belonging to a page template and not overloaded are identified by a yellow square (displayed in the top left corner of the control).

The controls belonging to a page template and overloaded are identified by a blue square (displayed in the top left corner of the control).

4. Save the page and run the test of the page (  among the quick access buttons).

## Managing the addition of a record

We are now going to manage the addition of a record into our page. At the moment, the page contains controls linked to the data file but we want to enter information in these controls and we want to save this information in the "Book" data file.

We are going to add two buttons into our page:

- a "Validate" button that will be used to save the data
- a "Return" button that will be used to exit from the page without saving.

### Creating the addition button

► To create the addition button:

1. Create a new button: on the "Creation" pane, in the "Usual controls" group, click .
2. The caption of this button is "Validate" and its name is "BTN\_Validate".

► To enter the code of the button:

1. Display the code of the button (select the button and press F2).
2. Enter the following code in the "Click code (Server)" process:

```
ScreenToFile()
HAdd(Book)
```

Let's take a look at this code:

- **ScreenToFile** is used to initialize the items with the values of linked controls. You also have the ability to use **PageToFile** (identical function). This function is equivalent to the following code lines:

```
Book.Title = EDT_Title
Book.Publisher = EDT_Publisher
Book.Authors = EDT_Authors
...
```

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

- **HAdd** adds the record into the data file. This function takes the values in memory and writes the content of the file items into the file itself. The indexes are automatically updated.

3. Save your page (  among the quick access buttons).

► Run the test of the page (  among the quick access buttons). Fill the controls and click the "Validate" button.

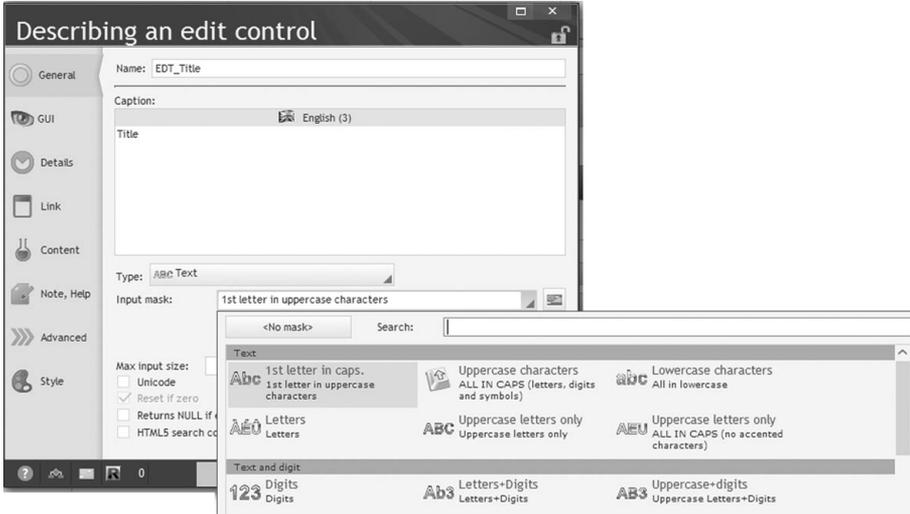
Several remarks:

- No mask is used. You have the ability to enter uppercase and lowercase characters in the different controls.
- The controls are not reset after the addition.
- No closing button: the browser must be closed.

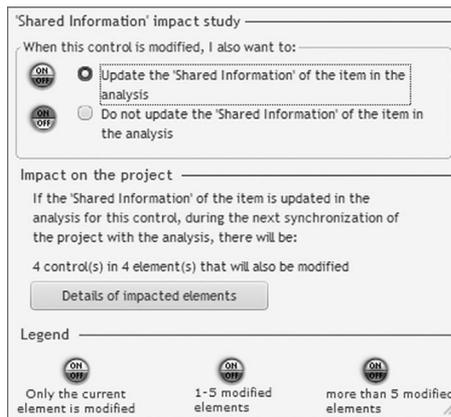
We are going to perform the necessary modifications.

## Adding an input mask

- ▶ First of all, the input mask. An input mask will be defined for the "Title" control.
  1. Display the description window of the "Title" control ("Description" from the popup menu).
  2. In the "General" tab, select the "1st letter in uppercase" mask.



3. Once the mask was modified, the style of the validation button found in the description window changes. An "On/Off" button is displayed. The following screen is displayed when you click this button:



This screen is used to manage the modification of the shared information in the analysis (the information regarding the control associated with the Title item). If the shared information is modified in the analysis, this modification will be applied to all the controls linked to the item during the next analysis generation.

4. Validate the description window of the control.

- ▶ Similarly, define an input mask for the "EDT\_EntryDate" control. The format of this mask is "MM/DD/YYYY". The format of the returned value is "YYYYMMDD". Don't forget to select "Display a calendar button". The date input will be simplified. If necessary, adapt the size of the control in the editor.

### Erasing data after the addition

- ▶ We are now going to manage the erasing of data in the controls once the "Validate" button was clicked. Two new WLanguage functions will allow you to perform this operation on all the page controls.
  1. Display the code of the "Validate" button (F2 on the button for example).
  2. Fill the server click code as follows:

```
HReset (Book)
ScreenToFile ()
HAdd (Book)
Reset ()
```

**Reset** resets all the controls for the next input. This is also used to signal to the user that the record was added.

**HReset** resets all the file variables (avoids to store the previous buffer when items are partially added into a record).

3. Save the page.

- ▶ Run the test of the page (  among the quick access buttons). Fill the controls and click the "Validate" button. Several controls can now be entered one after another. Some examples:

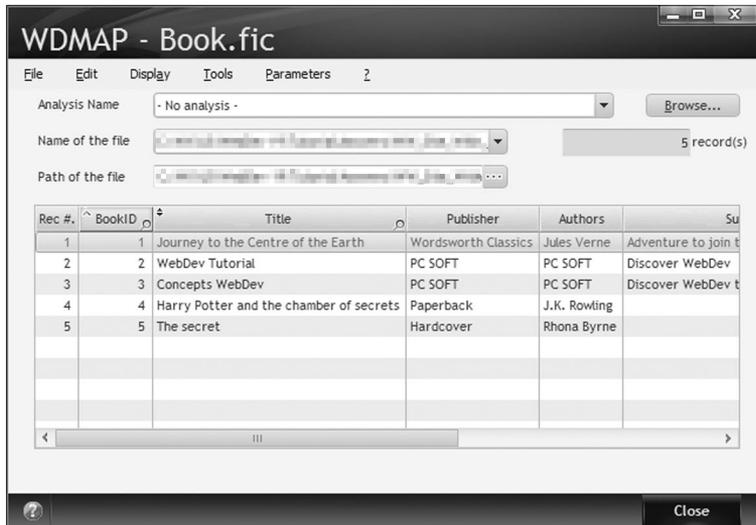
| <i>Title</i>                            | <i>Publisher</i> | <i>Author</i> | <i>Entry Date</i> | <i>Theme</i> |
|---|------------------|---------------|-------------------|--------------|
| Harry Potter and the chamber of secrets | Paperback        | Bégaudeau     | 01/09/2008        | Novel        |
| The secret                              | Hardcover        | Rhona Byrne   | 01/10/2008        | Novel        |

### Viewing the records

Several records have been added to our page. How can I see their content? WebDev proposes a tool used to view the content of the data files when developing the site (when the viewing pages have not been created yet). This tool is named WDMAP. We will be using it to view the content of the Book file.

► To start WDMAP:

1. On the "Tools" pane, in the "Database" group, click "WDMaP".
2. Select the "Book" file. The content of the data file is displayed.



## Managing the upload of the book cover

You probably remember that an image item was created in the Book file to display the cover of the book. We are now going to manage the input and the display of this image.

The purpose is to allow the Web user to associate an image with a book. This image will be uploaded on the server and saved in the "Cover" item of the data file.

The Image control corresponding to the cover of the book was automatically created when creating the controls in our PAGE\_Book page. We are now going to create the controls required to define this cover.

► To manage the upload of the book cover:

1. Create an Upload edit control: on the "Creation" pane, in the "Usual controls" group, expand "Edit". In the list of preset controls, select "Upload".
2. Click the position where the control must be created in the page.
3. Display the description window of the control, specify the name ("EDT\_Cover") and the caption ("Image (Cover)"). Validate the control description.
4. A specific edit control is created in the editor. Indeed, a button without caption is included in the control. This button will be automatically replaced by a "Browse" button displayed by the browser.
5. We are now going to modify the code of the "Validate" button to manage the upload of the image file and the addition of the image into the data file.

The server click code of the validation button becomes:

```
HReset(Book)
FileName is string
IF EDT_Cover<>" " THEN
  // Upload the image of the cover
  FileName = fExtractPath(...)
                UploadFileName(EDT_Cover,False),...
                fFileName+fExtension)
  UploadCopyFile(EDT_Cover,fDataDir(),FileName)
  IMG_Cover = CompleteDir(fDataDir()+FileName)
  // Add the image into the data file
  HLinkMemo(Book,Cover,...
                CompleteDir(fDataDir()+FileName)
END
ScreenToFile()
HAdd(Book)
Reset()
IMG_Cover = " "
```

In this code:

- **UploadFileName** is used to retrieve the name of the file to upload. Indeed, during the upload, the file is renamed with a temporary file name.
  - **UploadCopyFile** is used to copy and rename the uploaded file. This file can be copied into the data directory or into a directory accessible via an alias.
  - **HLinkMemo** is used to load the image in the memo item of the HFSQL "Book" data file.
- Close the code editor and check the modifications performed in the page.

## Closing button

We are now going to create a button used to close the current page and to go back to the previous page or to the home page.

► To create the closing button:

1. Create a new button: on the "Creation" pane, in the "Usual controls" group, click .
2. Click the position where the control must be created in the page.
3. The caption of this button is "Return" and its name is "BTN\_Return".
4. The server code associated with the button is as follows:

```
// Are we coming from a page?
IF PreviousPage()<>" " THEN
  PageDisplay(PreviousPage()) // Yes, display this page
ELSE
  // No, go back to the home page
  PageDisplay(PAGE_Home)
END
```



Note

WebDev also allows you to associate preset actions with the buttons. To manage a closing button (without management of previous page), all you have to do is define the "Display the Home page" action in the "General" tab of the control description.

- ▶ Save the page and run its test.

## Managing the search and the modification

---

We have managed the addition of records into the "Book" file. It's fine. But it would be even better to be able to find and modify a record. That's what we are going to do now, on the same page.

This page will be used to perform a search on the title. Two search modes will be implemented: an exact-match search and a generic search.

Then, we will have the ability to modify the record found.

### Exact-match search

To perform an exact-match search, we are going to select the title of the book in a combo box. The "Search" button is used to display the form of the corresponding book. A single book corresponds to the selected title.

In this first case, the search control is a combo box.

- ▶ To create the search control:
  1. Create a Combo Box control: on the "Creation" pane, in the "Usual controls" group, click "Combo box".
  2. The wizard for creating combo boxes is displayed. We are going to create a combo box based on the "Book" file. Select "Display the data coming from a file or from a query". Go to the next wizard screen.
  3. Select the "Book" file. Go to the next screen.
  4. We are going to display the title of the book. Select the "Title" item. Go to the next screen.
  5. The "Title" item will also be used to sort the list. Select the "Title" item. Go to the next screen.
  6. The item returned by the combo box will be the "BookID" identifier. This is the value that will be sought in the Book file. Go to the next screen.
  7. The combo box will be linked to no item. Go to the next screen.
  8. Give a name and a caption to the combo box ("COMBO\_Title" and "Sought title" for example).
  9. Validate. Position the combo box in the page (top left corner for example).

► To create the search button:

1. Create a button: on the "Creation" pane, in the "Usual controls" group, click .
2. Position the button beside the combo box that was just created.
3. This control is named "BTN\_ExactMatch" and its caption is "Exact-match search".
4. Enter the following server click code:

```
HReadSeekFirst(Book, BookID, COMBO_Title)
IF HFound(Book) THEN
    FileToPage()
END
```

**HReadSeekFirst** is used to perform an exact-match search. In this example, the search is performed on the Book file and on the BookID item. The sought value corresponds to the last parameter of the function. In this case, the sought value corresponds to the value selected in the combo box. This value is obtained by using the name of the combo box (COMBO\_Title).

**HFound** is used to check the result of the search. If **HFound** is set to True, a value was found; if **HFound** is set to False, no value was found. Any record found is read: it becomes the current record in the data file.

In this code, if the record was found, it is displayed by **FileToPage**.



Note

**FileToPage** performs the reverse operation of **PageToFile**: the data found in the items of the data file is displayed in the corresponding controls.

5. Save the page.

► Run the test of the page. Enter a value in the combo box and click the search button. The result is immediate.

## Generic search

We are now going to perform a generic search. Instead of searching for the exact value that was entered, we are going to search for all the elements that start with the value entered.

To perform this search, we are going to create an edit control that will be used to enter the sought name and a button that will be used to perform this search.

► To create the search control:

1. Create an edit control: on the "Creation" pane, in the "Usual controls" group, click .
2. The name of this control is "EDT\_Sought\_title" and its caption is "Sought title".

► To create the search button:

1. Create a button: on the "Creation" pane, in the "Usual controls" group, click .
2. Position the button beside the edit control that was just created.
3. This control is named "BTN\_Generic" and its caption is "Generic search".
4. Enter the following code:

```
HReadSeek(Book, Title, EDT_Sought_Title)
IF HFound(Book) THEN
    Reset()
    FileToPage()
ELSE
    Error("No title corresponds")
END
```

**HReadSeek** is used to perform a generic search. In this example, the search is performed on the Book file and on the "Title" item. The sought value corresponds to the value entered in the EDT\_Sought\_Title control. This value is obtained by using the name of the control.



Note

An exact-match search can be performed by **HReadSeek**: to do so, use the `hIdentical` constant.

**Reset** is used to clear all the controls, including the "EDT\_Sought\_Title" control.

5. Save the page.

► Run the test of the page. Enter a value in the edit control ("Tutorial" for example) and click the search button. The result is immediate. However, if several records correspond to the search, only the first one is displayed.

## Modifying the form displayed

When the result of the search is displayed, it may be interesting to modify the displayed information. Modifying the values of edit controls is child's play but they must also be taken into account in the data file. To do so, we are going to create a modification button.

► To create the modification button:

1. Create a button: on the "Creation" pane, in the "Usual controls" group, click .
2. Position the button below the "Validate" button.
3. This control is named "BTN\_Modify" and its caption is "Modify".
4. Enter the following code:

```
PageToFile()
HModify(Book)
ListDisplay(COMBO_Title)
```

In this code, **HModify** is used to modify the current record with the data found on the screen. **ListDisplay** is used to update the search combo box (if the name is modified for example).



Note

When modifying a record, integrity errors, duplicate errors, ... may occur. The mechanism for the automatic management of errors is enabled by default (as already seen during the addition).

**Note:** The modification of the cover image is not taken into account. This case must be managed in the code for upload management (see “Managing the upload of the book cover”, page 125).

5. Save your page and run its test.

## Browsing the forms

We are now going to add buttons used to browse the different records.

- ▶ Create four buttons named "BTN\_FIRST", "BTN\_PREVIOUS", "BTN\_NEXT" and "BTN\_LAST". The code of these buttons will be :

```
// Btn_first button: call the first one
HReadFirst(Book)
IF HOut(Book) = True THEN
    Info("No form to view")
ELSE
    FileToPage()
END
```

```
// Btn_previous button: call the previous one
HReadPrevious(Book)
IF HOut(Book) = True THEN
    Info("Beginning of file reached")
ELSE
    FileToPage()
END
```

```
// Btn_Next button: call the next one
HReadNext(Book)
IF HOut(Book) = True THEN
    Info("End of file reached")
ELSE
    FileToPage()
END
```

```
// Btn_last button: call the last one
HReadLast(Book)
IF HOut(Book) = True THEN
    Info("No form to view")
ELSE
    FileToPage()
END
```

**HReadFirst** is used to read the first record of the data file, according to the key used for the last search. In this case, we are reading the "Book" file.

**HReadLast** is based on the same principle, but this function reads the record with the greatest key value.

**HReadNext** reads the record whose key value is immediately greater than the one of the current record.

**HReadPrevious** reads the record whose key value is immediately less than the one of the current record.

In any case:

- **HOut** is used to find out whether the data file is empty.
- **FileToPage** is used to display the record on the page.

► Save the page and run its test. Click each one of the buttons to browse the data file.

## LESSON 2.5. MULTICRITERIA SEARCH

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

---

- Creating a query with parameters
- Creating a page with a table based on a query
- Managing the multicriteria search



Estimated time: 20 min

## Overview

In this lesson, you will learn how to handle the tables. In our site, we are going to create a page used to list the books according to the criteria defined by the Web user.

Indeed, a search was already performed on the title of a book, but in some cases, the title is not enough: the name of the author or the publisher can also be taken into account.

The table used to list the result of the search will be based on a query that will be created. The query is used to select the books according to the criteria defined by the Web user.

The page is as follows:

| Identifier of Book | Title           | Publisher | Authors | Summary                        |
|--------------------|-----------------|-----------|---------|--------------------------------|
| 3                  | Concepts WebDev | PC SOFT   | PC SOFT | Discover WebDev through images |
| 2                  | WebDev Tutorial | PC SOFT   | PC SOFT | Discover WebDev                |
|                    |                 |           |         |                                |
|                    |                 |           |         |                                |
|                    |                 |           |         |                                |
|                    |                 |           |         |                                |



Note

This lesson "gives an overview" of query creation. A full section of this tutorial is entirely devoted to the creation of queries ("Create your first query", page 275).

## Creating a query with parameters

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

1. Click  among the quick access buttons. Click "Query" in the wheel that is displayed. The wizard for query creation starts.

2. Select "Select".

Indeed, this query will be used to select records. Go to the next screen.

3. The description window of the query is displayed. To build the query, we are going to select the elements that will be displayed in the result.

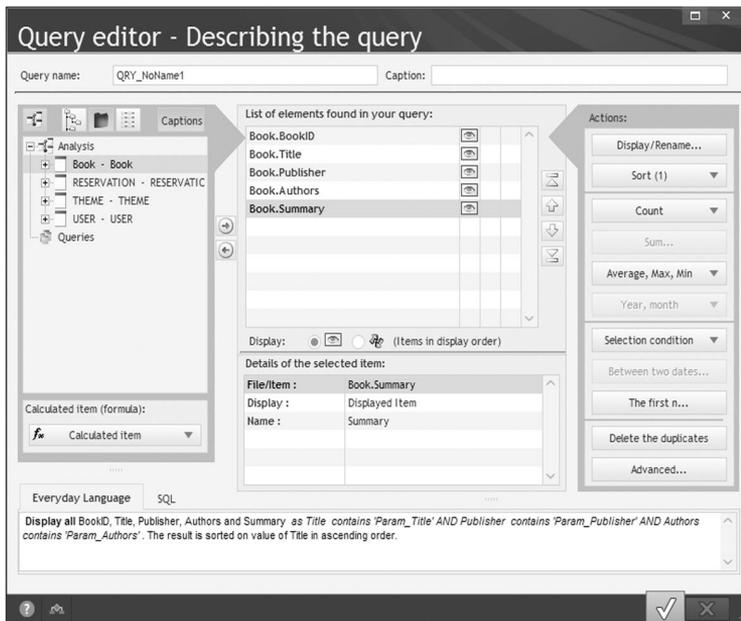
4. The list of data files appears on the left of the screen. To select an item of a data file, you must:

- Double-click the name of the file: the list of file items is displayed.
- Double-click the requested item. This item is displayed in the center of the screen.

We want to display:

- the identifier of the book (BookID item of Book file)
- the title of the book (Title item of Book file)
- the publisher of the book (Publisher item of Book file)
- the authors of the book (Authors item of Book file)
- the summary of the book (Summary item of Book file)

The description window of the query is as follows:



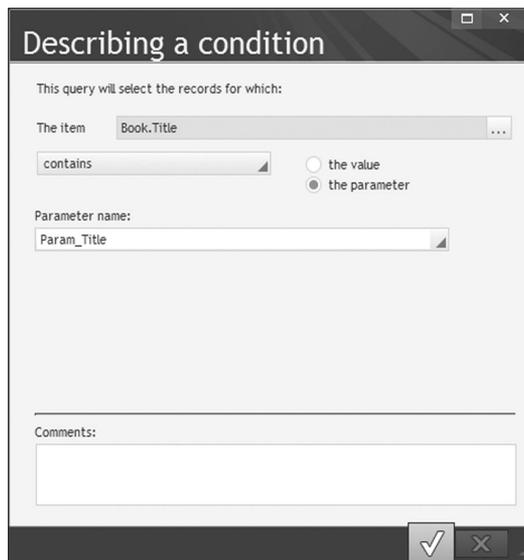
5. The data will be sorted by title. Select the "Book.Title" item then click the "Sort" button and select "Sort on the selected item". Specify an ascending sort on the item and validate.

6. We must now indicate that the books sought by the query must correspond to the elements entered by the Web user. Indeed, a condition can be specified for:

- the title
- the editor
- the authors

Let's start with the title. We must now indicate that the title contains the text entered by the Web user. To do so, select the "Book.Title" item then click the "Selection condition" button and select "New condition".

The selection condition defined for Book.Title is "contains". The Book.Title item must contain a parameter. This parameter is named "Param\_Title".



7. Repeat this operation for the following items:

- the selection condition defined for Book.Publisher is: "contains" the "Param\_Publisher" parameter.
- the selection condition defined for Book.Authors is: "contains" the "Param\_Authors" parameter.

8. Our query is over. Give a name to the query (QRY\_Find\_Book for example, top left of the screen). Validate the description window of the query.

9. Save the query.

10. Let's run the test of the query: click . The window that is displayed asks for the query parameters. Enter the information corresponding to the books entered during the test. The result is automatically displayed.

We are now going to create the page containing the table associated with this query.

## Creating a page with a Table control based on the query

---

This page will allow the user to enter various information used to perform the search for the requested books.

- ▶ To create the page:
  1. Create a new empty page: click  among the quick access buttons. Click "Page" in the wheel that is displayed.
  2. In the wizard for page creation, select the page template named "PAGETPL\_Main" to create your blank page.
  3. Validate. The new page is displayed in the editor.
  4. Give a name (PAGE\_Search for example), a title and a description to your page ("Multicriteria search for books" for example).
  5. Save the page.
- ▶ We are going to display a title in this page. By default, the title is the one of the page template: "Page title". To modify this title in the current page, all you have to do is overload this control in our page to display "Multicriteria search for books" instead.
  1. Click the "Page title" static control and display the popup menu.
  2. Select "Overload the control".
  3. Modify the caption of the control: the new caption is "Multicriteria search for books". This caption can be modified in the page or in the description window of the control.
  4. Save the page.
- ▶ To create the Table control:
  1. On the "Creation" pane, in the "Data" group, expand "Table" and select "Table". The wizard for creating a Table control starts.
  2. We are going to create a Table control based on the query that was just created. Select "Display the data found in a file or in an existing query". Go to the next wizard screen.
  3. Select the query that will be used as data source for the Table control: "QRY\_Find\_Book". Go to the next wizard screen.
  4. Select all the proposed items. Go to the next wizard screen.
  5. Select the search key: "Title". Go to the next screen.
  6. Enter the different parameters of the Table control: keep the options proposed by default. Go to the next screen.
  7. Keep the proposed browse mode. Go to the next wizard screen.
  8. Give a name and a title to the Table control ("TABLE\_Find\_Book" and "Sought books" for example). Validate.
  9. The Table control is automatically created. It can be resized.
  10. Save the page.
- ▶ To create the controls used to enter the sought information, we are going to create the controls from the description of the corresponding items in the analysis then remove the link between the controls and the file.
  1. The analysis associated with the project is displayed in the project explorer. The different data files described in the analysis appear in the pane.
  2. Double-click the "Book" file: the file items are listed.

3. Select the following items with the mouse:
    - Title
    - Publisher
    - Authors
  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 automatically linked to the item in the data file. We are going to delete this link. Indeed, this link is useless because these controls are only used to enter the search criteria.
  6. Select the first edit control that was created ("Title" for example):
    - Display the "Link" tab of the control description ("Description" from the popup menu).
    - Select "None" in the linked item.
    - Validate.
  7. We are going to repeat this operation on all the edit controls:
    - Select the other edit controls (click each control while keeping the CTRL key down).
    - Press the "F4" key: the modifications performed on the first edit control are applied to all the selected controls.
- We are now going to enter the code that will be used to initialize the Table control. This code will be entered in a specific button.
1. Create a new button. This button is named "BTN\_Find" and its caption is "Find".
  2. Display the code of the button ("Code" from the popup menu of the button).
  3. Enter the following code in the server click code of the button:

```
QRY_Find_Book.Param_authors = EDT_Authors
QRY_Find_Book.Param_publisher = EDT_Publisher
QRY_Find_Book.param_title = EDT_Title
HExecuteQuery(QRY_Find_Book)
TableDisplay(TABLE_Find_Book)
```

This code is used to initialize the parameters of the QRY\_Find\_Book query with the values entered in the page controls.

The query is run (**HExecuteQuery**). Then, the Table control is displayed.

- We must now save the page and run its test.

## LESSON 2.6. PRINTING THE DATA

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

---

- Creating a report
- Printing a report from a Web page



Estimated time: 10 min

## Overview

---

In this lesson, we will explain how to print data via reports.

In the previous lesson, we have created a page used to select the records of a multi-criteria search.

In this lesson, we are going to complete this page in order to print the result displayed in the table. The report that will be created is quite simple: we want to print the list of books displayed in the page.

The page being already created, we are going to create the corresponding report and to implement the print mechanism.



Note

This lesson "gives an overview" of report creation. A full section of this tutorial is dedicated to the management of reports ("Your first report", page 301).

## Creating a report

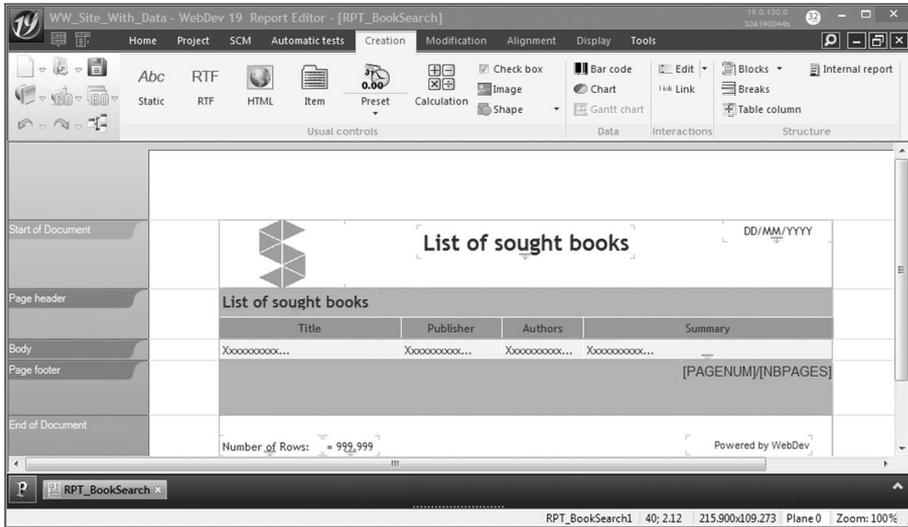
---

We are going to create the report that will be used to print the list of books displayed in the Table control found in the "PAGE\_Search" page.

► To create a report:

1. Click  among the quick access buttons. Click "Report" in the wheel that is displayed. The wizard for report creation starts.
2. Select a "Table" report. Go to the next wizard screen.
3. The report will be based on the Table control found in the PAGE\_Search page. Select "From a table control". Go to the next wizard screen.
4. Expand the "PAGE\_Search" page and select the "TABLE\_Find\_Book" table. Go to the next wizard screen.
5. Select the items displayed in the report. Uncheck the identifier of the book (COL\_BookID). Go to the next screen.
6. Keep the proposed paper format. Go to the next screen.
7. Choose a skin template for your report. Go to the next screen.
8. Give a name and a caption to the report (RPT\_BookSearch and "List of sought books" for example).
9. Validate. Use the landscape mode to display the entire table if necessary.

10. The report is automatically created and it is displayed in the editor.



11. Save the report.

Our report is created but how can we run its test? This report being based on a control found in a page, its test cannot be run directly.

To run the test of the report, a button used to print the report must be created in the "PAGE\_Search" page. That's what we are going to do.

## Starting the report print

In Web, the reports are not directly printed on a printer. Indeed, the printer used would be the one of the server and not the one of the Web user.

To avoid this constraint, all you have to do is print the report in an HTML page or in a PDF file.

A second constraint appears: the file corresponding to the report is created on the server. If several Web users create the same report at the same time, the data must not be overwritten. Therefore, we advise you to give a unique name to the generated report and to delete it from the server once it is displayed on the browser.

In our example, the report will be printed in a PDF file.

- ▶ To print the books found in the Table control:
  1. Open the "PAGE\_Search" page in the editor (if not already done).
  2. Create a button in the "PAGE\_Search" page: on the "Creation" pane, in the "Usual controls" group, click **Ok**.
  3. The name of this button is "BTN\_Print" and its caption is "Print".

4. This button contains the following server click code:

```
// Define the name of the file
PDFName is string
PDFName = CompleteDir(fDataDir())+"Books_searches_"+...
           GetIdentifier()+".PDF"

// Print
iDestination(iGenericPDF,PDFName)
iPrintReport(RPT_BookSearch)
// Display the PDF
FileDisplay(PDFName,"application/PDF")
// Delete the file
fDelete(PDFName)
```

Let's study this code:

- the unique file name is defined by **GetIdentifier**. This function is used to get a unique identifier.
- **iDestination** is used to define that the report will be printed in a PDF file and **iPrintReport** starts the report print.
- **FileDisplay** is used to display the PDF file generated in the browser.
- **fDelete** is used to delete the PDF file.

5. Save the code and run the test of the page. The report is displayed with the content of the elements found in the Table control.

## LESSON 2.7. 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: 10 min

## Overview

A site can be used by different contributors with different profiles. It is sometimes necessary to define different access levels according to the Web user (customer, salesman, manager for example).

Let's take a simple example: when implementing an e-business site, the site proposes the following features:

- Viewing the price list,
- Modifying the price list,
- Entering the orders,
- Managing customers.

The accesses differ according to the Web user. Some examples:

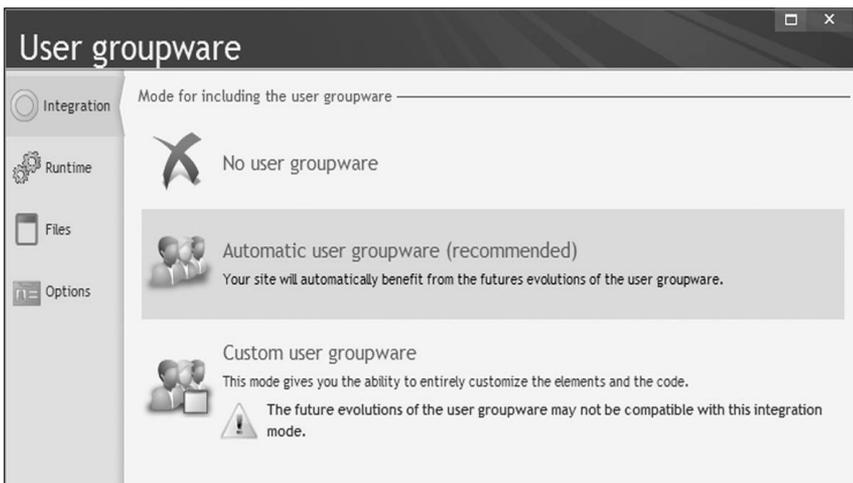
- the Web users can see the price list and place orders.
- the salespeople can see the price list, place the orders and manage the new customers.
- the sales directors have access to all the options.

WebDev allows you to manage these different access levels via the user groupware.

We are going to include the user groupware in our "WW\_Site\_With\_Data" site and configure it.

## Integrating the user groupware

- To include the user groupware in the "WW\_Site\_With\_Data" project:
1. On the "Project" pane, in the "Project" group, click "User Groupware". The window for configuring the user groupware is displayed.



2. Two integration modes are available in the "Integration" tab:
  - **Automatic user groupware:** all the elements found in the user groupware are integrated in your project. This allows you to benefit from all the evolutions of the groupware. Furthermore, the skin template of the project can be applied to the groupware pages.

- **Custom user groupware:** all the groupware elements are integrated in your project via an internal component. You have the ability to customize the different elements of the user groupware. However, the evolutions will not be taken into account.
3. Select "Automatic user groupware".
  4. Select the "Runtime" tab. Two startup modes are available:
    - Auto run: the groupware is started as soon as the application is started,
    - Manual start: the groupware will be started by programming.
  5. Keep the "Auto run" option.
  6. Select the "Files" tab. This tab is used to define the format and location of the data files found in the user groupware. In our case, we will be using HFSQL Classic data files, in the location specified by default.
  7. In the "Files" tab, select "Enable the history of connections". This option allows the supervisor to get information about the connected users.
  8. Select the "Options" tab. This tab is used to define the identification mode of the user:
    - management by the user groupware,
    - using a LDAP directory. In this case, when installing the application, the user will be able to enter the parameters of his LDAP directory.
  9. Validate. A message indicates that a Supervisor user is created.



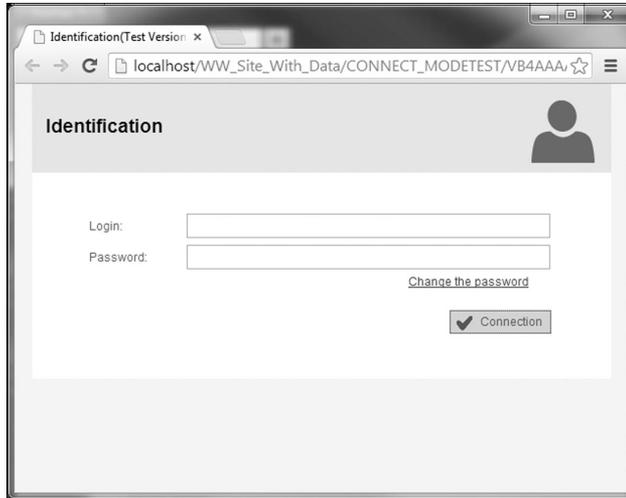
Note

A single user exists by default, the supervisor. During the first startup of the application, connect yourself by using "supervisor". The password used by the supervisor is "supervisor".

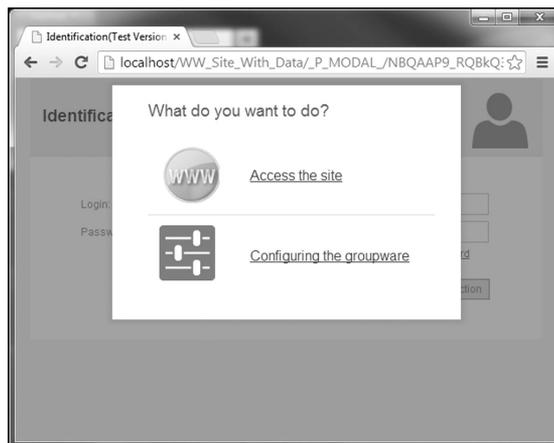
10. Validate this message. The user groupware is included in the application.

► Let's now run the test of our application:

1. Run the test of the project (  among the quick access buttons). A login page is displayed.
2. Connect yourself as supervisor.



3. Enter the password. Click "Connection" to validate.
4. A new menu is displayed, allowing you to run the test of the site or to configure the groupware.



5. Choose "Configuring the groupware".

## Configuring the user groupware

Configuring the groupware consists in defining the different users of the site (or Intranet application) as well as their rights on the different pages and controls.



Note

The configuration of users can be performed :

- when developing the site. The necessary data files (regarding the users and their rights) can be installed along with the site.
- when the site is installed, by the administrator.

### Creating the users and the groups

- ▶ To configure the user groupware, we are going to create a "Test\_1" user and associate this user with the "Tests\_site" group. Then, we are going to configure the management of rights for the group.
- ▶ To create a user:
  1. Click the "New" button found below the "Users" section. The screen for entering a new user is displayed.
  2. Enter the following information:
    - Login: Test\_1
    - Last name: Test\_1
    - Password: Test\_1

Enter the user details

Login of the user:

Email address:

LastName:

First name:

Phone:

The user is supervisor of the user groupware

User enabled (integrated user groupware)

Creating the password during the first startup

Force the following password:

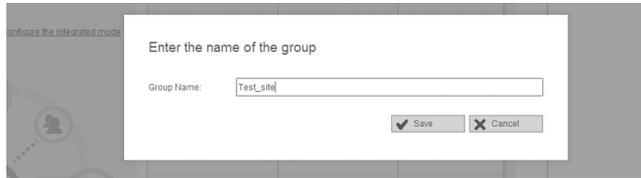
Password:

Confirm:

3. Click the "Save" button. The "Test\_1" user appears in the list of users defined for the user groupware.

- ▶ To create a new group of users:
  1. Click the "New" button found below the "Groups" section. The screen for entering a new group is displayed.

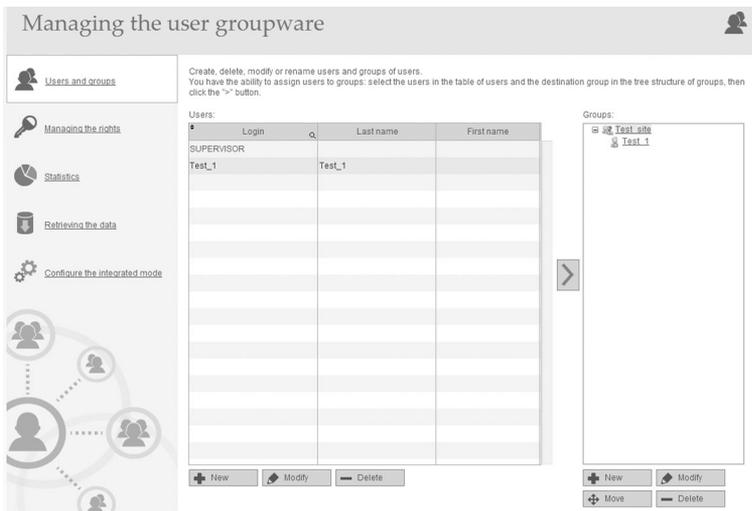
2. Enter the name of the group: "Tests\_site".



3. Click the "Save" button. The "Tests\_site" group appears in the list of groups defined for the user groupware.

- ▶ To associate the user with the group:

1. Select the "Test\_1" user.
2. Select the "Tests\_site" group.
3. Click the arrow.



4. The association is performed.

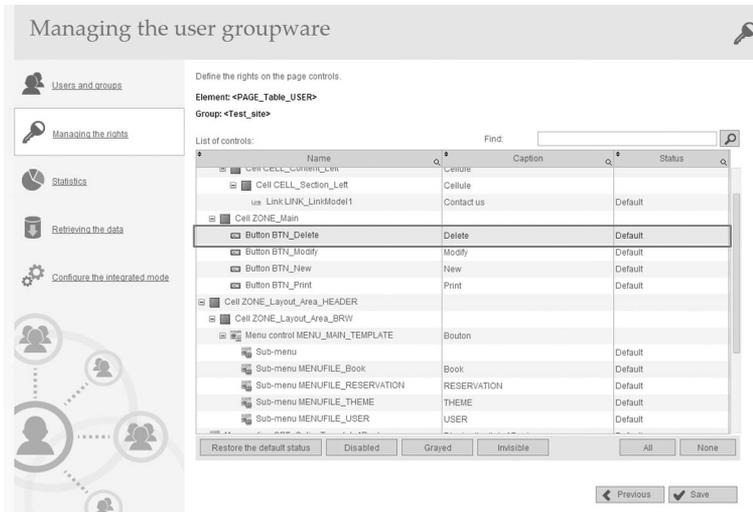
## Defining the rights

We are now going to define the rights for the "Tests\_site" group. These rights will be granted to all the users of the group. In our example, the users found in the "Tests\_site" group will not be allowed to delete a user (from the PAGE\_Table\_USER page).

- ▶ To define the rights:

1. Click "Managing the rights" on the left of the page
2. Select the "Tests\_site" group.
3. Click "Next".
4. The page that is displayed is used to select each page or report of the application.
  - For each page or report, you have the ability to specify whether the element can be accessed (or not) by the group.
  - For each page, you can define whether the page controls will behave like the application (default) or whether they will be disabled, invisible or grayed.

- ▶ To modify the rights in the "PAGE\_Table\_USER" page:
  1. Select the "PAGE\_Table\_USER" page in the list of pages.
  2. Click the "Next" button.
  3. The window for configuring the rights on the page controls is displayed.
  4. In the list of controls, select "Button BTN\_Delete".



5. Click the "Grayed" button.
6. Click the "Save" button.
7. Close the configuration page.
8. The WebDev editor is displayed.

## Running the test of the site

- ▶ We are now going to run the test of the site by using the "Test\_1" login.
  1. Run the test of the project. A login page is displayed.
  2. Connect yourself as "Test\_1" with the "Test\_1" password. Validate.
  3. The site is started.
  4. Select "User .. Display the list of USER".
  5. You will notice that the "Delete" button is grayed.
  6. Exit from the site and go back to the editor.

## Disable the management of user groupware

The user groupware will no longer be used in the rest of this tutorial. Therefore, it can be disabled.

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 2.8. THE DASHBOARD

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

---

- What is the dashboard?
- Automatic tests



Estimated time: 10 min

## Overview

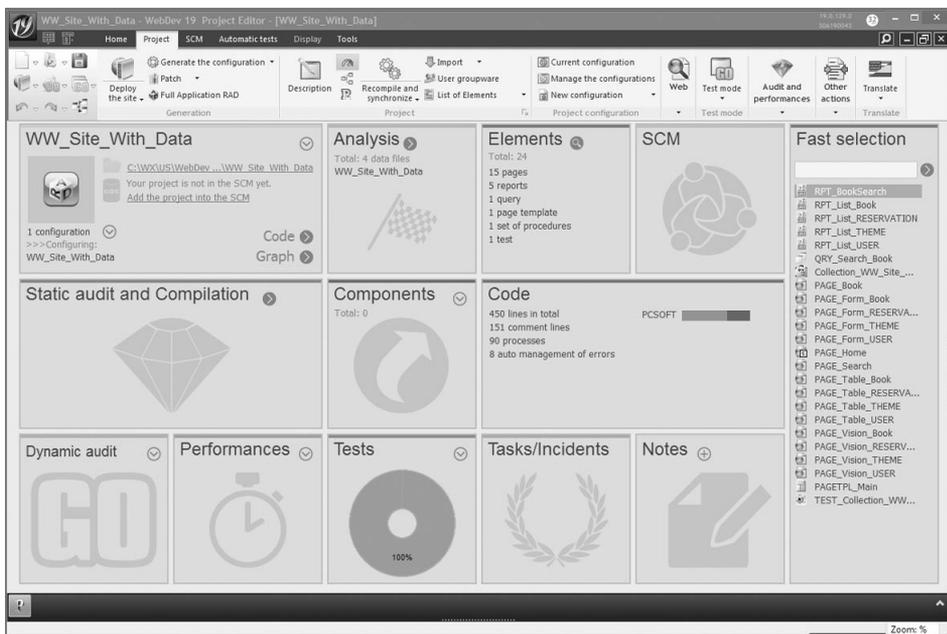
The project dashboard is an essential element for managing the WebDev projects. The project dashboard gives an overall view of the progress status of a project.

The dashboard includes several progress bars, lights and counters that give an overall view of the status of a project.

In this part, we will present the management of automatic tests.

The dashboard will be presented in details in "Dashboard", page 357.

- ▶ To display the project dashboard (if not already done), on the "Project" pane, in the "Project" group, click .



The project dashboard includes several Widgets. Each Widget displays various information about the project.

For example, the "Performances" Widget is used to start the performance profiler or to open the last performance report.

The "Static audit and Compilation" Widget is used to quickly see if the project contains compilation errors.

You have the ability to enable (or not) a Widget, to add one or to delete one at any time.

## Automatic tests

One of the interesting features of the dashboard is to show the tests that have been run on the project.

Several tests have already been run while developing our project.

The automatic tests are a category of specific tests. The automatic tests are used to automatically perform some operations of your global procedures. These tests are recorded as WLanguage scenarios and they can be easily modified in the code editor. Once recorded, the automatic test can be re-run as many times as necessary, to test for example the impact of a modification made to a procedure, a class, ...

Let's give it a try! We are going to create a global procedure in our project then an automatic test will be run on this procedure.

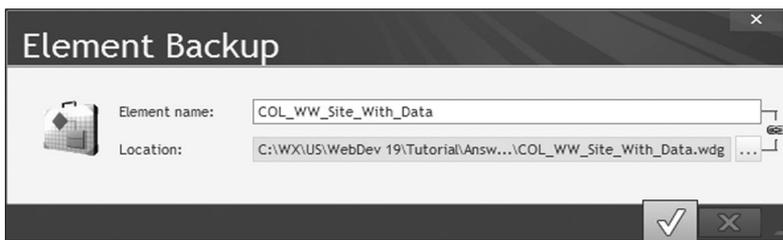
**Note**

In WebDev, the automatic tests can be run on the procedures or on the classes.

The global procedure that will be created is used to check the internal code entered for each record found in the "Book" file. This internal code must contain 2 letters, a dash and 3 digits.

► To create a global procedure:

1. In the project explorer, select "Procedures".
2. Select "New set of procedures" from the popup menu.
3. Specify the name of the set ("COL\_WW\_Site\_With\_Data") and validate.
4. Save the set of procedures (CTRL + S) and validate the backup window.



5. In the project explorer, select the set of procedures that was just created.
6. Select "New global procedure" from the popup menu.
7. Specify the name of the procedure: "CheckInternalCode" and click "Add".

8. The procedure appears in the code editor. Enter the following code:

```
PROCEDURE CheckInternalCode (InternalCode)
  IF MatchRegularExpression (InternalCode, ...
      "[A-Z]{2}[-][0-9]{3}") = True THEN
    RESULT True
  ELSE
    RESULT False
  END
```

Let's take a look at this code:

- The procedure receives an `InternalCode` parameter.
- **MatchRegularExpression** is used to check that the internal code corresponds to a set of sub-strings in a specific format. In our case, `[A-Z]{2}` is used to specify that the internal code must start with 2 letters in uppercase characters. `-` is used to define that the "-" character must be used after the 2 letters, `[0-9]{3}` indicates that 3 digits must be found after.



Note

The syntax of **MatchRegularExpression** may seem difficult to build. WebDev proposes a wizard to help you build the code corresponding to your case. To start the code wizard, type "MatchRegularExpression" in the code editor and click the "Wizard for MatchRegularExpression" option that is displayed in a combo box. A wizard is available for several WLanguage functions.

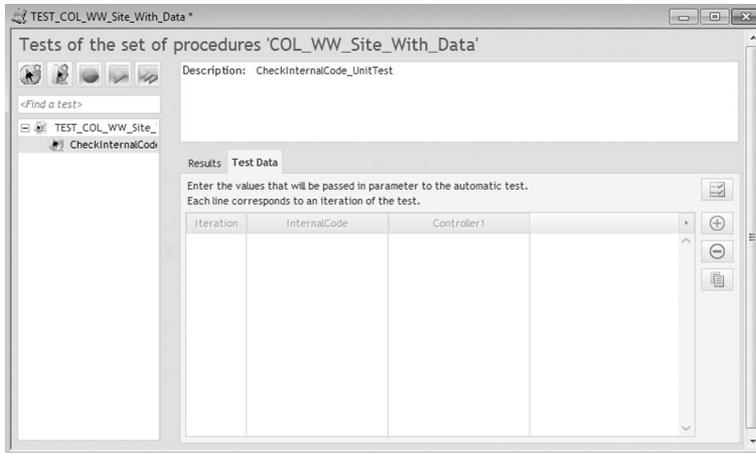
- If **MatchRegularExpression** is set to True, the internal code will be validated: the procedure returns True. Otherwise, the procedure returns False.

9. Save the code of the procedure.

We are now going to create a test on this procedure.

- ▶ To create an automatic test on a procedure:
  1. In the project explorer, select the procedure that was just created: "CheckInternalCode".
  2. Display the popup menu and select "Unit tests .. Create a unit test".

3. The editor of automatic tests is displayed.



4. Three columns are found in this table:

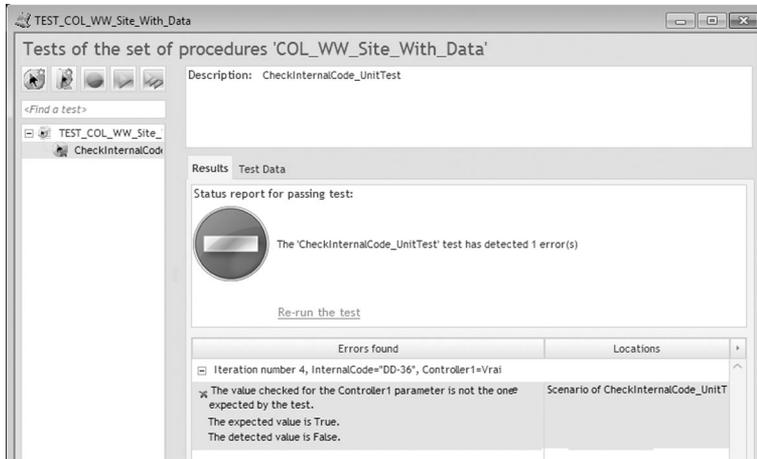
- Iteration that corresponds to the iteration in the test. Indeed, a set of tests will be used to run the test of our procedure. Each iteration will correspond to a test.
- InternalCode that corresponds to the value of the InternalCode parameter of our procedure. The value to test must be entered here.
- Controller1 that corresponds to the value expected for the result of the test (True or False in our case).

5. We are going to enter different values to run the test of our procedure. The "+" button is used to add a new iteration. Enter the following iterations:

| Iteration | InternalCode | Controller1 |
|-----------|--------------|-------------|
| 1         | AA-352       | True        |
| 2         | BBB-456      | False       |
| 3         | CC_333       | False       |
| 4         | DD-36        | True        |
| 5         | ee-360       | False       |

When entering these values, you have noticed that line 4 is incorrect.

6. Let's now run the test of our procedure (  among the quick access buttons). The editor is minimized and the test is run. The result is displayed in the "Results" tab. In this tab, a message informs you that a problem occurred on the iteration 4.



7. We are going to modify the iteration 4: click the "Test data" tab and modify the iteration 4: the controller must be set to False.
8. Run the test again. The test is available. The test can now be run at any time.
9. Close the test editor and save the test if necessary.

You have handled a simple example for creating unit tests (also called automatic tests) in WebDev. See the online help (keyword: "Automatic test") for more details.



**PART 3**

**Pages and  
controls**

**10**

**DEVELOP 10 TIMES FASTER**





# LESSON 3.1. THE PAGES

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

---

- How do I create a page?
- The characteristics of a page
- The processes associated with a page



Estimated time: 15 min

## How do I create a page?

---

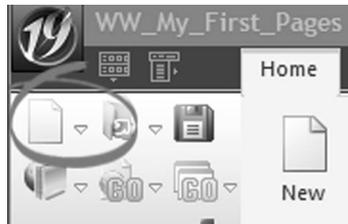
Since the beginning of this tutorial, several methods have been used to create pages:

- creating blank pages based on a template (part 1 of this tutorial)
- creating different types of pages (form, table, ...) based on data files (part 2 of this tutorial).

Several other methods can be used to create pages in WebDev. These methods are grouped in the wizard for page creation.

**Reminder:** To start the wizard for page creation:

1. Click  among the quick access buttons of the WebDev menu:



2. A window shaped like a wheel is displayed. This window is used to create all the elements that can be associated with a project.
3. Click "Page". The wizard for page creation starts.

The wizard for page creation is used to:

- create **blank pages or pages based on a page template** of the project.
- create **internal pages**. The internal pages are a specific type of page. Their operating mode will be presented later in this section.
- create **RAD pages** (Rapid Application Development): these pages are based on the data files described in the analysis and they contain the code required for them to operate. These pages are associated with a RAD pattern (that defines the features included in the page as well as the interface if necessary). These pages can be used immediately. Several types of pages are proposed: form page, page with table, page with loop, ... The RAD pages are available in dynamic mode or in semi-dynamic mode.
- create **RID pages** (Rapid Interface Development): these pages are based on the data files described in the analysis. They only contain the controls, the buttons and the code required by the elements included by the associated RID pattern. The corresponding code must be entered by the developer. These pages are linked to a RID pattern. These pages can be used immediately. Several types of pages are proposed: form page, page with table, ...
- create **framesets**. The framesets are used to display the requested information in different sections of the browser.
- create **pages based on a page template**. The page templates are used to define a set of criteria (graphic, control, code) that must be re-used in each page of the application. The creation and the use of a page template will be presented later in this part.

- create **pages for a Vista Gadget**. In this case, three types of specific pages can be created. See the online help (keyword: "Vista Gadget") for more details.

You still have the ability to create a blank page, without control, that uses (or not) a template. Let's now present the characteristics of a page in details.



Notes

In the "Ergonomics of the site", page 204, we will see how to improve the interface of the site pages by using the "zoning" mode.

## Description of a page: 7 tabs are available

Several pages have been created since the beginning of this tutorial. The first element displayed was the description window in order to enter the title of the page, its name and its description.

The description window includes several useful settings, saving you from writing several code lines.

Some of these features will be presented in a simple example.

### Practical example

To handle the description window of pages, we will be using an example created for this purpose.

1. Start WebDev 19 (if not already done).
2. Close the current project if necessary. The home window is displayed.
3. In the home window, click "Tutorial" and select the "Pages and controls" project. The project is loaded.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Pages and controls".

4. Open the "PAGE\_Button" page (found in the "Standard controls" folder of the project explorer).



Notes

#### Opening a page in the editor

Several methods can be used to open a page in the editor. Summary of the different methods:

- Press [CTRL] + [E] and select the page to open. The benefit of this method is that the page is previewed.
- In the "Project explorer" pane, double-click the name of the page to open.
- Click  among the quick access buttons and selecting the file corresponding to the page to open.

5. Display the description of the page ("Description" from the popup menu of the page).

## "General" tab

The first tab of the description window is the "General" tab. This tab is used to specify the general parameters of the page:

- The **name** of the page. This name is used in programming to handle the page.
- The **description** of the page. The description is intended for the developer. It gives additional information about the purpose of the page. It is used when printing the documentation.
- The **title** of the page. As already seen, the title is displayed in the title bar of the browser.
- The **type of the page**:
  - **Dynamic page** (default option in a dynamic site): choose this option if you are using server code or if you handle data files. A dynamic page uses the deployment engine to operate.
  - **Semi-dynamic page**: your page contains no server code but it can be generated from a data file or from a query (for a site presenting product forms, distributed on a CD for example). WebDev will generate as many HTML pages as necessary to display the entire content of the selected data file. You have the ability to define the number of records that must be displayed on each page!  
A semi-dynamic page requires no deployment engine.
  - **Static page** (default option in a static site): choose this option if your page includes no server action or if your page only contains images or static controls. A static page requires no deployment engine.
  - **PHP page**: choose this option if your page is included in a PHP site.

Note: Depending on the selected type of page, you have the ability to enter options in the "Options" area (found beside the type of page).

- The **target platform and the size of the screen**:
  - The target platform allows you to define the type of browser used to view the page.
  - The size of the screen is used to define the size of the page.
- **The standard for page generation**. You have the ability to define the HTML standard used to generate the page. This standard corresponds by default to the one of the project (defined in the "Advanced" tab of the description window of the project).
- The **edit mode** of the page. Two edit modes are available:
  - the edit in zoning mode: this edit mode allows you to use the simplified anchors (as already seen in the first part). This edit mode is used by default for all the new created pages.
  - the edit in mode compatible with version 16: in this case, the anchoring is managed via the extensible areas. This edit mode is kept for all the pages created with an earlier version. The edit mode can be modified at any time. All you have to do is regenerate the corresponding page to take the modifications into account.

## "GUI" tab

The "GUI" tab is used to specify the parameters for managing the controls and the page:

- The cursor of the page.
- The anchoring of the page in the browser.

## "Details" tab

The "Details" tab is used to define the parameters regarding the Ajax indicator, the referencing as well as the options for managing the page.



Notes

AJAX (Asynchronous Javascript and XML) is used to refresh the data modified in an HTML page without having to redisplay the entire page. For example, if some elements found in the page displayed (content of the basket, characteristics of a product, list of cities, map, ...) are modified, only these elements will be refreshed. The server does not have to send the entire page onto the computer of the Web user.

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

This tab includes:

- The mode for managing the **Ajax indicator** (if the processes are in Ajax mode)
- The referencing options: the page can be referenced or not. You have the ability to define expressions and keywords for each page. This feature will be presented later in this tutorial.
- The validity of pages.
- The independent HFSQL context
- The default target. This option is used to define a default target for all the controls that perform an action on the page (validation, page opening, ...).
- The management of the browser "Back" button.

## "Frame" tab

The "Frame" tab is enabled if the page is included in a frameset. The page is considered as being a "Frame". You have the ability to modify the options about:

- Width and height of the frame.
- Border of the frame.
- Display of a scrollbar.
- Resizing of the frame.

## "Language" tab

The "Languages" tab is used to define the different languages supported by the page (for the multilingual projects). See "Multilingual", page 418 for more details.

## "Note" tab

This tab allows you to enter a text. This text is printed in the documentation. You can, for example, enter information about the operating mode of the page, the rules for managing the processes.

## "Advanced" tab

The "Advanced" tab is used to manage the specific Web features. You have the ability to:

- choose not to generate the HTML file corresponding to the page
- let the browser manage the tab order between the page controls
- allow the access via *DynamicSiteDisplay*.

The "HTML" tab is used to:

- include HTML code that will be added to the page header (between the "<HEAD>" and "</HEAD>" tags in HTML code).
- include information as "META tag". For example, include an automatic refresh of the page, etc.

The "Javascript" tab is used to add Javascript files. These files will be used by the generated HTML page.

The "Device" tab is used to define the parameters that will be used when generating a site for iPhone or for BlackBerry. This tab only appears for an iPhone or BlackBerry platform ("General" tab in the page description).

## "Style" tab

This tab groups the parameters regarding the style of the page. You have the ability to select:

- The skin, the color palette and the style sheet associated with the page.
- The style of the page elements:
  - For the browser background, the background color, the background image of the page and its display mode.
  - The default text color, the color and the style of page links. These options can be modified control by control.
  - The characteristics of the border and page background.
  - The margins.

# LESSON 3.2. THE CONTROLS

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

---

- The different types of controls
- The standard controls
- The specific controls



Estimated time: 2h

## Introduction

---

WebDev proposes more than 50 types of controls for communicating with the Web users. They are used to enter or view values.

The displayed values can come from a calculation performed by programming, from a file found in a database or from an assignment.

The entered values can be used to perform calculations, they can be saved in a file found in a database or assigned to other controls.

This lesson will allow you to discover the different types of controls, to run their test and to program them.

To simplify this lesson, the different controls have been divided into two categories:

- the standard controls: these controls are the most frequently used.
- the specific controls: these controls are used to display a specific interface or to manage specific features.

### Practical example

To learn how to use the different types of controls, we are going to use the "WW\_Pages\_and\_controls" project that was created for this purpose. To open this project in WebDev:

1. Close the current project if necessary. The home window is displayed.
2. In the home window, click "Tutorial" and select the "Pages and controls" project.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Pages and controls".

## The standard controls

---

The "Standard controls" are the controls that are frequently used in the Intranet or Internet sites created in WebDev. These controls are as follows:

- Static, HTML static, Rich Text Area
- Edit control
- Button
- Link
- Image, clickable image
- Radio button, check box
- List
- Combo box
- Table
- Looper
- TreeView
- Graph
- Cell
- Rating

These controls will be presented in details. A specific page is available for each type of control, allowing you to check the main features of the control. These pages are grouped in the "Standard controls" folder in the project explorer.

**Type of control: Static***(static page, semi-dynamic page and dynamic page)***Summary**

The static controls are used to display a static text. Their content cannot be modified by the Web user but these controls can be modified by programming in server code (like all the controls).

The static controls are used to display an information, a title in large characters for example. The content of a static control can be defined in the editor or by programming.

The static controls can also be used to display a text according to the language used in the project without additional programming.

**Practical example**

- ▶ Open the "PAGE\_Static.WWH" page.
- ▶ Run the test of the page. Click the buttons to find out how the simple static controls and the formatted static controls can be used.

**The different types of static controls**

WebDev proposes several types of static controls:

- 1. The simple static controls** : these are "standard" static controls onto which a single font style (size, color, ...) can be applied.
- 2. The HTML static controls** : these static controls allow you to format the entire text or part of the text. The style can be modified for one or more sections of text (size, color, font).
- 3. The formatted static controls** : these static controls (also called formatted display controls) are used to display information according to a display mask. These static controls are very useful to display a date, a time, a currency value, ...
- 4. The Rich Text Areas** : these static controls allow you to format the entire text or part of the text. This type of control recognizes the HTML tags. The Rich Text Area control is used to format the paragraphs: title, sub-title, sub sub-title, page footer, email address, note, page header, ...

**Handling the static controls by programming**

Let's take a look at the code used to handle the static controls (we advise you to study the code of the different buttons found in this page).

To retrieve the caption of "STC\_Get" ("Get" button), the syntax is:

```
Str is String
Str = "Content of static control " + CR + STC_Get
// or
// Info("Content of static control " + CR + STC_Get)
```

You even have the ability to change the color of the text displayed in the static control ("Text" button) with **..Color**.

```
STC_Color..Color = LightRed
```

**LightRed** is a WLanguage constant corresponding to the Light Red color.



Notes

The properties are used to modify some characteristics of the controls by programming: color, caption, font, ... We will not present all the properties. We will discover some of the main properties as we progress through this lesson.

All the properties are relative to a given control. Their syntax is as follows:

```
Control_Name..Property_Name
```

The list of properties available for the controls is found in the online help of Web-Dev (keyword: "Properties, Properties of the page controls").

### Type of control: Edit control



(*static page, semi-dynamic page and dynamic page*)

### Summary

The edit controls are the first controls that you have been using. They are used to enter data or to display data coming from variables, calculations or data files.

They can be used to enter a password, a quantity to order, the name of a customer, an address, ...

When you create a page, there are 9 chances out of 10 that you will be using an edit control. Let's take a closer look at the operating mode of this type of control.

Regardless of the information to enter or display, you have the ability to choose the corresponding type:

- text,
- password: The characters found in the controls are replaced by stars ("\*") when they are displayed on the screen.



Tip

This type of edit control is ideal for entering the password of a Web user in a login page.

- numeric,
- time,
- date,
- duration,
- currency,
- currency+Euro.

Several input masks (or display masks if the control is read-only) correspond to each type of control. We have already seen the benefit of this mask when developing the "Site\_With\_Data" site.



Reminder

The numeric edit controls have no defined type. The input mask selected for the control will define whether the control is a real, a double real, an integer, ...

## Practical example

- ▶ Open the "PAGE\_Edit.WWH" page. This page presents the management of text, numeric and date edit controls.
- ▶ Run the test of the page.
- ▶ Enter values in the "Editable" controls. No input can be performed in the other controls. Indeed, they have been defined as read-only. Click the different buttons and see what happens.
- ▶ Go back to the editor and study the code of buttons.

## All the details

A simple assignment is all it takes to display and retrieve a value in an edit control, regardless of the type of the edit control.

### Some examples of initialization:

- Initialization code of a "text" edit control:

```
EDT_EDIT2= "Hello" // Displays Hello in the text control
```

- Initialization code of a "numeric" edit control:

```
// Displays the value 20.6 in the numeric control  
EDT_Numeric = 20.6
```



### Reminder

You also have the ability to initialize the value of an edit control by using the "Content" tab of the control description.

### Some examples of retrieval processes:

- Click code of a server or browser button used to retrieve the value of a text edit control:

```
Str is String  
Str = EDT_EDIT2 // Retrieves the content of the text control
```

- Click code of a button (server or browser) to retrieve the value of a numeric edit control:

```
Value is Real  
Value= EDT_Numeric  
// Retrieves the content of the numeric control
```

## Managing the color in the edit controls

In the input area, the text color and the background color can be modified by programming with **..Color** and **..BrushColor**.

```
// Changes the color of the text  
EDT_Oper..Color = LightRed  
// Changes the background color  
EDT_Oper..BrushColor = LightYellow
```

## Characteristics of an edit control

As for the pages, a description window is available for the controls. This description window is used to configure several options of the control (style or operating mode).

We are now going to describe the characteristics of the edit control, keeping in mind that several options are available in the description of other controls. We won't go into details about all the available options in this tutorial. See the context-sensitive help for more details.

### "General" tab

As already seen before, the "General" tab of the edit control is used to define:

- the name of the control
- its caption
- its type
- its input mask.

Some options can be very useful:

- **Max input size:** Maximum number of characters that can be entered or assigned in a Text edit control.
- **Visible rows:** For the text edit controls, this option is used to enter a text over several lines. You have the ability to insert CR characters (Carriage Return) into the text.



Tip

This type of option is perfect for entering comments or notes. Using this parameter is essential for the controls linked to a "text memo" item.

- **Reset if zero:** The content of the control is automatically cleared if the value taken by the control is null.
- **Returns NULL if empty:** Returns the **NULL** value if the control has no value (Caution: No 0 and no empty string).



Tip

The **NULL** value is very useful to pass parameters to a query and to make the conditions optional for multicriteria searches.

### "GUI" tab

The GUI tab is used to define the characteristics of the control interface.

**The initial status of the control** corresponds to the status of the control when displaying the page. This characteristic is available for nearly all types of controls.

When displaying a page, an edit control can be:

- **Enabled:** the control is active; it will be in edit in the page. The Web user will be able to enter data in the control.
- **Read-only:** the control is inactive. No value can be entered by the Web user in the control. The control will behave like a static control. However, the content of the control can be selected with the mouse.
- **Grayed:** the control is grayed. No value can be entered by the Web user in the control.

- **Visible** (common to all controls): the control "exists" in the page but it can be visible or invisible. This status can be very useful in programming when a control must be displayed or not according to conditions!

The initial status of a control is chosen during its creation. It can be modified by programming in server code by using the `..State` property for the "Enabled" and "Read-only" options and the `..Visible` property to make it visible or not.

#### WebDev proposes the following types of controls:

- **Automatic** (common to all the controls found in a dynamic page): if the control is modified by programming, the control is considered as being "dynamic".
- **Static** (common to all the controls found in a dynamic page): the content of the control cannot be modified by programming.
- **Dynamic** (common to all the controls found in a dynamic page): the content of the control can be modified by programming.

**The control can be overlaid** (common to most controls): if this option is checked, this control can be positioned above another control. You will have the ability to move it with the mouse and to store its new position.

The **"shortcut"** allows the Web user to be directly positioned on the control by pressing a combination of keyboard keys ([ALT] + [Letter] for example).

The shortcuts are defined in the "Details" tab of the control description.

A specific rollover cursor can also be defined for each control.

The **"Groups"** button (available in the dynamic pages) is used to "group" several controls together in order to modify the status of these controls in a single operation.

#### "Details" tab

The Details tab is used to define the advanced characteristics of the control.

The control can be **accessible by TAB**. In most cases, the TAB key allows the Web user to move between the different page controls. This enables you to define the tab order in the page (this topic will be presented later in this part).

The control can be in **mandatory input**: in this case, a message prevents from validating the page if no value was entered in the control by the Web user.

**Management of Euro**: Characteristics regarding the Euro for the "Currency + Euro" controls (displayed currency, currency used in programming, ...).

**Type of control: Button****(static page, semi-dynamic page and dynamic page)****Summary**

The buttons are used to start processes. They are also called "Controls". The process associated with the button will be run when the button is enabled (when the button is clicked by the Web user).

**Practical example**

- ▶ Open the "PAGE\_Button.WDW" page in the editor. This page presents several types of buttons.
- ▶ Run the test of this page.

**Types of operations for the buttons**

A button can perform one of the 3 following types of operations (information that must be specified in the description window, "General" tab):

1. Send the value of controls to the server (submit)
2. Reinitialize the page controls (reset)
3. No specific operation

**"Send the value of controls to the server" button (submit)**

The "Send the value of controls to the server" button are the most common ones.

During a click on this type of button:

- the browser code is run (if it exists).
  - a call to the server is performed.
  - the page is returned to the browser.
- ▶ To check this, click the "Value of control" button. The message corresponding to the browser code ("Send to server") is displayed, then the date is modified and the message corresponding to the server code is displayed.

**Tip**

This type of button is recommended to display data coming from a database in the browser.

**"Reinitialize the page controls" button**

When the button is enabled, all the page controls containing a value entered by the Web user are reset. The controls containing a value assigned by programming are not reset. The server click code is not run.

- ▶ To check this, enter a text in the "Information" control and click the "Reinitialize" button. The content of the control is cleared.

**Tip**

This type of button is recommended to cancel the input of a form for example.

### Button whose type is No specific operation

When this button is enabled, only the browser click code is run. The value of the page controls is not returned to the server.

- ▶ To check this, click the "Browser action" button. Only the browser code is run.

Note: In the code editor, a compilation information message is displayed, indicating that the server code will never be run.



Tip

This type of button is recommended to perform processes that do not require to save the information entered by the Web user.

### Characteristics of a button

The image of the button can be:

- an image created by yourself.
- chosen among the ones proposed in the WebDev catalog (in the "Cliparts" or in the "Graphic buttons" for example).



Tip

We advise you to:

- give no caption to a graphic button whose image includes a caption, otherwise the button will contain two captions.
- avoid using graphic buttons and text buttons in the same page (visual aspect, ...).

The image associated with a button is copied into the `_WEB` directory of the project.

The image of a graphic button can be modified by programming (browser code):

```
// Modifies the image associated with the button
// The image is found in the _WEB directory
BTN_Image..Image = FolderWeb() + ...
                        "/ball_beach_s_gekko_goldenyellow_72.png"
```

**Note:** *FolderWeb* is used to find out the path of the directory containing the images in browser code.

### Style of a button

WebDev proposes several "styles" of buttons:

- **Button with fixed image or with rollover effect (also called two-state button):** The two states are the normal status and the rollover status. All you have to do is associate the button with an image for each status. This style is defined in the "Style" tab by selecting the "Button (Images)" element.
- **Button associated with an image sheet:** normal status, rollover status, grayed status, pressed status, ... All you have to do is associate the button with an image containing the drawings of the different button states. This style is defined in the "Style" tab by selecting the "Button (Images)" element.
- **Standard button** ("text" button). The appearance of the button is defined in the style sheet of the project.
- **Custom button.** The appearance of the button is defined via the HTML code entered in the button caption and via the style options ("Style" tab, "Border/Background" element for example).

**Type of control: Link***(static page, semi-dynamic page and dynamic page)***Summary**

The links are used to display static text. You have the ability to define the action that will be performed during a click on the link (display a page, perform a process, open the messaging software of the Web user, ...).

In most cases, the links allow the Web user to send an email to the site manager, to open a new browser to an Internet address, ...

The content of a link and its action can be defined in the editor during the link description or by programming (WLanguage ..URL property).

**Practical example**

- ▶ Open the "PAGE\_Button.WDW" page in the editor. This page presents the use of a Link control.

**Type of control: Image***(static page, semi-dynamic page and dynamic page)***Summary**

The Image controls are used to display images. A browser process or a preset action can be performed during a click on an image.

**Practical example**

- ▶ Open the "PAGE\_Image.WDW" page in the editor. This page presents several types of images:
  - Generated image
  - Static image (homothetic or stretched)
- ▶ Run the test of this page.
- ▶ Go back to the editor.

**Types of images proposed by WebDev**

WebDev proposes several types of images in the dynamic and semi-dynamic pages. These types of images can be configured in the "General" tab of the description window of the control.

The available types are as follows:

- **Static:** the image associated with the Image control is fixed.  
The image will never change and it can be modified by programming in browser code only.  
In most cases, this type of image is used to display a logo or a static banner.
- **Dynamic:** the image can be modified by programming (server or browser code).  
In most cases, this type of image is used to display the photos of a product catalog.

- **From a database:** the image is a generated image that can be linked:
  - to an item of a data file containing the path of the image only.
  - to an "Image memo" item of a data file.

By default, this image is displayed in "homothetic" mode so that it is not distorted regardless of the size of the Image control.

- **Generated:** the image is "built" by programming. It can be an image whose source file (GIF, JPEG, PNG, ...) was defined. In most cases, this type of image is recommended when using the drawing functions or the chart functions of WLanguage.

## Characteristics of an image

The description window of an image contains several specific parameters. We are going to present them in details.

### Name and location of the image ("General" tab)

Like for the controls, the name of the image will be used in programming to modify the attributes of the image (path, visibility, ...).

The path of the image specifies the initial location of the image file.

The "Catalog" button is used to access the image catalog supplied with WebDev ("The image catalog", page 175).

### Display characteristics ("General" tab)

WebDev allows you to configure the display mode of the image:

| Display mode | Example   | Description   |
|--------------|---|---|
| 100 %        |   | The image is displayed with its initial size. The size of the control can be modified but it does not affect the image.   |
| Centered     |  | The image is centered in the image control. The size of the control can be modified. If the image is smaller (in width or in height) than the control, bars are displayed on the sides. |
| Stretched    |  | The image is stretched to occupy the entire surface of the control. The size of the image is adapted to the size of the control.  |
| Tiled        |  | The image, in its initial size, is repeated as many times as possible to occupy the entire surface of the image control.  |
| Homothetic   |  | The image is proportionally enlarged so that it is entirely displayed in the image control. The image is aligned to the top and to the left.  |

|   |   |  |
|---|---|--|
| Homothetic extended                     |  | The image is proportionally enlarged so that the smallest side of the image is entirely displayed in the image control.  |
| Homothetic centered                     |  | The image is stretched or reduced (according to the case) while keeping the width/height ratio to occupy the greatest possible surface of the control. The image is not distorted. |
| Homothetic centered extended            |  | The image is proportionally enlarged so that the smallest side of the image is entirely displayed in the Image control and it is centered in the image control.                    |
| Homothetic without enlargement          |  | The image can be reduced if necessary but it will not be enlarged above its initial size if the control is too large.  |
| Homothetic centered without enlargement |  | The image is centered and it can be reduced if necessary but it will not be enlarged above its initial size if the control is too large.   |

WebDev supports the GIF images with a transparent background (dynamic page only). In this case, you must:

- create an image whose background color is "Light Magenta" (RGB color: 255, 0, 255). The magenta is automatically considered as being the transparency color.
- create an image whose color of the top left pixel is the reference color for the transparency. You can define the color that will be the "transparent" color.



**Caution!**

The "transparent" color is applied to the entire image. Therefore, transparent areas may appear in the middle of the image. This may cause unexpected effects so don't forget to check the images!

### Actions during the click on the image ("General" tab)

During the click on an image, you have the ability to trigger an action (run the code of a button for example).

The "Other actions" button is used to define preset actions during the click on the image.

You also have the ability to choose the target of the action (new browser, current page, ...).

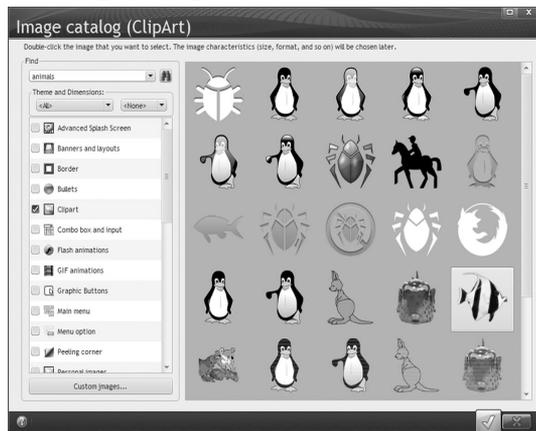
## The image catalog

As soon as an image can be displayed in a control or in a page, WebDev proposes to use the image catalog via the "Catalog" button. This catalog contains hundreds of images, clip-parts, ...

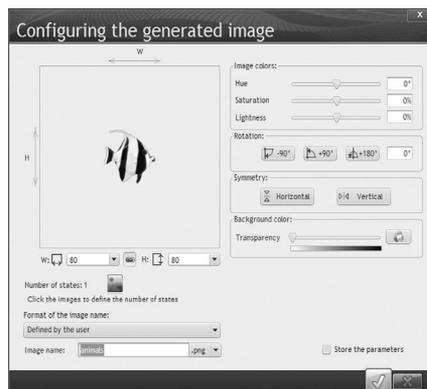
To select a clipart, you have the ability to perform a search according to a keyword.

In the search window:

- Enter a keyword ("Animals" for example)
- Specify (if necessary) the theme, the dimensions and the category.
- Click the search button (the magnifier). The images related to the keyword are displayed in the window.
- Click the requested image (a penguin for example) and validate.



- Specify the characteristics of the image to generate: size, lightness, extension, ...



The name of the specified image is displayed in the description window of the image and the image file (in the requested format) is copied into the project directory.

## Other characteristics

The description window of the image control is also used to define:

- **the border of the image** (outline). In the "Style" tab, "Border/Background" element, you have the ability to configure the thickness of the border as well as the horizontal and vertical margins inside the border. All these values are set to 0 by default (no border).
- **the dimensions of the image**, the width and height of the image. These characteristics are also specified in the "GUI" tab.
- the **alternative text**: In the "Details" tab, this option is used to display a text if the image cannot be displayed by the browser (image not found, format not supported).



Tip

To improve the referencing of your pages, an alternative text should be entered in all your Image controls.

## Type of control: Clickable image



*(static page, semi-dynamic page and dynamic page)*

## Summary

The Clickable Image controls are used to display "reactive" images. You have the ability to:

- define the position of the mouse during a click on the image ("click area"). This option is available for the dynamic pages only.
- define areas on the image ("Map Area").

## Practical example

- ▶ To view the different features proposed by the Clickable Image controls, open the "PAGE\_MapAreaImage.WWH" page.
- ▶ Run the test of the page.
- ▶ Go back to the editor.

## Characteristics of the Clickable Image controls

The characteristics of the Clickable Image controls are the same as the characteristics of the image controls. A new option is the operating mode.

The operating mode allows you to define the type of clickable image to use:

- **Clickable image**: the image is a simple clickable image. An action can be associated with the click on the image. However, you cannot retrieve the coordinates of the mouse or define click areas.
- **Click area** (dynamic page only): the image is a clickable image and the coordinates of the mouse can be retrieved (**PageParameter**).

Let's see an example of server code used to retrieve the coordinates of the mouse during a click on a click area:

```
// Retrieves the coordinates of the mouse
x is int = MouseXPos()
y is int = MousYPos()
// Informs the Web user
Info("You've clicked at position X=" + x + "; Y=" + y)
```

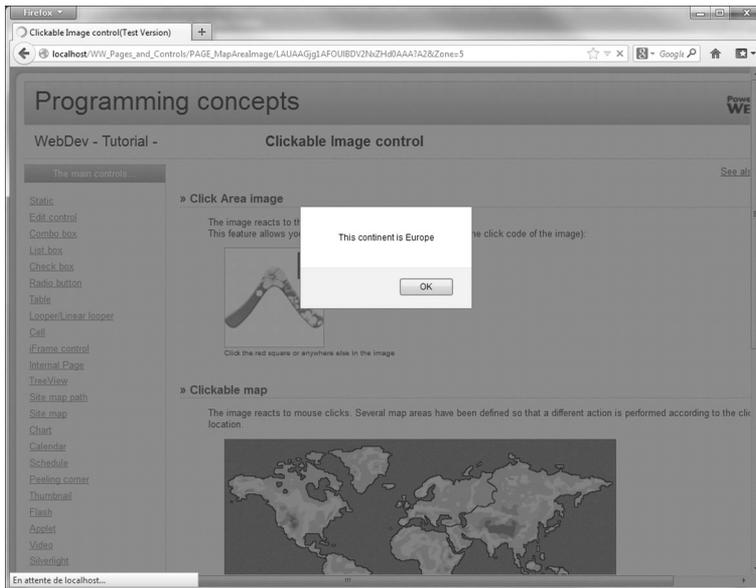
- **Map area:** clickable image with definition of clickable map areas. An action can be associated with the click on the image and on each map area.

Note: The image behaves like a button. A browser action and/or a server action can be performed.

### Example for managing a Map Area

To define map areas on the clickable image, select "Map area" in the "General" tab of the description window of the clickable image. You will have the ability to define the different areas of the image. For these areas, the action that will be performed during a click differs from the action of the image.

- ▶ Run the test of the "PAGE\_MapAreaImage" page.



- ▶ To draw a map area:
  1. Right-click the Image control.
  2. Select one of the following options from the popup menu:
    - "Insert a map area .. A rectangle": to define simple fixed areas.
    - "Insert a map area .. A circle": to define rounded areas.
    - "Insert a map area .. A polygon": to define complex areas, like a map.
  3. Position the mouse at the location where the area must be drawn. Keep the left mouse button down and draw the shape of the map area.

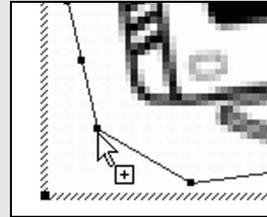
To precisely select the points of the area, you have the ability to work in zoom mode (use CTRL + Wheel to modify the zoom in the editor).



Tip

If you choose to draw a "polygon" area, you have the ability to add points even once the area is defined. To do so, you must :

1. Keep the [CTRL] key down.
2. Click with the mouse at the specified location on the border of the area (a "+" sign is displayed below the mouse cursor).



Several map areas can be defined on the same clickable image. Different processes can be run according to the location clicked by the Web user.

**Type of control: Radio button**



*(static page, semi-dynamic page and dynamic page)*

### Summary

Radio buttons are used to select a single option among the proposed ones.

The caption of the options must be entered when describing the radio button in the editor. They can be modified by programming (server code).

Your choice:

- Coffee with sugar
- Coffee without sugar
- Tea
- Chocolate

### Practical example

- ▶ Open the "PAGE\_RadioButton.WWH" page.
- ▶ Run the page test and click the different buttons, modify the radio button. See what happens. The code of each button is displayed in the page.
- ▶ Go back to the editor.

### Characteristics of radio buttons

An integer (that can take a value from 1 to N, N being the total number of options found in the radio button) is associated with each option found in the radio button. The first option is the option #1, the second option is the option #2, ... The options are numbered from top to bottom and from left to right.

In our page: "RADIO\_Choice = 2" means that "Coffee without sugar" is selected, "RADIO\_Choice = 3" means that "Tea" is selected and "RADIO\_Choice = 4" means that "Chocolate" is selected, ...

To initialize a radio button, an integer value must be assigned to the control (server code and/or browser code):

```
RADIO_Choice = 3 // Selects the option 3: "Tea"
```

To find out the selected option, all you have to do is retrieve the integer value of the control (server code and/or browser code):

```
n is int = RADIO_Choice
```

### Type of control: Check box



*(static page, semi-dynamic page and dynamic page)*

### Summary

Check boxes are used to specify whether the value of each option is "True" ("box checked") or "False" ("box unchecked").

The caption of the options must be entered when describing the check box in the editor. It can be modified by programming (server code).

You like to:

Dance

Read

Sing

Do sport

### Practical example

- ▶ Open the "PAGE\_CheckBox.WWH" page.
- ▶ Run the page test and click the different buttons, modify the check box. See what happens. The code of each button is displayed in the page.
- ▶ Go back to the editor.

### Characteristics of check boxes

An integer (that can take a value from 1 to N, N being the total number of options found in the check box) is associated with each option found in the check box. The first option is the option #1, the second option is the option #2, ... The options are numbered from top to bottom and from left to right:

- "CBOX\_Choice[2]=True" means: "Read" is checked
- "CBOX\_Choice[3]=False" means that "Sing" is not checked

When several options are found, the check box behaves like an array of options.

To access an option, all you have to do is pass the subscript of the option by using the syntax with square brackets [ ].

To initialize a check box, the corresponding choice must be initialized with "True" or "False" according to the case (server code and/or browser code).

```
CBOX_Choice[2] = True // To check option 2 "Read"
```

To find out the value of an option, all you have to do is retrieve the value of the control ("True" or "False", server code and/or browser code).

```
IF CBOX_Choice[2] = True THEN
    // Option checked
ELSE
    // Option not checked
END
```



Tip

The programmers who are used to handle the boolean expressions may omit to run the test with "True" or "False":

```
IF CBOX_Choice[2] THEN
  // Option checked
ELSE
  // Option not checked
END
```

The caption of a check box or radio button can be modified by **..Caption** (server code). For example:

```
// Changes the caption of the check box
CBOX_Choice[..Caption = "You prefer to"
// Changes the 3rd option
CBOX_Choice[3]..Caption = "Dance at night"
// Changes the 1st and 3rd options
CBOX_Choice[1]..Caption = "Read a book"
CBOX_Choice[3]..Caption = "Dance the twist"
```

However, no option can be added by programming.

**Type of control: List box**



*(static page, semi-dynamic page and dynamic page)*

## Summary

The list boxes are always expanded. This is the reason why they are also called "drop-down list boxes". They are used to display several elements and to select one or more elements.

When creating a "List Box" control, a wizard starts and proposes several types of list boxes:

- a list box whose content is linked to a file or to a query (browsing list box available in the dynamic or semi-dynamic pages).
- a list box whose content is linked to a variable (list box on source).
- a list box whose content is defined in the editor or by programming (memory list box).

First names:

|               |
|---------------|
| Anthony       |
| Celine        |
| <b>Fabian</b> |
| Frederick     |
| John          |
| Julia         |
| Karen         |
| Luke          |
| Mary          |
| Pascal        |



Caution !

**Rules for defining the type of list box to create:**

- Use a list box linked to a file or to a query when you want to propose to the user a list of values that are contained in a data file. If the list box is linked to a data file, the entire content of the data file is directly proposed. If the list box is linked to a query, the list box will contain a selection of records found in the data file.
- Use a list box whose content is defined in the editor if the values are "STATIC". They do not change during the execution of the program.
- Use a list box filled by programming if the content of the list box is the result of calculations or if the elements that must be included in the list box cannot be selected by a query.
- Use a list box whose content is linked to a variable when the content of the list box comes from calculations performed via a variable.
- The content of list boxes must be as small as possible. Indeed, the more elements are found in the list box, the longer the page will take to load!

**Practical example**

- ▶ Open the "PAGE\_ListBox.WWH" page.
- ▶ Run the test of this page.
  - The 3 lists boxes at the top are single-selection: memory list box, browsing list box and list box on source.
  - The list box at the bottom is a multi-selection memory list box.
- ▶ Click the different buttons and see what happens.
- ▶ At run time, in a multi-selection list box (according to the Internet standard):
  - to select several elements one by one, press the [CTRL] key and the left mouse button.
  - to select several elements at once, use the "lasso". Click with the mouse, keep the mouse button down, move the mouse to select the elements, release the mouse button.
  - to select several consecutive elements at once, click the first element, press the [Shift] key, keep it down and click the last element of the selection.
- ▶ Go back to the editor.

**Mode for filling a list box**

The wizard for creating a list box proposes to create several types of list boxes:

- list box linked to a data file or to a query (called browsing list box)
- list box filled by programming (called memory list box)
- list box filled from a variable (called list box on source)

Each type of list box corresponds to a specific fill mode.

### Filling a memory list box

Two methods can be used to fill a memory list box:

- in the editor,
- by programming.

**In the editor:** the content of a list box can be defined in the description window of the control ("General" tab, "Initial content" control). Several values can be entered. Press the [Enter] key to go to the next row.

To fill a list box **by programming**, you must use **ListAdd**. For example:

```
// Adds Emma into LIST_Memory  
ListAdd(LIST_Memory, "Emma")
```

- ▶ In the practical example, the initialization of the memory list box is performed in the "Fill the list box" button.

### Filling a list box from a data file or from a query (semi-dynamic page or dynamic page)

For a browsing list box, the list box is filled from the elements specified when creating the list box. This information is also available in the "Content" tab. This information is as follows:

- the data file or the query that will be used as data source for the fill operation.



Notes

If the query does not exist when creating the control, you have the ability to create the query at the same time as the control. The query will be included in the window that contains the control.

Caution: if you are using an embedded query, this query will be used for this control only. It cannot be re-used in your project.

This data file or this query will be automatically read without having to write a single code line.

- the item to display in the list box.
- the search key used: this item will be used for sort when filling the list box; it specifies the order in which the values will be displayed in the list box.
- the retrieved value: when selecting an element of the list box, the retrieved value can be:
  - the subscript of the selected row,
  - a value of the data file.

In most cases, an item of the data file is chosen as retrieved value. It is the unique key in most cases, giving you the ability to perform searches in the data files for example.

► In the practice example:

1. Display the description window of the list box on file ("Description" from the popup menu).
2. In the "Content" tab, check the characteristics of the browsing list box.



Notes

### The different types of browsing list boxes

Two types of browsing list boxes are available:

- Browsing list box "Loaded in memory": The content of the list box is loaded in memory, then it is displayed in the list box. The access to the different elements of the list box is faster. This mode is reserved for the data files containing less than 10 000 records.
- Browsing list box with "Direct access (no limit)": The list box directly displays the content of the data file or query. The current record corresponds to the selected element. Each move performed in the list box involves an access to the linked data file. This type of list box is reserved for the data files containing more than 10 000 records.

### Filling a list box on source

A list box can be filled with the information found in a WLanguage array for example. The array is a global variable of the page:

```
Array_Month is array of 12 strings
Array_Month = ["January", "February", "March", "April", ...
              "May", "June", "July", "August", "September", ...
              "October", "November", "December"]
```

For a list box on source, the list box is filled from the elements specified when creating the list box. This information is also available in the "Content" tab. This information is as follows:

- the name of the source variable,
- the variable to display in the list box,
- the stored variable.

► In the practice example:

1. Display the description window of the "Month" list box ("Description" from the popup menu).
2. In the "Content" tab, check the characteristics of the list box on source.

### Characteristics of a list box

Once the list box is created, its characteristics can be modified in the description window of the control ("Description" from the popup menu). The different tabs are used to specify all the options found in the list boxes.

For example, the "Details" tab is used to:

- define whether the list box is multi-selection.
- define whether the list box is sorted, ...

## Programming a list box

The method for programming the List Box controls is very easy in WebDev: all you have to do is use the WLanguage functions that start with LIST. These functions can be used on the memory list boxes, browsing list boxes or list boxes on source.



Notes

You don't know whether the function can be used on a memory list box, on a browsing list box or on a list box on source? See the online help! To do so, press F1 on the name of the function.

### Modifying a list box (memory list box only)

Two methods can be used to modify an element found in a list box (server code):

- **ListModify**
- the direct assignment by using the subscript of the element to modify.

```
ListModify(LIST_Memory,"Emma") // Modifies the current element
ListModify(LIST_Memory,"Emma",1) // Modifies the element 1
LIST_Memory[1] = "Emma" // Modifies the element 1
```

Note: To access an element in a list box, its subscript must be specified. The subscript of the first element is set to "1".

### Retrieving the selected element

In a single-selection list box, to retrieve the selected element, you must know the subscript of the selected element. **ListSelect** returns the subscript of the selected element. If no element is selected, **ListSelect** returns "1".

```
Subscript is int
FirstName is string
Subscript = ListSelect(LIST_Memory)
FirstName = LIST_Memory[Subscript]
// Retrieves the selected element
// Or
FirstName = LIST_Memory[LIST_Memory]
```

Note: for a browsing list box, if the retrieved value corresponds to a value of the data file, all you have to do is use the name of the list box to find the selected value.

In a multi-selection list box, to retrieve all the selected options, you must use **ListSelect**. As long as **ListSelect** does not return "-1", this function returns the subscript of the selected option.

```
i is int = 1
s is string
n is int = ListSelect(LIST_Multi, i)
WHILE n <> -1
  s = s + CR + LIST_Multi[n]
  i++
  n = ListSelect(LIST_Multi, i)
END
Info("The selected options are: ", s)
```



Notes

In server code, you have the ability to use a FOR EACH loop. The code becomes:

```
s is string
FOR EACH SELECTED ROW OF LIST_Multi
    s = s + CR + LIST_Multi[LIST_Multi]
END
Info("Selected option: ", s)
```

To find out the number of selected elements, you must use **ListSelectCount**.

To find out the number of elements in the list box, you must use **..Occurrence** or **ListCount**:

```
IF LIST_Memory..Occurrence = 0 THEN // Empty list
```

#### Deletion (in the memory list boxes only)

To delete a value, you must use **ListDelete**.

To delete all the values from the list box, you must use **ListDeleteAll**.

```
// Deletes the selected element
ListDelete(LIST_Memory)
```

#### Search (server code/browser code)

To find an element in a list box, you must use **ListSeek**.

```
IF ListSeek(LIST_Memory, Value) = -1 THEN
    // does not exist
ELSE
    // already exists
END
```

### Type of control: Combo box



(static page, semi-dynamic page and dynamic page)

### Summary

A combo box is also called "drop-down list". When creating a "Combo box" control, you have the ability to create:

- a "Browsing" combo box whose content is linked to a data file or to a query (dynamic or semi-dynamic page).
- a "Memory" combo box whose content is defined in the editor or by programming (server code).
- a combo box on source, linked to a WLanguage variable.



The same rules as the ones applied to the list boxes will be used to choose the fill method.

The combo box is expanded when the control is selected.



Caution!

Unlike the list boxes, the combo boxes are not multi-selection: a single element can be selected in the combo box.

This paragraph only presents the features specific to the combo boxes. See the paragraph about the list boxes for more details regarding the operating mode of combo boxes.

### Practical example

- ▶ Open the "PAGE\_Combobox.WWH" page in the editor.
- ▶ Run the test of this page. This page contains:
  - a sorted memory combo box,
  - a browsing combo box,
  - a combo box on source.
- ▶ Click the different buttons and see what happens.
- ▶ Go back to the editor.

### Characteristics of the combo box

Once the combo box is created, its characteristics can be modified in the description window of the control ("Description" from the popup menu).

All the characteristics specified in the wizard are available as well as several other interesting options.

The "Details" tab is used for example to define whether the combo box is sorted.

Type of control: **Table**



*(semi-dynamic page and dynamic page)*

### Summary

Do not confuse "Table" and "Table control".

We talk of **Table** for a SQL database.

A **Table control** is used to view the data in a table. In this case, the data can be entered or it can come from a data file (or from a table). Got it!

The table control of WebDev is used to view or modify the data:

- in memory: we talk of **memory table** or table filled by programming.
- coming from data files or queries: we talk of **browsing table**.
- coming from WLanguage variables: we talk of **table on source**.

The tables allow you to view several data in a single page.

The method for managing the tables depends on the type of the table (memory table, browsing table or table on source). The type of the table is defined in the wizard when creating the table.

We are going to present the different types of Table controls.

### Practical example

- ▶ Open the "PAGE\_Table.WWH" page and run its test. This page presents the processes performed on the memory tables and on the browsing tables.

## Mode for filling the table

A table can be a browsing table (linked to a data file or to a query), a memory table or a table on source.

### Filling a memory table

A memory table can be filled by programming with **TableAddLine**. This function accepts the name of the table and the content of the row in parameter. The content of the row is given column by column: each column value is separated by "a comma".

```
// Add an entire row
TableAddLine(TABLE_MEM, "Monday", "Scrambled eggs", ...
    "Irish stew", "Vanilla ice cream")
```

### Table linked to a data file or a to a query

The following information must be specified when creating a table control filled from a data file or from a query:

- the data file or the query that will be used as data source for the fill operation.



Notes

If the query does not exist when creating the control, you have the ability to create the query at the same time as the control. The query will be included in the page that contains the control.

Caution: if you are using an embedded query, this query will be used for this control only. It cannot be re-used in your project.

This data file or this query will be automatically read without having to write a single code line.

- the items displayed in a table row.
- the search key used: this item is used to sort the table as it is filled and therefore it specifies the order in which the values will be viewed in the table.

For example, the products as well as their caption can be displayed in a table.



Notes

### The different types of browsing tables

Two types of browsing tables are available:

- Browsing table "Loaded in memory": The content of the data file is loaded in memory and displayed in the table. The access to the different file elements is faster. The data being loaded in memory, all the table columns can be sorted. This mode is reserved for the data files containing less than 10 000 records.
- Browsing table with "Direct access (no limit)": The table displays the content of the data file or query directly. The current record corresponds to the selected element. Each move in the table involves an access to the linked file. This type of table is reserved for the files containing more than 10 000 records.

## Characteristics of the table

Once the table is created, its characteristics can be modified in the description window of the control ("Description" from the popup menu).

All the characteristics specified in the wizard are available as well as several other interesting options.



Notes

The description window of a table includes two sections:

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

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

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

Some interesting information:

- The tables can be single-selection or multi-selection ("Details" tab of the table description).
- The "Style" tab is used to define the colors of the odd and even rows, the color of the selected row, the image displayed when loading the table (if the table is in Ajax mode)

Regardless of the table type (browsing table, memory table or table on source), the columns (in Ajax mode) can be:

- adjusted by the user (with the mouse)
- with search: a magnifier will be displayed, allowing you to find a value in the column.
- sortable. Arrows will be displayed in the title of columns, specifying that the column can be sorted and indicating the sort direction.



Notes

The magnifier or the sort arrow are visible on all the columns found in a memory table, in a table on source or in a browsing table loaded in memory. The magnifier and the arrow are visible on the columns linked to a key (index) for the browsing tables with "Direct access (no limit)".

See the online help for more details.

- link ("Details" tab of the column description). You have the ability to define the action of the link and its destination.

## Programming a memory table

In a memory table, each table row is identified by a subscript.

To handle a row or a column, the subscript of the relevant row must be specified. If the subscript is not specified, the selected row is affected.

- A memory table can be handled row by row or column by column.
- A memory table can be in edit or read-only.
- A memory table can be single-selection or multi-selection.

The method for programming the Table controls is very easy in WebDev: all you have to do is use the WLanguage functions that start with TABLE. These functions can be used on the memory tables, tables on source and/or browsing tables.



Notes

You don't know whether the function can be used on a memory table, on a table on source or on a browsing table? See the online help! To do so, press F1 on the name of the function.

We will now present some of the functions used to handle a memory table:

- **TableModifyLine** modifies a table row (either the current row, or the row whose subscript number is specified).
- **TableSelect** returns the subscript of the current row (selected row) or the subscript of the selected rows for a multi-selection table.
- **TableSelectPlus** is used to select a table row by programming according to its subscript and/or to select several rows.
- **TableDelete** is used to delete a table row.
- **TableDeleteAll** clears the entire memory table.

## Tips for the tables

- ▶ Run the test of the "PAGE\_Table.wdw" page. At the top right of the tables, you will notice that the icon  is displayed. This icon is used to display a popup menu for the table. This popup menu is used to export the table in Word, Excel, XML and PDF format.

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

**Type of control: Looper***(semi-dynamic page and dynamic page)***Summary**

The loopers are used to repeat "n" times a set of controls. During this repetition, various information can be displayed in the controls found in the looper. This enables you to display, in each repetition, the records found in a database.

Several types of loopers are available:

- **Memory looper:** the looper is filled by programming.
- **Browsing looper:** the looper is automatically filled from a data file or from a query, without a single code line.
- **Looper on source:** the looper is filled from a WLanguage variable without a single code line.

For each control found in the looper, several characteristics can be modified during the repetition: value, caption, text color, status, ...

The looper can be moved like a control. When it is moved, all the controls associated with the looper are moved at the same time.

When creating a looper, you have the ability to define the number of columns that will be displayed. This option is used to get an horizontal repetition that does not exceed the specified number of columns. When the number of columns is reached, a new vertical repetition is added.

**Practical example**

- ▶ Open the "PAGE\_Looper.WWH" page and run its test. This page uses a memory looper that lists the adverts.

The method for managing the loopers differs for:

- a memory looper,
- a looper linked to a data file or to a query.
- a looper on source.

Let's take a look at the different types of loopers.

**All the details****Memory looper (filled by programming in server code)**

A looper can contain several controls (static control, edit control, image, ...). Each control found in the looper can be associated with an attribute.

This attribute is used to modify a property of the associated control (its value, its caption, its color, its visibility, ...). You also have the ability to create an attribute in the looper without associating this attribute with a control: we talk of free attribute (this attribute can contain the value of a counter for example).

A memory looper behaves more or less like a Table control.

In a memory looper, each looper row is identified by a subscript.

To add a row into a looper (server code), use **LooperAddLine**. This function expects in parameter:

- the name of the looper,
- the values of the attributes (in order) separated by commas:

```
// Add a row into the looper
LooperAddLine(LOOP_Advert, ...
    "Advert #1", ...
    "IMAGE1.JPG", ...
    "For sale GOLF GTI 3." + CR + "Year 1998" + CR + ...
    "Very good condition. M.O.T. OK." + CR + ...
    "5 doors, leather seats, tinted glasses." + CR + ...
    "Black, aluminum wheel." + CR + ...
    "38000KM. € 9000 negotiable.", 1, 1)
```

**LooperModifyLine** modifies a looper row (server code).

**LooperDelete** is used to delete a looper row (server code).

#### **Looper linked to a data file or to a query**

When creating the browsing looper, you must specify the data file or the query that will be used as data source for the fill operation.

This data file or this query will be automatically read without having to write a single code line.

The selected search key is used for sort when filling the looper, so it is used to specify the order in which the values will be viewed. For example, a looper can be used to display:

- the photo of a product
- the description of the product
- the price of the product
- ...

The characteristics of a looper are available in the description window of the looper ("Description" from the popup menu).

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

Type of control: Tree-View



*(semi-dynamic page and dynamic page)*

### Summary

A "TreeView" control is similar to a list box whose content is organized hierarchically. In other words, the data is organized in the same way as in the file explorer of Windows.

### Practical example

- ▶ Open the "PAGE\_TreeView.WWH" page and run its test.
- ▶ Click the different links of the TreeView control and see what happens.
- ▶ Go back to the editor and study the code of the control.

### All the details

The TreeView control is managed by programming. The WLanguage functions used to program a treeview start with *TreeXXX*.

These functions are similar to the functions for managing the tables filled by programming. For example, *TreeAdd*, *TreeDeleteAll*, *TreeDelete*, *TreeSelect*, ...

The TreeView control can also be handled row by row to modify the properties of a row. You have the ability to use:

- *..Color* to modify the color of a row,
- *..BrushColor* to modify the background color of a row,
- *..Font* to modify the characteristics of the font for a row, ...

We won't go into details about the functions for managing the treeviews. See the online help (keyword: "TreeView, Handling a treeview by programming") for more details.

Type of control: Cell



*(static page, semi-dynamic page and dynamic page)*

### Summary

A cell is used to group several controls in a box. A cell can be used in addition to areas, to create an insert in a page for example.

A cell control is a container. A cell can contain several controls (of any type).

### Practical example

- ▶ Open the "PAGE\_Cell.WWH" page and run its test.

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

**Type of control: Chart****(static page, semi-dynamic page and dynamic page)****Summary**

The Chart control is used to include a chart in a page. The data source of this control can be:

- defined by programming,
- a data file or a query,
- a table column,
- a List Box control,
- a WLanguage array.

Several types of charts can be created in 2D or in 3D: Pie, Column, Stock, Line, Scatter, ...

The display options are used to produce different visual effects for each type of chart.

**Practical example**

- ▶ Open the "PAGE\_Chart.WWH" page and run its test. This page uses a Chart control. The data found in this chart is defined by programming with *grAddData*.  
The description window of the Chart control is used to define the main parameters of the control.

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

**Type of control: Rating****(static page, semi-dynamic page and dynamic page)****Summary**

The Rating control allows:

- the Web user to give a mark.
- the site to display a rate.

The control uses a star by default but any other image can be used.

**Practical example**

- ▶ Open the "PAGE\_Rating.WWH" page and run its test. This page uses a Rating control.

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

## Specific controls

---

The "Specific controls" group the controls that are used for a specific feature. These controls are as follows:

- Calendar
- Schedule
- Calendar
- Treeview table
- Captcha
- Bar code
- HTML table
- Slider
- Row
- HTML control
- Web Camera
- Java applet
- Flash and Flex
- Silverlight
- iFrame
- Site map path
- Site map
- Pager
- Thumbnail

We are going to study these different types of controls. The "WW\_Pages\_and\_controls" project contains an example for each type of control. We advise you to run the test of the page associated with the control, to study the code found in the page and to see the online help for more details. These pages are grouped in the "Specific controls" folder in the project explorer.

**Type of control: Calendar**

*(static page, semi-dynamic page and dynamic page)*

**Summary**

The Calendar control is used to create a calendar. This calendar allows the Web user to select a date.

**Notes**

When describing a Date edit control, you also have the ability to display a Calendar button. This button allows the Web user to select the requested date.

**Practical example**

- ▶ Open the "PAGE\_Calendar.WWH" page and run its test. This page presents the different modes for using a calendar.

**Type of control: Scheduler**

*(static page, semi-dynamic page and dynamic page)*

**Summary**

The Scheduler control is used to manage the display of multiple resources and it is found in several sites.

**Practical example**

- ▶ Open the "PAGE\_Scheduler.WWH" page and run its test. This page presents the different modes for using a schedule. The "Global declarations of the page" process contains the initialization code of the Scheduler control.

**Type of control: Organizer**

*(static page, semi-dynamic page and dynamic page)*

**Summary**

The Organizer control is used to display and handle the organizers. The display can be done on a daily, weekly or monthly basis.

**Practical example**

- ▶ Open the "PAGE\_Organizer.WWH" page and run its test.

**Type of control: TreeView Table**

*(static page, semi-dynamic page and dynamic page)*

**Summary**

A treeview table is a cross between a Table control and a Treeview control.

This control is based on the structure of a Table control (column, row, cell) but it also contains a TreeView column that introduces the notion of tree structure.

**Practical example**

- ▶ Open the "PAGE\_TreeViewTable.WWH" page and run its test.  
The buttons found in this page are used to perform different processes on the table.

**Type of control: Captcha**

*(static page, semi-dynamic page and dynamic page)*

**Summary**

The "captcha" are the distorted texts that the Web user must decode and re-enter in a site control before validating a form.

The purpose of the "captcha" is to prevent hackers from attacking a site via a robot that would enter cascading information.

By implementing this type of security in a form, you can be almost sure that the information was entered by a human being!

**Practical example**

- ▶ Open the "PAGE\_Captcha.WWH" page and run its test.  
The "New Captcha" button is used to create a captcha to identify (**CaptchaDisplay**). The "Check" button is used to check the content of the captcha and the content of the edit control (**CaptchaVerify**).

**Type of control: Bar Code**

*(static page, semi-dynamic page and dynamic page)*

**Summary**

The Bar Code control is used to display a bar code in a page. All the types of bar codes can be used. It can be a bar code intended to be printed or scanned by a cell phone (QR Code).

**Practical example**

- ▶ Open the "PAGE\_BarCode.WWH" page and run its test. This page presents the different types of bar codes that can be displayed.

**Type of control: HTML table***(static page, semi-dynamic page and dynamic page)***Summary**

An HTML table is a positioning table used to position the controls in the page. Like any cell control, an HTML table is a container and it is used to avoid the unexpected effect when moving a group of controls downwards when a looper or a table is located beside this group of controls.

Each part of the table can be compared to a cell control and it can contain several controls of any type.

**All the details**

The HTML table can be adapted to suit your own requirements. The table cells can be split or merged in order to organize the different controls included in the HTML table.

We won't go into details about the HTML tables in this tutorial. See the online help (keyword: "HTML table") for more details.

**Type of control: Slider***(static page, semi-dynamic page and dynamic page)***Summary**

A Slider control corresponds to a graphic object used to make a value change. The current slider value can be retrieved by programming in a variable and it can be used in calculations.

**Practical example**

- ▶ Open the "PAGE\_Slider" page and run its test.

**Type of control: Horizontal rule***(static page, semi-dynamic page and dynamic page)***Summary**

The horizontal rules are used to draw a line or to define a block of color (a line to separate two groups of edit controls in a page for example).



Notes

We advise you to use the "Border" objects available in the catalog of controls of the "Wizards, Examples and Components" pane.

We won't go into details about this type of control in this tutorial. See the online help (keyword: "Horizontal rule") for more details.

**Type of control: HTML***(static page, semi-dynamic page and dynamic page)***Summary**

The HTML control is used to include HTML code in a WebDev page. This HTML code can correspond to:

- a set of standard HTML tags,
- Javascript or VBScript scripts, ...

**Practical example**

- ▶ Open the "PAGE\_HTMLControl.WWH" page and run its test.  
This page is used to display the content of another site in an HTML control.

**Type of control: Web Camera***(static page, semi-dynamic page and dynamic page)***Summary**

A Web Camera control is used to view a video source (by sequence of images) coming from an external camera connected to the server or to any computer accessible from the server.

See the online help (keyword: "Web, Web Camera control") for more details.

**Type of control: Java Applet***(static page, semi-dynamic page and dynamic page)***Summary**

A Java Applet control is used to include a Java© applet in a WebDev page.

In most cases, this Java© applet (file with ".class" extension) is found in the directory of the project images ("`<Project Name>_WEB`" directory).

**Caution:** Make sure that all the necessary files (".class" files, image files, etc.) and the applet are positioned at the same location.

**Practical example**

- ▶ Open the "PAGE\_AppletControl.WWH" page and run its test.
- ▶ Go back to the editor and display the control description ("Description" from the popup menu).

In the "General" tab, the Java Applet control includes several characteristics specific to the Java applets:



- **Applet:** Name of the applet file (".class" extension). Caution: the name of the file is case sensitive (Java® is case sensitive).
- **Database:** Directory where the files required by the Java® applet are installed (".class" files, images, text files, ...). In most cases, this directory corresponds to the directory of the project images: "<Project Name>\_WEB".
- **Parameters and values:** list of parameters expected by the applet and values of these parameters.

**Type of control: Flash and Flex**



**(static page, semi-dynamic page and dynamic page)**

## Summary

A Flash control is used to include a Flash® animation in a WebDev page.

Similarly, the Flex control is used to display a Flex animation in a WebDev page.

In most cases, this Flash® or Flex animation (file with ".SWF" extension) is found in the directory of the project images ("<Project Name>\_WEB").

Note: The Flash® or Flex animations are built on the same architecture.



Notes

**Caution:** depending on the browser used, some Flash® or Flex animations may not operate.

## Practical example

- ▶ Open the "PAGE\_FlashControl.WWH" page and run its test. This page explains how to use the Flash control and how this control can interact with the WebDev page.
- ▶ Go back to the editor and display the control description ("Description" from the popup menu). In the "General" tab, the Flash control (or the Flex control) includes several specific features :



- **Paused at startup:** the animation is loaded and paused.
- **Read in loop:** the animation is played in loop.
- **Display the menu:** the default popup menu of Flash© animations is enabled via the right mouse click.
- **Device font:** the fonts used to display text in the Flash© animation will be the closest fonts found on the computer of the Web user.
- **Display the warning messages:** if this option is checked, the warning boxes concerning the Flash© plug-in will be displayed.
- **Allows the 'full screen' mode:** if this option is checked, the Web user will be able to display the Flash© animation in full screen mode.
- **Quality:** specifies the quality of display for the Flash animation©.
- **Scale:** indicates how the Flash© animation must be resized when the browser window is resized.
- **Window mode:** specifies how the Flash© animation must be displayed.
- **Alignment:** configures the alignment of the animation in the Flash control.

You can make a Flash© control interact with a WebDev page. To do so, check "Control accessible from Javascript" in the "Details" tab of the description window of the Flash control:



### Type of control: Silverlight



*(static page, semi-dynamic page and dynamic page)*

#### Summary

The Silverlight 2 control is used to include Silverlight 2 applications in a WebDev page.

#### Practical example

- ▶ To understand the operating mode of the Silverlight control, open the "PAGE\_Silverlight.WWH" page and run its test.  
We won't go into details about this type of control in this tutorial. See the online help (keyword: "Silverlight control") for more details.

### Type of control: IFrame



*(static page, semi-dynamic page and dynamic page)*

#### Summary

iFrame is an HTML concept: this concept indicates that an independent "WEB" content is inserted into a specific area of the current page.

For example, in a WebDev page, you have the ability to display:

- another page of the current site
- a page from another site.

#### Practical example

- ▶ Open the "PAGE\_IFrame.WWH" page and run its test.  
We won't go into details about this type of control in this tutorial. See the online help (keyword: "IFrame control") for more details.

**Type of control: Site map path****(static page, semi-dynamic page and dynamic page)****Summary**

The site map path allows the user to navigate through the site pages via a menu such as "Device >> Photo >> Digital".

**Practical example**

- ▶ Open the "PAGE\_SiteMapPath.WWH" page and run its test.  
We won't go into details about this type of control in this tutorial. See the online help (keyword: "Site Map Path control") for more details.

**Type of control: Site map****(static page, semi-dynamic page and dynamic page)****Summary**

The Site Map control allows the Web user to:

- have a global view of the pages found in the site.
- access a specific page directly.

The "Site Map" control is also used to improve the referencing of a site.

**Practical example**

- ▶ Open the "PAGE\_SiteMap.WWH" page and run its test.  
We won't go into details about this type of control in this tutorial. See the online help (keyword: "Site Map") for more details.

**Type of control: Pager****(semi-dynamic page and dynamic page)****Summary**

A looper control (or a table control) can contain several records. It may be sensible not to display all the records at the same time, because this could slow down the page display and infuriate the Web user! The pager is used to browse the records page by page.

For each looper or table control, you have the ability to define the maximum number of rows per page ("General" tab in the description window of the looper or "Details" tab in the description window of the Table control).

If the number of rows in the table or in the looper is greater than the maximum number of rows per page, a pager can be associated with the table or with the looper.

The creation of a pager is proposed when creating the looper or the table. The pager can also be created once the table or the looper was created.

Example: The pagers are used to "divide" the display of a loop:

| List of Products  |  |
|---|--|
| Product Description   | Gum  |
| Unit Price Before Tax   | 0,50 € <input type="button" value="Delete"/>   |
| Product Description   | Blue pen                                       |
| Unit Price Before Tax   | 0,40 € <input type="button" value="Delete"/>   |
| Product Description   | Pencil   |
| Unit Price Before Tax   | 0,25 € <input type="button" value="Delete"/>   |
| Product Description   | "Gold" pen                                     |
| Unit Price Before Tax   | 158,00 € <input type="button" value="Delete"/> |
| <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value=""/> > <input type="button" value=""/> >> |  |

In this example, 32 records have been added into a loop. The maximum number of rows per page is set to 5. A pager was associated with the loop in order to display 5 records per page (over 7 pages).

### All the details

We will not go into details about the pagers in this tutorial. See the online help (keyword: "Pager") for more details.

### Type of control: Thumbnail



*(static page, semi-dynamic page and dynamic page)*

### Summary

The thumbnails are used to automatically manage the display of images in thumbnail mode. During a click on the thumbnail, the initial image will be automatically displayed in a popup page: no programming is required.

### All the details

- ▶ Open the "PAGE\_Thumbnail.WWH" page and run its test.  
We won't go into details about this type of control in this tutorial. See the online help (keyword: "Thumbnail control") for more details.

## LESSON 3.3. ERGONOMICS OF THE SITE

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

---

- Improving the ergonomics of controls and pages
- Improving the ergonomics of the sites



Estimated time: 20 min

## Overview

---

WebDev proposes several tools allowing you to improve the ergonomics of your pages and controls. We will present some simple solutions to improve the usability of your pages.

These operations can be performed on the "Pages and controls" project.

► To open the project:

1. Close the current project if necessary. The home window is displayed.
2. In the home window, click "Tutorial" and select the "Pages and controls" project.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Pages and controls".

## Handling the styles of controls

---

A style is a set of parameters (colors, fonts, ...) affecting the appearance of a control.

Several styles are available for each type of control. All these styles define the style sheet. In HTML, these style sheets correspond to a file with a CSS extension (Cascading Style Sheet).

Each WebDev project is associated with a style sheet (CSS file).



Notes

If a page template is used for your project, the styles of the page template are included in the style sheet of the project.

**For each type of control, you can:**

- use a WebDev style directly. The WebDev styles are proposed by default. These styles define the overall aspect of the control.
- use CSS styles. A CSS style defines the aspect of a control element (for example, the aspect of the caption for an edit control).

If several controls of a project use the same style, the style modification performed from a control will be applied to all the other project controls that use this style.



Caution!

You have the ability to overload a style for a control. In this case, when updating the style, the control will keep its overloaded properties.

You can dissociate a control from its style, only the aspect of the current control will be modified.

We advise you to reduce the number of "dissociated" styles because this option significantly increases the size of the HTML pages (and therefore the time for loading the page).

You want to re-use the styles defined for a site in another site? Nothing's easier, all you have to do is copy the style sheet from a project to another one.

The style sheet to use is specified in the project description ("Skin/Styles" tab). Reminder: To display the project description, on the "Project" pane, in the "Project" group, click "Description".



Notes

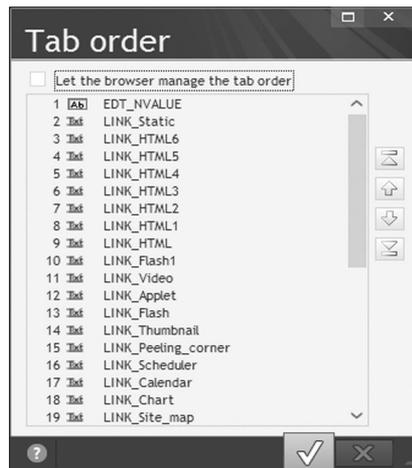
WebDev gives you the ability to use free CSS style sheets. These style sheets will be used by the generated HTML pages.

## Defining the tab order of controls

The tab order of controls is the order in which the controls will be entered when pressing the [TAB] key at run time on the browser of the Web user.

The default tab order is the order in which the controls were created. It can be modified:

- by specifying an automatic tab order, the first control in edit will be the control found in the top left part of the window, the second one will be the one immediately to its right or below, ...
  - by specifying a tab order by selection.
- ▶ To define an automatic tab order, on the "Page" pane, in the "Edit" group, expand "Navigation" and select "Define in automatic".
- ▶ To edit the tab order of controls, on the "Page" pane, in the "Edit" group, expand "Navigation" and select "Edit":



You also have the ability to let the browser manage the tab order.

- ▶ To define a tab order by selection:
1. Open the "PAGE\_Edit.WWH" page available in the "Page and controls" project.
  2. Select the controls in the requested tab order while keeping the [CTRL] key down.
  3. On the "Page" pane, in the "Edit" group, expand "Navigation" and select "Define from selection".

You can view the tab order of controls by pressing the [F5] key.

## Displaying a help in the pages

---

Several methods can be used to propose a help for the controls: help messages, tooltips and indication text. They are intended to guide and inform the Web user about a specific action.

Now let's take a look at the different solutions.

### The tooltip

The tooltip is displayed when the control is hovered by the mouse cursor. It remains displayed as long as the control is hovered by the mouse cursor.

A tooltip is convenient when the associated text does not exceed a few words. The tooltips are very suitable for the image buttons and for the images as these controls have no caption in most cases.

If the tooltip is available for the requested type of control, the text of the tooltip can be entered in the "Note, Help" tab or in the "Help" tab of the description window of the control.

### Help message

The help message is displayed in the status bar of the browser window.

The message (if it exists) is displayed when a control is hovered by the mouse cursor. It remains displayed as long as the control is hovered by the mouse cursor. A help message is convenient when the associated text does not exceed one line. This message is used to give additional information about the data currently entered.

**Tip**

A help message can be displayed at any time by **Message**:

```
Message("The name must not exceed 25 characters")
```

If the help message is available for the requested type of control, the text of the message can be entered in the "Note, Help" tab or in the "Help" tab of the description window of the control.

### Indication text

Another type of help is available for the edit controls: the indication text. This text is displayed in gray in the empty control. This help text is automatically erased as soon as the Web user types the first character in the control.

This indication text is entered in the "Note, Help" tab of the edit control.

**Tip**

The indication text can also be defined by programming with **..Hint** :

```
EDT_Customer..Hint = "Enter the name of the customer"
```

## Create your pages by using the zoning mode

### Overview

The zoning mode is used to create and edit the pages of your site in zones. The zones are used to define the architecture of the page: header, page footer, inset, ...

The zone contains the controls that are included in it. If a zone is moved, the controls found in the zone are moved accordingly.

The zones are materialized and they can be easily handled in the editor.

Example: Increasing the size of a header automatically moves the areas below.





The zoning mode is used to easily design the pages and it brings several features (among which the anchoring of controls).

### How to use the zoning?

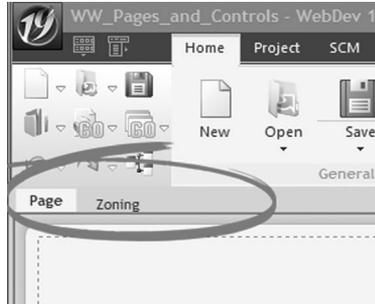
The "WW\_Pages\_and\_controls" example contains several pages allowing to understand the different modes for splitting/dividing the page in zoning mode. These pages are grouped in the "Zoning" folder in the project explorer.

#### ► To open the project:

1. Close the current project if necessary. The home window is displayed.
2. In the home window, click "Tutorial" and select the "Pages and controls" project.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Pages and controls".

- ▶ Let's take a look at the first page for zoning management:
  1. Open the "PAGE\_zoning01" page (available in the "Zoning" folder in the project explorer).
  2. In the project editor, two tabs are available above the page: "Page" and "Zoning".



- The "Page" tab is used to create the page controls.
  - The "Zoning" tab is used to define and handle the page zones.
3. Click the "Zoning" tab. Two tools can be used to split the page:
    - Splitting pages (on the "Page" pane, in the "Edit" group, expand "Zoning" and select "Split the layout"). This tool is used to split an existing layout zone into two separate zones.
    - Subdividing pages (on the "Page" pane, in the "Edit" group, expand "Zoning" and select "Sub-split the layout"). This tool is also used to create two new zones while keeping the initial zone; therefore, three zones will be found after this operation: the initial zone and 2 new zones inside this initial zone.
  4. Two zones have been created in this page. Run the test of the page.

### Practical example

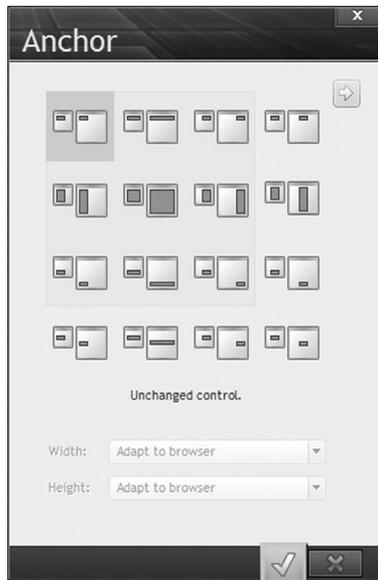
- ▶ Open the "PAGE\_Zoning" pages (numbered from 01 to 05) and run their tests: These pages present the use of the "Split the layout" option. These pages also use the anchoring options. During the test, change the size of the browser to see its impact on the different zones.
- ▶ Open the "PAGE\_Zoning06" page and run its test: This page presents the use of the "Sub-split the layout" option. The different zones of this page are anchored. During the test, change the size of the browser to see its impact on the different zones.

## Anchor the controls

WebDev proposes an automatic mode for anchoring the controls.

To anchor a control, you must:

1. Select the control that must be anchored in the page.
2. Display the popup menu of the control and select "Anchor".
3. The window used to define the anchoring is displayed:



The controls can be anchored in position and in size (height and width). The anchoring is used to define whether the control must be enlarged according to its content.



Notes

**Caution:** To use the anchors, the page must use the "Zoning" edit mode. This edit mode is defined in the "General" tab of the description window of the page.

## Use the rulers and the alignment options

One of the most important points when creating a site is the aspect of the interface. An important point is how the different controls are standardized, how they are aligned in the page.

Several WebDev tools help you create interfaces that comply with the programming standard: grid, rulers, alignment options, automatic positioning, ... A few tips to choose the best suitable tool.

When creating our pages in the previous lessons, you have probably noticed that dotted lines appeared when moving your controls in the page. This is the **automatic positioning**. These "rulers" help you position your controls in real time. You can see straightaway whether the control is aligned with the previous control.

To configure the automatic positioning:

1. On the "Display" pane, click the group button  of the "Options" group.
2. In the window that is displayed, select the "Magnetism" tab. You have the ability to enable or disable the automatic positioning.



Notes

To move your controls without using the magnetism or the rulers, keep the [SHIFT] key down during the move.

You may also want to align some page controls once they have been created (after their move for example). You have the ability to use the alignment options. These options are grouped in the "Alignment" pane.

To use the features of this toolbar, all you have to do is select several controls (with the lasso or with the CTRL key) and choose one of the preset alignments. The control taken as reference to perform the alignment is the first selected control.



Notes

WebDev allows you to view the effect of the alignment option without applying it. To do so, select the controls to align and hover the requested alignment icon with the mouse cursor. To apply the option, all you have to do is click the alignment icon.

If none of the preset alignments suits you, you have the ability to define a custom alignment: on the "Alignment" pane, in the "Other alignments" group, the "Custom alignment" option is used to configure all the options.

The rulers allow you to position your controls to the nearest millimeter. To display the rulers, press [CTRL] + [R].

## Use the tabs to group your controls

---

Your page contains several controls? Why not use tabs to group the information.

The tabs are used to group the information by theme. The Web user can access a theme by clicking the requested "pane" directly. The principle of tabs is now used in several e-business sites.

The principle for handling tabs is very easy. You must:

1. Create the tab control: on the "Creation" pane, in the "Containers" group, click "Tab".
  2. Create as many panes as necessary and give a caption to each pane ("General" tab in the control description).
  3. Associate the controls with the relevant tabs. A control can be:
    - common to all the tabs (in this case, it is associated with no specific tab).
    - associated with a specific tab.
- Two methods can be used to associate the control with a tab:
1. By moving the control:
    - Click the requested tab to enable one of its panes.
    - Click the control to associate and move it to the active pane.  
The control is automatically associated with the tab and with the tab pane.
  2. By using the popup menu:
    - Move (if necessary) the relevant control onto the tab.
    - Right-click the control to associate and select "Associate with a tab..." from the popup menu.
    - Select the tab and the tab pane, and validate.  
The control is automatically associated with the tab and with the tab pane.

A control can be associated with a single tab pane. The control will be active and visible when the tab pane is enabled.

Your tab is ready to operate, no programming is required.

## Improve the navigation in your site

---

WebDev proposes several tools allowing you to improve the navigation in a site:

- the tabs (as already seen)
- the drop-down menus (see the online help for more details).
- the Site Map Path controls (as already seen in the previous lesson).
- the Site Map controls (as already seen in the previous lesson).

All these elements help you create a user-friendly interface and allow the Web user to easily find the requested information.

## Use modern dialog boxes

With the new Web technologies, the sites can look more like Windows applications and they can display dialog boxes.

In order for the Web user to easily identify the active page, the system for graying the pages is automatically used. The inactive page is grayed and the active pages can be easily identified by the Web user.

Several methods can be used to implement the dialog in server code and in browser code. You now have the ability to use Popup pages.



The popup pages can have a rectangular shape or any other shape. To display them, all you have to do is use **PopupDisplay** and **PopupClose**.

See the online help (keyword: "Dialog, With the Web user") for more details.

### Practical example

The "WW\_Pages\_and\_controls" example contains an example for using the popup pages.

- ▶ To open the project:
  1. Close the current project if necessary. The home window is displayed.
  2. In the home window, click "Tutorial" and select the "Pages and controls" project.  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Pages and controls".
- ▶ Open the "PAGE\_POPUP" page and run its test. This page displays the personal details of a contributor and it is used to select the branch in a Popup page.

## Interface of the site: Use the page templates

The first purpose of a site is to provide a service (e-commerce, search, ...) but one of the most important aspect is the interface: if the interface is user friendly, the Web user will be more likely to visit again.

To improve the interface of your site and to standardize its appearance, WebDev enables you to use page templates. In this tutorial, all the sites that were created or handled included page templates.

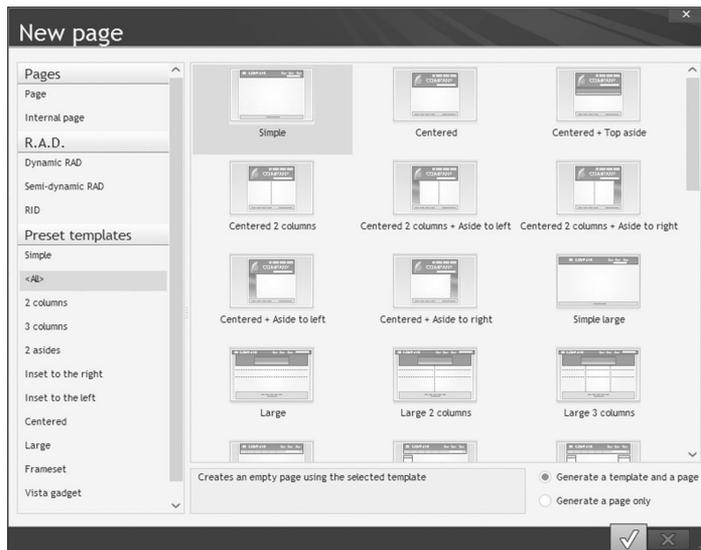
### What is a page template?

A page template is a template. This template can be used for any new page that is created.

In most cases, a page template contains the "default" presentation for the site pages. For example: a logo, an identification area, a link area, menus in tab format or drop-down menus, ... A page template allows you to implement a style book.

A page template can contain images, controls, code lines, procedures ... A page can be linked to several templates. All the pages that use a page template inherit from all the characteristics of the template.

When creating a blank page, you have the ability to select the template associated with this page ("All" enables you to see all the available preset templates).



If the template is modified, all the pages that refer to this template are also modified. The page templates have been used in part 2 of this tutorial (see "Customizing the generated site", page 117).

## How do I create a page template?

► To create a template:

1. Click  among the quick access buttons. In the wheel that is displayed, hover "Page" and click "Page template". Select the source of the template and validate.
2. The model is currently created. Create the controls, images, procedures ... required by the template.
3. Save the template and specify its name.



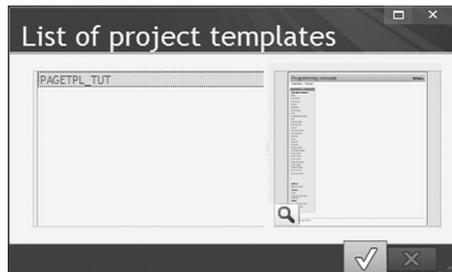
Tip

Don't forget to give a specific name (with a prefix) to all the controls found in your template in order to easily identify them.  
For example: "TPL\_EDT\_PSEUDONYM" for an edit control of the template.

Note: Several page templates can be applied to the same page.

## How do I apply a page template?

- If one or more templates have already been created in your project, they can be applied to your existing pages or to your new pages:
1. Open an existing page found in your project or create a new page.
  2. On the "Page" pane, expand "Templates" and select "Inherit from a template". The following screen is displayed:



3. Select a template to import and validate.

The controls, the code of controls and the template procedures are imported (a small yellow square is displayed on the left side of each template control).

4. The controls, codes and procedures of the template cannot be modified from the page to which the template was applied.

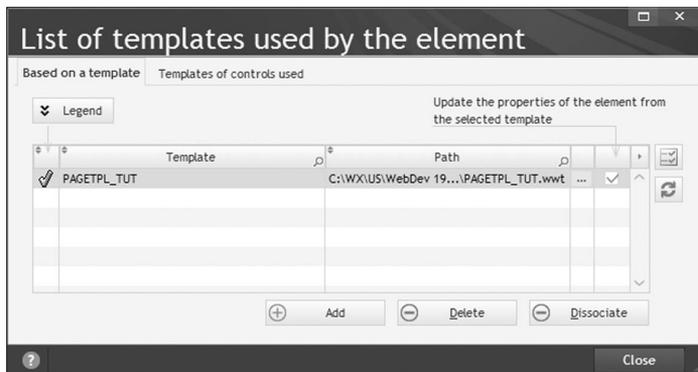
To do so, the template must be opened: click one of the template controls (identified by a yellow square) and select "Open the template" from the popup menu.

## Updating the template

To update a template, on the "Page" pane, expand "Templates" and select "Update the associated pages".

## How do I dissociate a template imported into a page?

- ▶ To dissociate a template imported into a page, which means to separate the template from the page (the page will no longer be updated when the template is modified):
  1. On the "Page" pane, expand "Templates" and select "List of templates used". A selection window is displayed:



2. Select the template that must be dissociated and click "Dissociate".
3. Validate. The template controls are freed in the page and they belong to the page.



Tip

### Our advice: Page templates or Framesets?

We recommend that you use page templates rather than framesets in your Web-Dev sites!

Easier to use and to maintain, the templates are faster to load at run time in the browser! Indeed, a single page is generated for a page with an associated template while a frameset generates as many pages as the number of frames!

## LESSON 3.4. REUSABILITY

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

---

- Re-using the project elements



Estimated time: 10 min

## What is the reusability?

You want to re-use the same directory picker in all your projects? You want to use the same "Close" buttons at the same location in your sites?

Several methods can be used to manage the reusability of the created features. The page templates, allowing you to easily define the style of a site, were presented in the previous lesson.

This lesson will explain how to re-use the features.

## Re-using a set of controls

You want to re-use a set of controls? To do so, the following tools are available in WebDev:

- Supercontrols
- Control template
- Internal pages

We are going to present these three tools in details, with a summary table for each tool to help you choose the most suitable one.

For each tool, the "Pages and controls" project contains a page allowing you to view and check the feature.

### Practical example

To open the project:

1. Close the current project if necessary. The home window is displayed.
  2. In the home window, click "Tutorial" and select the "Pages and controls" project.
- Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Pages and controls".

### The supercontrol

The supercontrol is an advanced type of control. A supercontrol groups a set of controls for a specific purpose (directory picker, file picker, ...). The supercontrol contains:

- the necessary controls
- the necessary code.

Summary table

|          |  |
|----------|--|
| Creation | <p><b>Creating a supercontrol:</b></p> <ol style="list-style-type: none"> <li>1. Create the supercontrol (on the "Creation" pane, in the "Containers" group, click "Supercontrol").</li> <li>2. Include the controls in the supercontrol.</li> </ol> <p><b>Refactoring:</b></p> <ol style="list-style-type: none"> <li>1. Select the controls.</li> <li>2. On the "Modification" pane, in the "Transformations" group, expand "Refactoring and swapping" and select "Create a supercontrol from the selection".</li> </ol> |
|----------|--|

|             |  |
|-------------|--|
| Reusability | Via the dictionary   |
| Upgrade     | Via the dictionary (via the subscription mechanism)                      |
| Teamwork    | Sharing the dictionary by network.<br>Source Code Manager not available. |

## Internal page

The Internal Page control is used to include a page (and its code) in another page.

At run time, the page to merge will be dynamically merged to the host page.

An internal page is a specific page. All types of controls can be found in this page. An internal page is a file whose extension is "WWH".

The "Internal page" control is used to dynamically share the same interface section within one or more sites.

### Summary table

|             |   |
|-------------|---|
| Creation    | <p><b>Creating an internal page:</b></p> <ol style="list-style-type: none"> <li>1. Creating the internal page. To create an internal page, you must: <ul style="list-style-type: none"> <li>• Click  among the quick access buttons.</li> <li>• In the wheel that is displayed, hover "Page".</li> <li>• Click "Internal page".</li> </ul> </li> <li>2. Define the controls and the code.</li> <li>3. Save.</li> </ol> <p><b>Refactoring:</b></p> <ol style="list-style-type: none"> <li>1. Select the controls.</li> <li>2. On the "Modification" pane, in the "Transformations" group, expand "Refactoring and swapping" and select "Create an internal page from the selection".</li> </ol> |
| Reusability | Via the "Internal page" control.  |
| Upgrade     | Via the dictionary (via the subscription mechanism) or via the SCM.   |
| Teamwork    | Share via the network or via the SCM  |
| Benefits    | Ability to dynamically modify (by programming) the internal page used in the Internal Page control.   |
| Drawbacks   | <ul style="list-style-type: none"> <li>- No improvement is allowed in the interface: no overload, no move of controls.</li> <li>- Rectangular area.</li> </ul>  |

**Example:** Page whose interface will never change regardless of the project: option page.

### Practical example

- ▶ Open the "PAGE\_InternalPage.WWH" page and run its test. This page uses two Internal Page controls allowing you to enter the identity and the address of a person.

## Control template

A control template is a set of controls that can be re-used in several pages. 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 "WWT".

### Summary table

|             |   |
|-------------|---|
| Creation    | <p><b>Creating a control template:</b></p> <ol style="list-style-type: none"> <li>1. Creating the control template. To create a control template, you must: <ul style="list-style-type: none"> <li>• Click  among the quick access buttons.</li> <li>• In the wheel that is displayed, hover "Page".</li> <li>• Click "Control template".</li> </ul> </li> <li>2. Define the controls and the code.</li> <li>3. Save.</li> </ol> <p><b>Refactoring:</b></p> <ol style="list-style-type: none"> <li>1. Select the controls.</li> <li>2. On the "Modification" pane, in the "Transformations" group, expand "Refactoring and swapping" and select "Create a control template from the selection".</li> </ol> |
| Reusability | Via the "Control template" control.   |
| Upgrade     | Via the dictionary (via the subscription mechanism) or via the SCM.   |
| Teamwork    | Share via the network or via the SCM  |
| Benefits    | The control templates can be overloaded: source code can be added, the controls can be moved in the page that uses the control template. The controls can be modified.  |

**In most cases, we recommend that you use a control template.**

### Practical example

- ▶ Open the "PAGE\_ControlTemplate" page and run its test. This page uses two controls linked to a control template to enter a time period.

## LESSON 3.5. QUESTIONS/ANSWERS

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

---

- Tips for handling the controls



Estimated time: 10 min

**Question** How do I display a page in a new window of the browser?

Two methods can be used to display a page in a new browser window:

- by programming, with **BrowserOpen**, **PageDisplay** or **DynamicSiteDisplay**
- via the page editor by using the options of a button or link.

**By programming**

► To open a new browser window, use the following code:

**1. BrowserOpen:**

```

WL      Click (onclick) of Button1 (browser)

BrowserOpen("", "Browser_New", ONResizable+ONStatus, 640, 480, 100, 100)
ChangeTarget("Browser_New")

Click of Button1 (server)

PageDisplay(Page1)

```

**2. PageDisplay:**

```

WL      Click (onclick) of Button1 (browser)

PageDisplay("StaticPage", NewBrowser, "Browser_New", ONResizable+ONStatus, 640, 480, 100, 100)

```

**3. SemiDynamicPageDisplay:**

```

WL      Click (onclick) of Button1 (browser)

SemiDynamicPageDisplay("SemiDynPage", "", NewBrowser, "Browser_New", ONResizable+ONStatus, 640, 480, 100, 100)

```

**4. DynamicSiteDisplay:**

```

WL      Click (onclick) of Button1 (browser)

DynamicSiteDisplay("WWWSite", "New", "", NewBrowser, "Browser_New", ...
ONResizable+ONScrollBar, 640, 480, 100, 100)

```

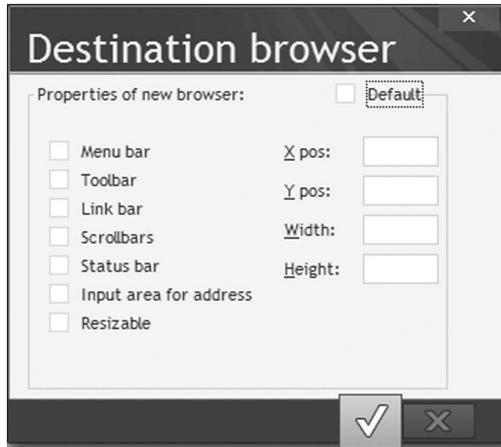
See the online help (keywords: "BrowserOpen", "PageDisplay", "SemiDynamicPageDisplay" and "DynamicSiteDisplay") for more details.

**By using the page editor**

► To specify that the target will be a new browser when a link or a button is clicked:

1. Display the description window of the button or link.
2. In the "General" tab, choose "New browser (\_blank)" in the "Destination" list.

3. Click . A window for configuring the browser is displayed:



4. Select the requested options and validate.

### Question How do I modify the color of a static by programming?

The color of the statics is defined in the style of the control (in the description window of the control). However, the color of a static can be modified by programming. The syntax is as follows:

```
-- Click of BTN_Button1 (server)
// Color the static in red
STC_Static1..Color = PastelRed

// Color the background of the static in dark gray
STC_Static2..BrushColor = DarkGray

// Restore the initial color of the static
STC_Static1..Color = iDefaultColor
```

**RGB** is used to define a color from the values of the Red, Green and Blue components.

```
-- Click of BTN_Button1 (server)
nColor = RGB(<red>, <green>, <blue>)
```

You also have the ability to modify the background color of an edit control in browser code. To do so, assign the HTML color code to **..BrushColor**:

```
-- Click of BTN_Button2 (browser)
CTL_ERROR..BrushColor = "FF0000"
```

**Question** How do I make a button invisible?

A button can be made invisible by programming with the following server syntax:

```
-- Click of BTN_Button1 (server)
BTN_Button2.Visible = False
```

Set the value to "True" to make the button visible.

This syntax can also be applied to all types of controls and to the groups of controls.

**Question** How do I create a vertical menu in a WebDev page?

The menu can be displayed vertically in the menu editor.

- ▶ To create a menu:
  1. Create a new page if necessary.
  2. On the "Creation" pane, in the "Navigation" group, click "Menu". A wizard for menu creation is displayed, allowing to choose the style of the menu to create. Select "MENU\_Btn\_V" for example. Validate the creation of the menu.
  3. A menu containing two options is inserted into your page.
  4. Click the menu to switch to "menu edit" mode (a yellow border is displayed around the menu).
  5. Right-click the requested menu option.
    - Click "Option description" to modify the selected option.
    - Click "Menu description" to modify the style of the menu.
    - Click "Add after" to add a menu option.
    - Click "Insert an option" to insert a menu option at the selected location.
    - Click "Insert a sub-menu" to add a sub-menu into the tree structure of the selected menu option.
  6. Right-click the menu again and select "Menu description".
  7. Choose the requested orientation for your menu: horizontal (by default) or vertical.
  8. Press the [ESC] key to exit from the "edit" mode of the menu.

**Question** How do I add a popup menu to a table or to a looper?

A popup menu can be added to a table control or to a looper control; to do so, select "Popup menu" from the description window:

- of the table, "Details" tab.
- of the looper, "General" tab.

The popup menu can be accessed in the page via . A click performed on this image is used to display the popup menu proposing to export the data from the table or looper to an Excel, Word or XML file (created on the server and downloadable).

**Question** How do I pass parameters to a page?

The method for passing parameters to a page is similar to the method for passing parameters to a procedure.

In the declaration code of the global variables of the page, enter the following WLanguage code:

```
-- Declaring the global variables of the page  
PROCEDURE PageName (NameParam1, NameParam2, ...)
```

When the page is opened by **PageDisplay**, the parameters must be passed after the name of the page:

```
PageDisplay (PageName, ValueParam1, ValueParam2, ...)
```

If a parameter is initialized when declaring the procedure, this parameter becomes optional:

```
-- Declaring the global variables of the page  
PROCEDURE PageName (NameParam1, NameParam2 = "DefaultValue")
```



Notes

We advise you to pass parameters to a page rather than to declare global variables in the project.

To find out the value of a parameter passed to the current page, use **PageParameter**.

**Question****How do I group the controls in order to modify their properties by programming?**

Perform the following operation:

1. Select several controls with the mouse.
2. Display the popup menu of the selection and select "Groups .. Associate the selection...".
3. Click "New" and enter the name of the group that was created, then validate.

The controls are associated with this group. You can:

- Modify the properties of the controls found in the group with the following syntax:

```
GroupName..<PropertyName> = Value
```

- Access the value of the properties for the controls found in the group with the following syntax:

```
Value = GroupName..<PropertyName>
```

**Caution!**

Only the properties common to all the controls belonging to the group can be modified.

**Question****How do I transform a check box into a radio button?**

On the "Modification" pane, in the "Transformations" group, expand "Refactoring and swapping" and select "Radio button/Check box".

**Question****How do I add an image to the page background?**

On the page:

1. Right-click and select "Description".
2. Select the "Style" tab.
3. Select the background image of the page ("Background image").

The "Fixed background image" option allows you not to scroll the image if the page cannot be entirely displayed in the browser.

**Note:** A larger image will increase the time required to display the page.

**Question** How do I display the date of site update?

To automatically display the date of site update:

1. Create a formatted display control: on the "Creation" pane, in the "Usual controls" group, expand "Text" and select "Formatted display control".
2. Enter the following code in the initialization process of the control:

```
MySelf = fDate(CompleteDir(fExeDir()+ ...  
                "<Library Name>.WDL", "", fModify)
```



**PART 4**

**Databases  
and analyses**

**10**

**DEVELOP 10 TIMES FASTER**





# LESSON 4.1. INTRODUCTION

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

---

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



Estimated time: 10 min

## Overview

---

You may have to handle data when designing a static site or a dynamic site. To store the data, you must create a "database".

In WebDev, when creating a project that handles data, you must create an analysis.

An "analysis" contains the description of the files (or tables) containing the site data.

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



Notes

Several tools allowing you to perform maintenance operations on the HFSQL databases are supplied with WebDev. They can be accessed from the HFSQL Control Center.

Several database formats are supported by WebDev. The most common ones are:

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

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

- Native Access
- OLE DB access
- Direct ODBC access
- ODBC access via OLE DB

## The different modes for accessing the databases

---

### Native Access

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

In WebDev, a native access (compatible with WinDev) 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)

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

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

### Direct ODBC access

An access via direct ODBC uses a multi-database access standard. The 32-bit ODBC layer must be installed on the Web server. In most cases, this layer is already installed in the recent versions of Windows. This can be checked in the Control Panel of Windows on your development computer by selecting "ODBC administrator" (or ask the site manager for the deployment computer).

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

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

### OLE DB access

An access via OLE DB 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 development computers and on the deployment computers (version 2.6 or later).

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

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

**ODBC access via OLE DB**

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

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

## LESSON 4.2. BROWSING THE DATA FILES AND THE QUERIES

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

---

- HReadFirst browse
- HReadSeek browse
- FOR EACH browse
- FOR EACH WITH browse



Estimated time: 20 min

## HReadFirst browse

---

The **HReadFirst** browse is used to browse a data according to a search key item (or index). In this case, **HReadFirst** is combined with **HReadNext** that is used to read the next record.

For example, the following code is used to read the Customer file according to the "Name" key item, from the first record to the last one.

```
HReadFirst (Customer, Name)
WHILE NOT HOut (Customer)
  // Process the record
  HReadNext (Customer, Name)
END
```

In this code, the read loop is used to check the value of **HOut**. **HOut** is used to find out whether the end of the data file has been reached.

When browsing a query, **HReadFirst** automatically re-runs the query.



Note

The same method can be used to browse the data file from the last record to the first one. To do so, use **HReadLast** and **HReadPrevious**.

## HReadSeek browse

---

The **HReadSeek** browse is used to position in the data file in order to browse the records corresponding to a condition. This condition is applied to the search key (all the orders passed by a customer for example). **HReadNext** is used to read the next record corresponding to the condition.

For example, the following code is used to find the customers whose name starts with "Smith":

```
HReadSeek (Customer, Name, "Smith")
WHILE HFound (Customer)
  // Process the record
  HReadNext (Customer, Name)
END
```

In this code, the browse loop is used to check the value of **HFound**. **HFound** is used to find out whether a record corresponding to the condition was found.



Note

**HReadSeek** performs a generic search: all the records that start with the specified value will be sought (in our example, Smith and Smither will be found).

To perform an exact-match search:

- use **HReadSeekFirst**.
- use the *hIdentical* constant associated with **HReadSeek**.

## FOR EACH browse

---

The FOR EACH browse is used to perform a full browse on a HFSQL data file, query or view.

The browse can be performed:

- on the best key of the data file, automatically detected by the HFSQL engine.
- on a specific key.

For example, the following code is used to browse the Customer file according to the "Name" item, from the first record to the last one.

```
FOR EACH Customer ON Name
    // Process the records
END
```

This code is equivalent to the code presented for the *HReadFirst* browse:

```
HReadFirst(Customer, Name)
WHILE NOT HOut(Customer)
    // Process the records
    HReadNext(Customer, Name)
END
```

The browse can be performed from the last record to the first one by specifying the direction:

```
FOR EACH Customer ON Name FromEnd
    // Process the records
END
```

## FOR EACH WITH browse

---

The FOR EACH WITH browse is used to browse a HFSQL data file according to a condition.

The browse can be performed:

- on the best key of the data file, automatically detected by the HFSQL engine.
- on a specific key.

For example, the following code is used to browse the Customer file according to the customer number and by specifying a condition on the customer name.

```
// Browse with filter
FOR EACH Customer ON CustomerNum WHERE "Name = 'Smith'"
    // Add customers into the list
    ListAdd(CustomerList, Customer.CustomerNum)
END
```

This code is equivalent to the code presented for the *HReadSeek* browse:

```
HReadSeek(Customer, Name, "Smith")
WHILE HFound(Customer)
    // Process the record
    HReadNext(Customer, Name)
END
```

## What type of browse should I choose?

---

All the types of browsing operations are equivalent. The type of browse depends on the keys found in the data file, on the size of the file, on the type of search performed ("Contains" search or not).

## LESSON 4.3. MANAGING THE ERRORS

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

---

- The automatic management of errors
- The programmed management of errors



Estimated time: 20 min

## Overview

---

When managing a database, several types of errors may occur when adding or modifying a record:

- Password error on a data file
- Error of mandatory input
- Duplicate error
- Integrity error, ...

WebDev proposes several modes for managing these errors:

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

### Practical example

Let's take a look at these modes for error management via a simple example. This example includes two data files (Product and VAT) allowing us to check the duplicate errors and the integrity errors.

- ▶ Open the "WW\_Managing\_HFSQL\_Errors" project:
  1. Close (if necessary) the current project to display the home window.
  2. In the home window, click "Tutorial" and select the "Managing HFSQL errors" project. The project is loaded.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Managing the HFSQL errors".

## Managing the errors in automatic mode

---

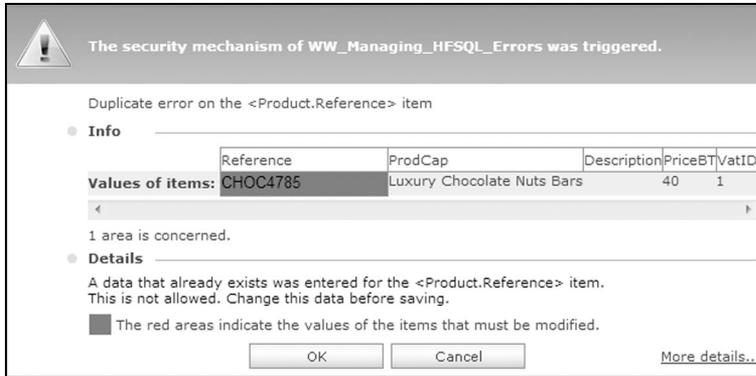
The automatic mode is enabled by default. In this mode, WebDev proposes standard pages for each type of error. These pages indicate the problem to the Web user.

No specific code is required in the site.

Let's now check the type of error that can be displayed in our simple example.

- ▶ Run the "WW\_Managing\_HFSQL\_Errors" project by clicking the "GO" button (  among the quick access buttons).
- ▶ Click the "Automatic" link.
- ▶ We are now going to check the duplicate errors. To do so, click the "Duplicates: Fill the controls" button. A new product is automatically entered in the form at the bottom of the page.

- ▶ Click the "OK" button to add data into the database. The following screen is displayed:



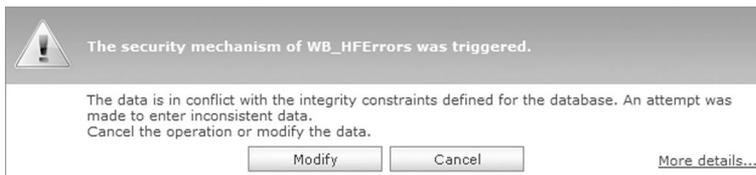
This dialog box explains the problem: a value that already exists was entered for the "Reference" item. A duplicate error occurs. This dialog box allows the Web user to correct the value directly.

The "OK" button is used to validate the input in the error window.

The "Cancel" button is used to go back to the screen that triggered the error.

Click the "Cancel" button.

- ▶ Let's now check the integrity errors. To do so, click the "Integrity: Fill the controls" button.
- ▶ Click the "OK" button to add data into the database. The following screen is displayed:



This dialog box indicates that an integrity error occurred (in our case, no VAT rate was entered).

The "Modify" button is used to go back to the screen currently in edit.

In the automatic mode, the help dialog boxes are displayed as soon as an error regarding the management of the database occurs. In some cases, it may be interesting to manage these errors manually. In this case, the developer can take a more precise action on the errors that occurred.

- ▶ Click the "Cancel" button. Close the browser.

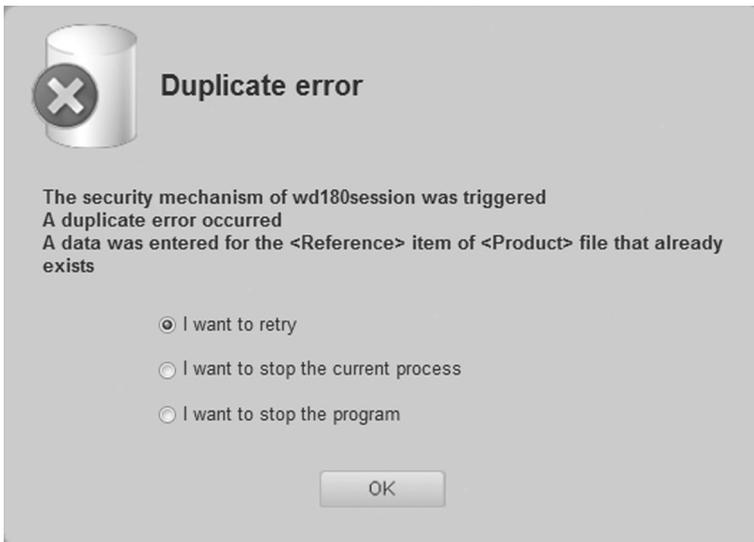
## Managing the errors in advanced programmed mode

The management of errors in advanced programmed mode consists in customizing the process of the error by programming. WebDev gives you the ability to use:

- a custom page for managing the errors.
- a specific procedure.

In our example, we will be using a specific page.

- ▶ Run the "WW\_Managing\_HFSQL\_Errors" project by clicking the "GO" button (  among the quick access buttons).
- ▶ Click the "Programmed" link.
- ▶ We are now going to check the duplicate errors. To do so, click the "Duplicates: Fill the controls" button. A new product is automatically entered in the form at the bottom of the page.
- ▶ Click the "OK" button to add data into the database. The following screen is displayed:



This dialog box displays the reason for the problem using a custom interface. Select "I want to stop the current process" and validate.

- ▶ Let's now check the integrity errors. To do so, click the "Integrity: Fill the controls" button. A new product is automatically entered in the form at the bottom of the window.

- Click the "OK" button to add data into the database. The following screen is displayed:



Select "I want to stop the current process" and validate.

- Let's take a look at the code that was used to display these messages:
1. In the editor, open the "PAGE\_Program\_Errors" page (double-click the name of the page in the "Project explorer" pane).
  2. Display the processes of the page ("Code" from the popup menu of the page). The initialization code of the page contains the following code lines:

```
HOnError ("*", hErrDuplicates, PAGE_Duplicate2)  
HOnError ("*", hErrIntegrity, PAGE_Integrity2)
```

Once again, **HOnError** is used to configure the mode for managing the database errors. The parameters of the function are used to specify the range of the modification:

- "\*" to specify that all the data files found in the analysis are affected.
  - the **hErrDuplicates** constant to specify that only the duplicate errors are affected. Then, the name of the page to use must be specified (PAGE\_Duplicate2).
  - the **hErrIntegrity** constant to specify that only the integrity errors are affected. Then, the name of the page to use must be specified (PAGE\_Integrity2).
3. Close this code window.

We won't go into details about all the code of the pages used. See the online help for more details.

# LESSON 4.4. DATA ENCRYPTION

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

---

- What is data encryption used for?
- Using the data encryption.



Estimated time: 15 min

## What is data encryption used for?

---

One of the WebDev features for the HFSQL data files is to guarantee the confidentiality of the data found in the HFSQL data files.

This confidentiality is guaranteed by the encryption of the data files.

The encryption of data files is used to make the content unreadable to any person who does not own the decryption key. Indeed, a data file is encrypted according to a key (also called "password"). This password is (and must be) known by nobody but you.

When a data file is described in the data model editor, you have the ability to define whether an encryption must be performed:

- on the data file itself.
- on the index.
- on the memo files linked to the data file.

Several types of encryption are available:

- 128-bit encryption (high performance).
- RC5 encryption (128 bits, 12 loops, less performance but more secured)
- RC5 encryption (128 bits, 16 loops, less performance but more secured).
- automatic encryption if password. This encryption mode uses a RC5 encryption (128 bits, 16 loops).

The key (or "password") defined by yourself will be valid for the data file only (and not for the entire analysis). You have the ability to define an encryption for all the data files or for some of the data files found in your analysis.

The data found in the file (and/or in the index and/or in the memo files) will be "encoded" according to the key (or "password") defined when generating the data files.

When decoding the data file, the password can be:

- found in the code of the site. In this case, the Web user will not know the password.
- enabled by an advanced operating mode: password read in a file, enabled by a specific process ...

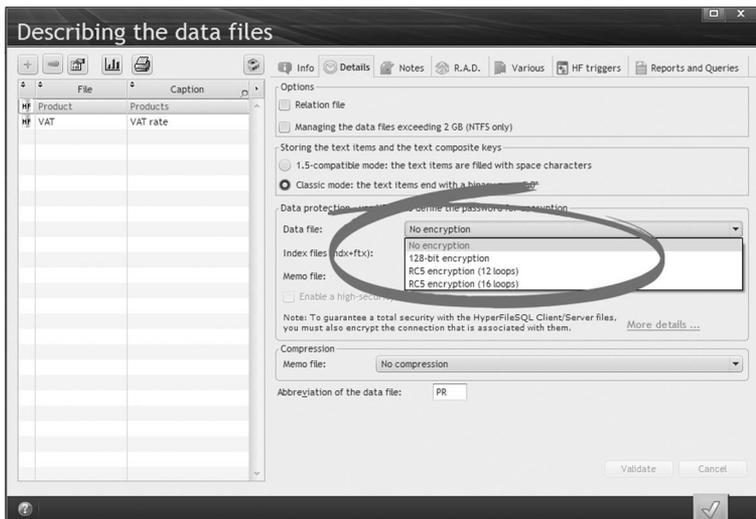
The password must be managed by the developer, in other words ... you! WebDev will encode or decode according to the specified password.

## How do I encrypt the data files?

### Implementing the encryption of the data files

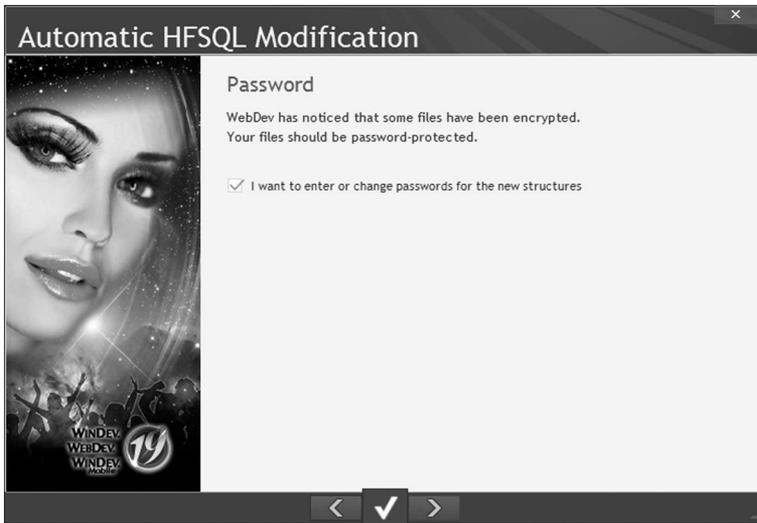
We are going to implement the data encryption on one of the examples supplied in this tutorial, "WW\_Managing\_HFSQL\_Errors".

- ▶ Open (if necessary) the project named "WW\_Managing\_HFSQL\_Errors":
  1. Close (if necessary) the current project to display the home window.
  2. In the home window, click "Tutorial" and select the "Managing HFSQL errors" project. Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Managing the HFSQL errors".
- ▶ To use the encryption on a data file:
  1. Load the analysis of your project in the data model editor: click  among the quick access buttons.
  2. Select the data file that must be encrypted. In our example, we are going to encrypt the "Product" data file.
  3. In the description of the data file ("Description of data file" from the popup menu, "Details" tab), choose the type of encryption for your data file, your memo or your index.

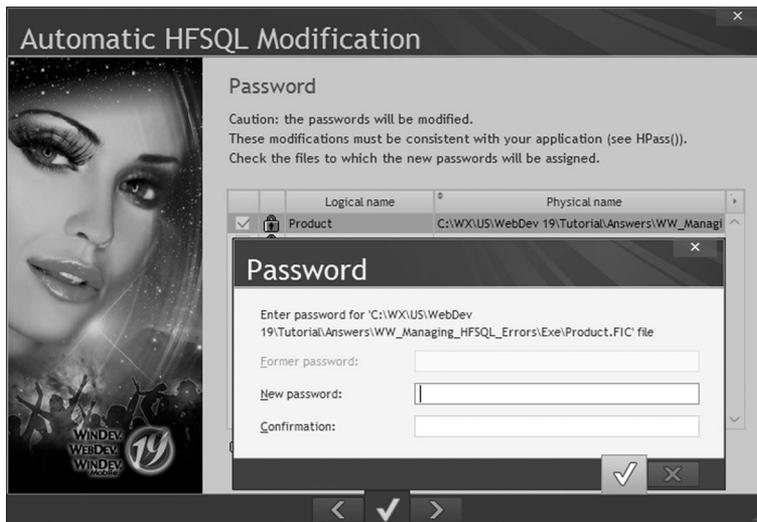


You also have the ability to enable a higher security. In this mode, the encryption password must be entered whenever an automatic data modification is performed (new setup, update, ...).

When generating your analysis, the wizard proposes to enter or to modify (if your data file was already encrypted) the encryption password.



The list of data files for which an encryption was requested is displayed. You have the ability to select the data files to encrypt and to enter the passwords for each one of the data files.



## Managing the encryption in WLanguage

To manage an encryption password in WLanguage, you can:

- specify the password with the functions for opening and creating data files (*HCreation*, *HCreationIfNotFound*, *HOpen*).

Example for using *HCreationIfNotFound*:

```
HCreationIfNotFound(CUSTOMER, "Password")
IF HErrorPassword() THEN
    Trace("Wrong password")
END
```

- use *HPass* before the first read or write operation performed in the encrypted data file.

Example for using the *HPass* function:

```
// Open a file with password and error check
HPass(CUSTOMER, "Password")
HCreationIfNotFound(CUSTOMER)
IF ErrorOccurred THEN
    Trace("HFSQL error: " + HErrorInfo())
    RETURN
END
```

See the online help (keyword: "HPass", "FicCryptMethod", "MmoCryptMethod", "NdxCryptMethod") for more details.

# LESSON 4.5. ADVANCED CONCEPTS

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

---

- The log process
- The transactions
- The replication
- The triggers



Estimated time: 15 min

## Overview

---

This lesson is going to present important concepts for database management. The purpose of this lesson is to present these concepts and their implementation with a HFSQL database.

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

## The log process

---

### What is the log process used for?

The log file is a specific file allowing you to fulfill the following requirements:

- restore the database to its previous status further to a physical problem, process errors or input errors, ...
- keep track of the operations performed on a "sensitive" data file (who modified or read the file for example).
- manage the replication (see "The replication", page 254).

The log file acts as a security mechanism for your database. It allows you to quickly restore your database to its last status before the problem, since the last backup, without having to re-enter everything.

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

## The transactions

---

A transaction is a set of indissociable operations performed on a data file:

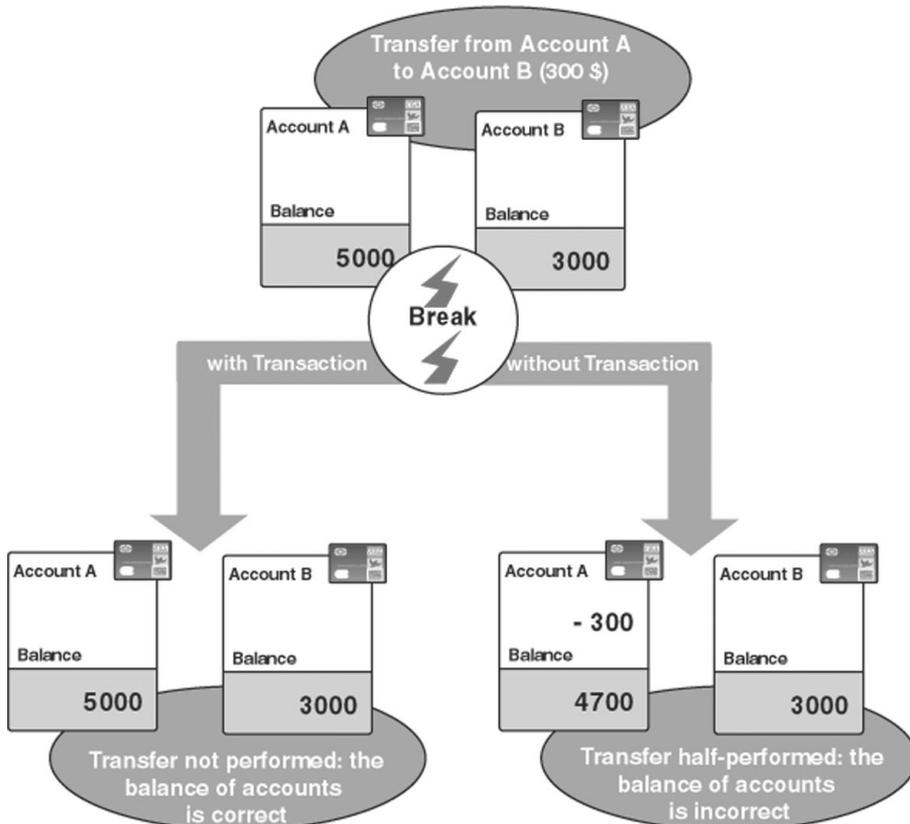
- either all the operations of the transaction are performed.
- or no operation is performed.

The management of transactions is the best way to ensure the integrity of a set of indissociable write operations performed on HFSQL data files.

The transactions are also a reliable way for securing your processes on HFSQL data files.

### A simple example of the need for transactions in some types of processes

During a bank wire transfer, an account is debited while another account is credited. A failure may occur on the network while the operations are performed (power outage, computer crash). This is the reason why this process must be managed by a transaction. If a breakdown occurs, all the operations to perform are canceled. This way, we don't end up with an account being debited without the other one being credited!



## The transactions on HFSQL

Each write operation performed during a transaction is stored in a specific file. The transaction can be canceled at any time: all the operations performed since the beginning of transaction will be canceled.

The transactions will be canceled in the following cases:

- program error.
- end of program without validating the transaction.
- power failure or application shutdown.

When the application is restarted, the consistency of the database can be restored:

- via "WDTRANS".
- during the first call to *HTransactionStart*.
- during the first call to *HTransactionCancel*.

Once the write operations included in the transaction are completed, the site calls *HTransactionEnd* to validate the operations of the transaction.

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

## The replication

---

The data replication is a very powerful feature. The replication is the operation allowing the maintenance of remote databases with identical structures. Each one of these databases evolves independently.

Via the replication, the operations performed on each one of the databases are applied to all the other databases.

WebDev allows you to easily perform these operations.

WebDev proposes two types of replication:

- **The logged replication** (based on the log process). This type of replication is used to replicate the HFSQL databases between themselves. This type of replication can be implemented by the WLanguage functions or by WDReplic.
- **The universal replication** that is used to replicate any type of database (a HFSQL database with an Oracle database for example).

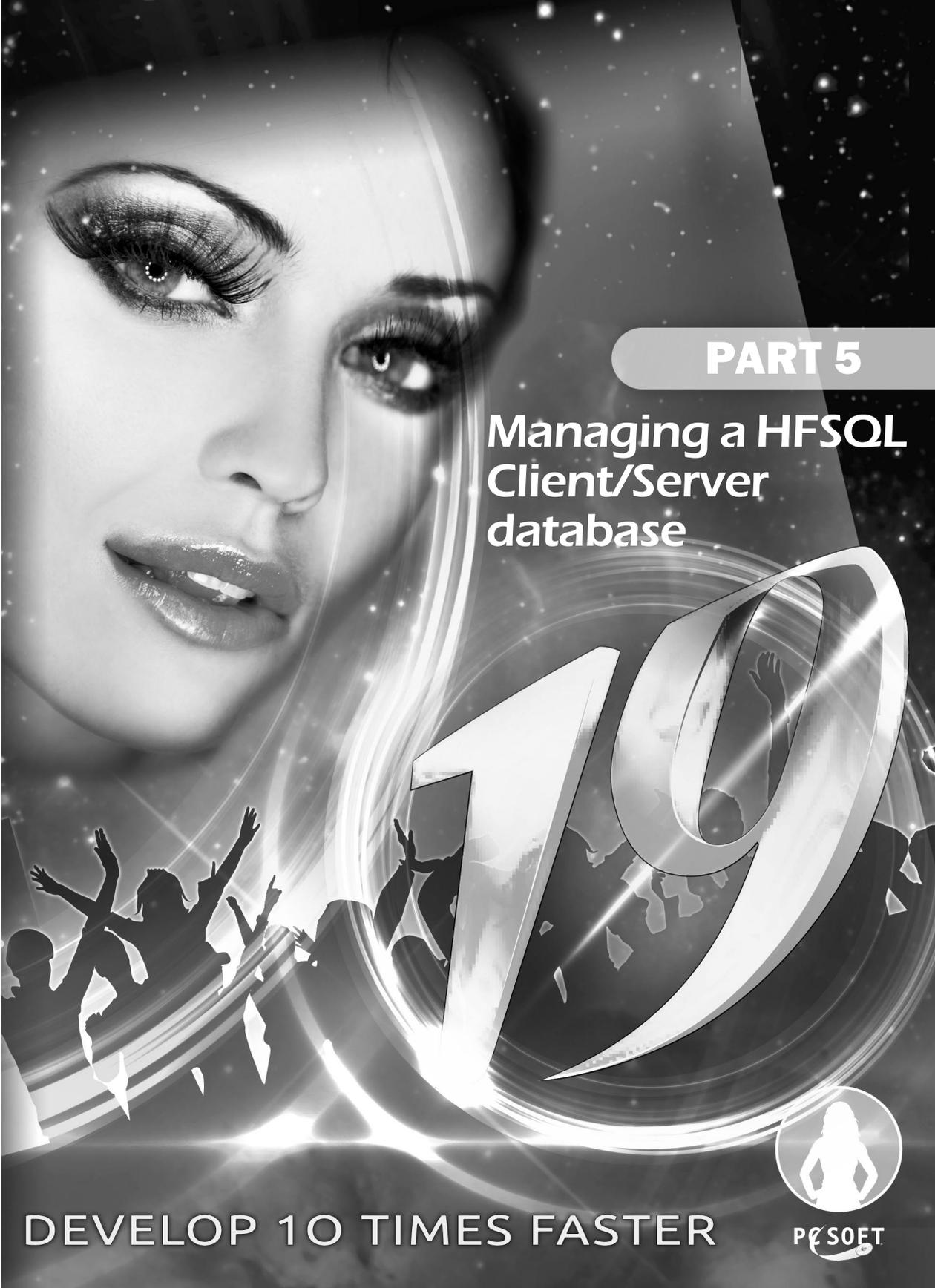
## The triggers

---

A trigger is used to automatically activate an action when an addition, a modification or a deletion is performed in a HFSQL data file.

During an operation (addition, modification or deletion) performed on a HFSQL data file, an action can be run before or after the execution of this operation. For example, when performing a deletion in a data file, you have the ability to ask for the confirmation before deleting the record.

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



**PART 5**

**Managing a HFSQL  
Client/Server  
database**

**10**

**DEVELOP 10 TIMES FASTER**





# LESSON 5.1. INTRODUCTION

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

---

- Principle of Client/Server
- Why switch a site to HFSQL Client/Server?

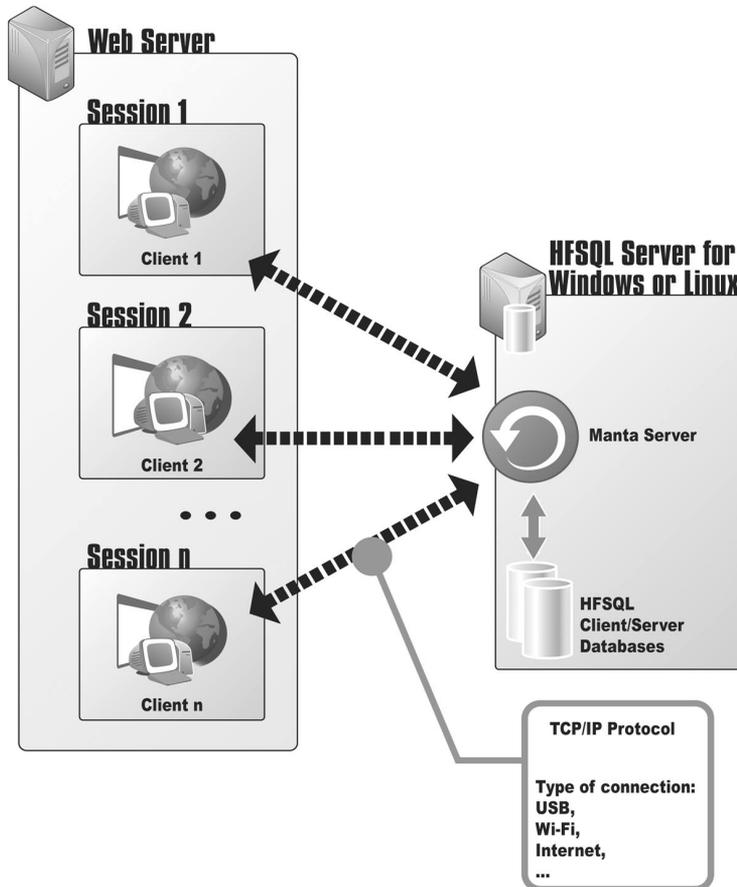


Estimated time: 5 min

## Overview

WebDev allows you to create applications that access HFSQL Client/Server databases.

A HFSQL Client/Server site consists in running the site on different user computers (called client computers) and in storing the databases and the processes on a server. This operating mode makes response times faster and more reliable, and it simplifies the maintenance of the database.



WebDev allows you to:

- create a HFSQL Client/Server site from scratch.
- modify an existing WebDev site into a HFSQL Client/Server site.

## **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:

- The use of HFSQL Client/Server is more secured (use of a login and password and definition of rights granted to the users).
- No management of directories: all the database files are grouped at the same location.
- The databases in Client/Server mode can be used via an Internet connection.

## LESSON 5.2. IMPLEMENTING A HFSQL CLIENT/SERVER SITE

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

---

- Installing a local HFSQL server
- Creating a site in HFSQL Client/Server mode
- Migrating a site to HFSQL Client/Server mode
- Using the HFSQL Control Center
- Features available in HFSQL Client/Server mode



Estimated time: 10 min

## Overview

---

In this lesson, we are going to perform all the operations required to develop and deploy a HFSQL Client/Server site.

## Installing a local HFSQL server

---

The first operation before developing a HFSQL Client/Server site consists in installing a HFSQL server.

This server can be installed locally on the development computer (that's what we are going to do). In deployment, this server can be installed on a specific computer.

The setup program of the HFSQL server is available on the WebDev CD. If you do not own this CD, the setup of the HFSQL server is also available from our Web site ([www.windev.com](http://www.windev.com)).

To install the HFSQL server locally:

1. Start the setup program of WebDev.
2. Choose "Install or update a HFSQL Client/Server server".
3. Accept the license agreement.
4. Choose the platform ("For Windows on this computer").
5. Choose "Install a new server".
6. Select the setup directory and specify the name of the server and the port. The port 4900 will be used by default. You have the ability to install the HFSQL Control Center if this one is not found or not accessible from your computer.



Caution!

The HFSQL Control Center is required to manage the HFSQL Client/Server database.

7. The setup is performed. By default, to connect to the server in administrator mode, use the "admin" user without password.

## Creating a site in HFSQL Client/Server mode

---

The method for creating a HFSQL Client/Server site is identical to the method for creating any WebDev site.

You must:

1. Create the project by requesting to create a new database.
2. Create the analysis by specifying that the databases used by the project will be "HFSQL Client/Server" databases.
3. Specify the characteristics of the connection to the HFSQL Client/Server server that will be used.
4. When creating a data file in the analysis, indicate that this file is in HFSQL Client/Server mode and specify the connection used.

## Migrate a HFSQL Classic site to HFSQL Client/Server mode

### Overview

Migrating a HFSQL Classic site to Client/Server mode is the most common operation.

WebDev proposes several methods to perform this migration on the development computer:

- perform the migration from the data model editor.
- perform the migration from the HFSQL Control Center.

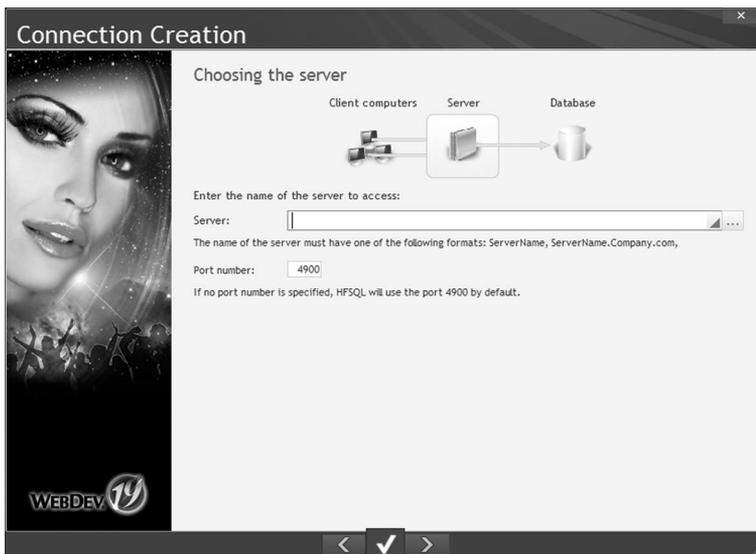
We will be using the first method to migrate the site for library management that was created in part 2 of this tutorial.

### Migrating the example

A version of the project studied in part 2 is supplied with this tutorial. We are going to migrate this project and to run its test in Client/Server mode.

To migrate the project:

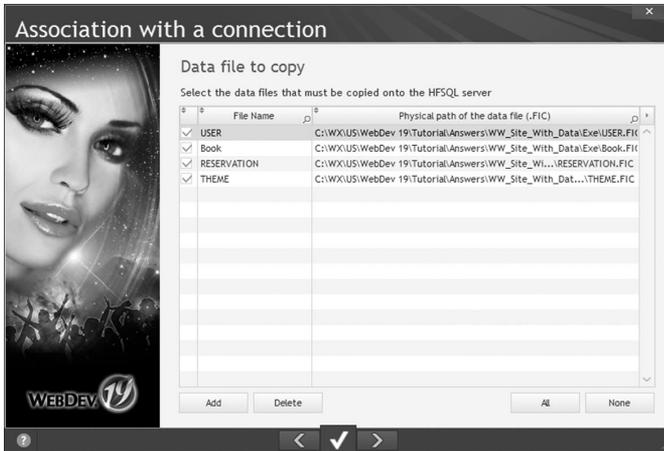
1. Close the current project if necessary.
2. In the home window, click "Tutorial" and select "Site with data (Answer)".  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Site with data (Answer)".
3. Load the analysis of your project in the data model editor: click  among the quick access buttons of the WebDev menu. The data model editor is displayed.
4. On the "Analysis" pane, in the "Connection" group, click "New connection". A wizard is opened, allowing you to create a connection.
5. Select the type of connection to create: "HFSQL Client/Server". Go to the next screen.



6. In the following planes, specify:
  - the name of the server (name of your computer for example) and the port number,
  - the name of the user and his password (leave this information empty to use the administrator).



11. Select the data files that will be copied onto the server: in our case, all the data files found in the EXE directory.



Go to the next screen and validate.

12. The data files of the analysis are automatically transformed into HFSQL Client/Server data files and associated with the selected connection.

13. Generate the analysis.



Notes

**HOpenConnection** can be used to go back to HFSQL Classic mode: all you have to do is specify the path of the directory containing the HFSQL Classic data files. When switching to HFSQL Client/Server, check the code of your project: in HFSQL Client/Server mode, **HSubstDir**, **HChangeDir**, ... are useless.

14. The development project was migrated. You may also have to migrate the deployed application (if the deployed application uses HFSQL Classic data files for example). This operation is configured when creating the setup program of the application.

## Features available in HFSQL Client/Server mode

HFSQL Client/Server proposes several features specific to the Client/Server mode:

- Transactions
- Log
- Stored procedures
- Triggers
- Hot automatic data modification
- Hot reindexing
- Scheduled backups
- Incremental backups
- Server 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 5.3. MANAGING A CLIENT/SERVER DATABASE

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

---

- The HFSQL Control Center
- Creating a user in the HFSQL Control Center
- Saving the database



Estimated time: 20 min

## Overview

---

Now that we know how to create and migrate a site in HFSQL Client/Server, let's see how it can be managed.

Indeed, a Client/Server database requires:

- a specific configuration of the computers (setup of a HFSQL server, ...)
- a management performed via the HFSQL Control Center.

## Configuring the computers

---

To use a HFSQL Client/Server database, a HFSQL server must be installed on the server. Several HFSQL servers that use different ports can be installed on the same computer. One or more databases can be installed on each server.

For example, a test HFSQL server that includes a test database and a production HFSQL server that uses a different port can be installed **on the same computer**.

## The HFSQL Control Center

---

The HFSQL Control Center is used to perform all the management operations on the HFSQL Client/Server servers and databases.

We are going to present the most important features.

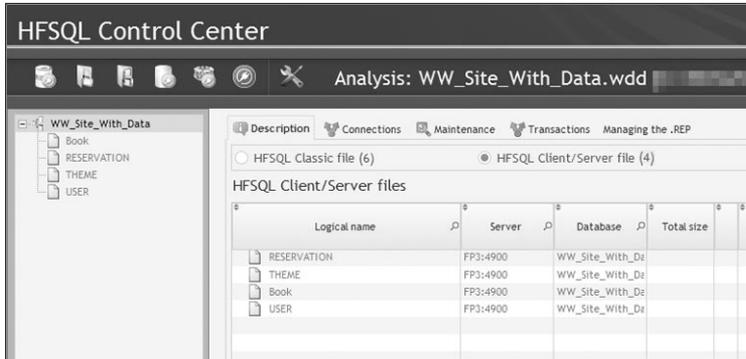
To start the HFSQL Control Center and to access the data:

1. In the WebDev menu, on the "Tools" pane, in the "Database" group, click "HFSQL". The home window of the HFSQL Control Center is displayed. The analysis of the current project is automatically selected.



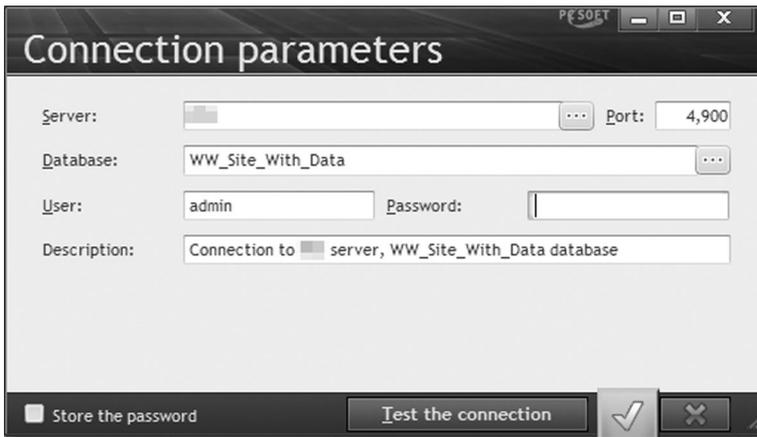
Validate the screen. The HFSQL Control Center is displayed.

2. Click "HFSQL Client/Server file". The list of data files in HFSQL Client/Server format is displayed.



The Control Center lists the data files found in the analysis linked to the current project. No connection is established.

3. To view the data files, double-click one of the data files in the list on the left. The HFSQL Control Center displays a connection window used to establish the effective connection to the HFSQL Client/Server server used.



Specify the password if necessary and validate.

4. The information about the different data files that use this connection is displayed.
- The "Content" tab displays the records found in the data files.
  - The "Description" tab gives information about the data files (list of items as well as their name, their type, ...).

The entire HFSQL Client/Server database can be managed from the HFSQL Control Center.

## Creating a user account in the HFSQL Control Center

A single user account is created when installing a HFSQL server and when creating a database: the administrator account ("Admin" login without password).

Using a user account allows you to secure the access to the data. Indeed, all the users of the application 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 the application).

Do not confuse the management of rights for the Client/Server databases with the user groupware.

Some users may not have the rights to write into some data files for example.

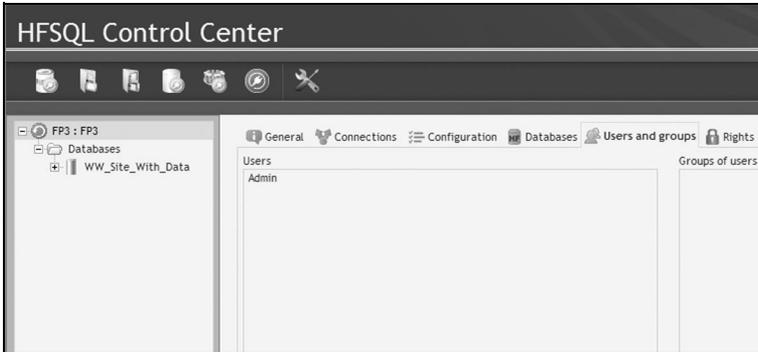
To run a simple test, we are going to create a user and allow this user to view the records found in the Book file.

1. In the HFSQL Control Center, click  and validate the closing of connections if necessary.
2. The home window of the HFSQL Control Center is displayed.



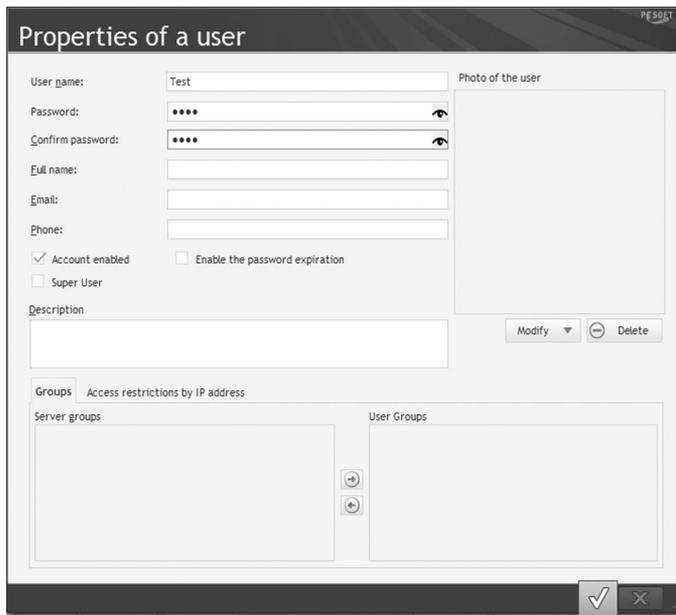
3. The option "Connect to a HFSQL server" is selected by default. Specify the characteristics of the server that was installed in the previous lesson and validate.
4. The characteristics of the HFSQL server are displayed:
  - in the panel on the left, the computer, the name of the HFSQL server and the list of databases found on this server.
  - in the right section of the screen, the different tabs are used to manage the HFSQL server.
5. In the right section of the screen, select the "Users and Groups" tab. This tab is used to manage the users of the server.

Only the "Admin" user exists at this time.



6. To create a new user, click the "New" button (found below the list of users). The screen used to define the characteristics of the user is displayed.

Enter the following information:



(use "Test" as password)



Notes

Several characteristics can be noticed :

- **Super User:** The users defined as "Super user" are allowed to perform all the actions on the server, on the databases and on all the data files.
- **Account enabled:** If this option is not checked, the user exists but he is not enabled (collaborators on holiday for example).
- **Password expiration:** You have the ability to specify a password valid for a limited number of days.

7. Validate the creation of the user. By default, no rights are granted to this user.

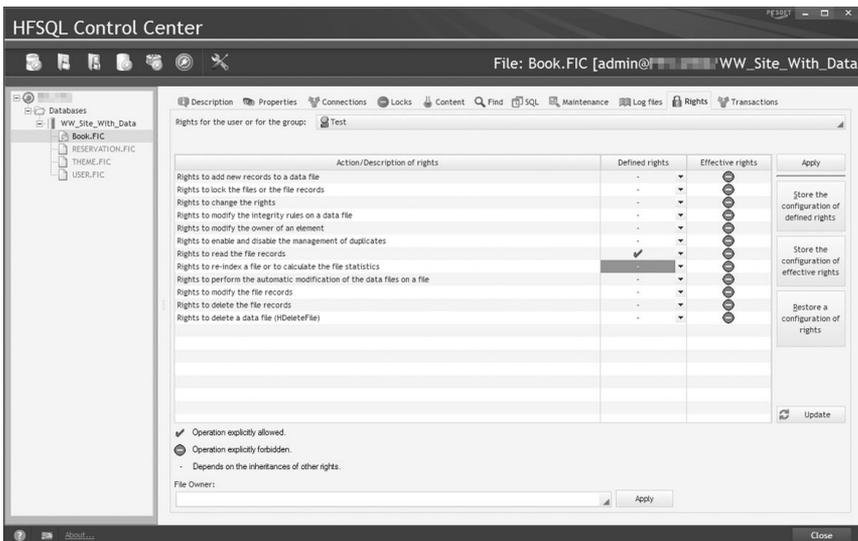
We are now going to grant rights to the user: the "Test" user can connect to the database and view the Book file.

To grant the rights to connect to the database:

1. In the HFSQL Control Center, select the "WW\_Site\_With\_Data" database.
2. Click the "Rights" tab.
3. Select the "Test" user in the list of users.
4. In the list of rights, select "Rights to connect to the server (encrypted and unencrypted connection)". Select the green checkmark in the "Defined rights" column.
5. Click the "Apply" button. The rights become effective.

To grant the rights to read the Book file:

1. In the HFSQL Control Center, select the "WW\_Site\_With\_Data" database, then select the "Book" file.
2. Click the "Rights" tab.
3. Select the "Test" user in the list of users.
4. In the list of rights, select "Rights to read the file records" and select the green checkmark in the "Defined rights" column.



5. Click the "Apply" button. The rights become effective.

Similarly, the rights can be defined:

- on the HFSQL server
- on the database
- on the database files.

In our example, the "Test" user will be able to browse the records found in the Book file. If this user tries to perform another action, a message will be displayed: "The Test user does not have sufficient rights to XXXX" (where XXXX corresponds to the action performed).

Once the account is created, it can be used when the application connects to the server (when *HOpenConnection* is used).



Notes

The users and their rights can also be managed by programming with the WLanguage functions. See the online help for more details.

## Saving the database

---

The backup of the database can be performed via the "Backups" tab available when selecting the database in the left pane of the HFSQL Control Center.

## Conclusion

---

The HFSQL Control Center is a complete tool for managing the databases, allowing you to:

- stop or restart a server if a problem occurs,
- manage the users and their rights,
- reindex the data files if necessary,
- perform backups of the database.

**The HFSQL Control Center is a redistributable tool that can be installed on the computers of the users who work with HFSQL Client/Server databases. The HFSQL Control Center must be used by the administrator of the database.**



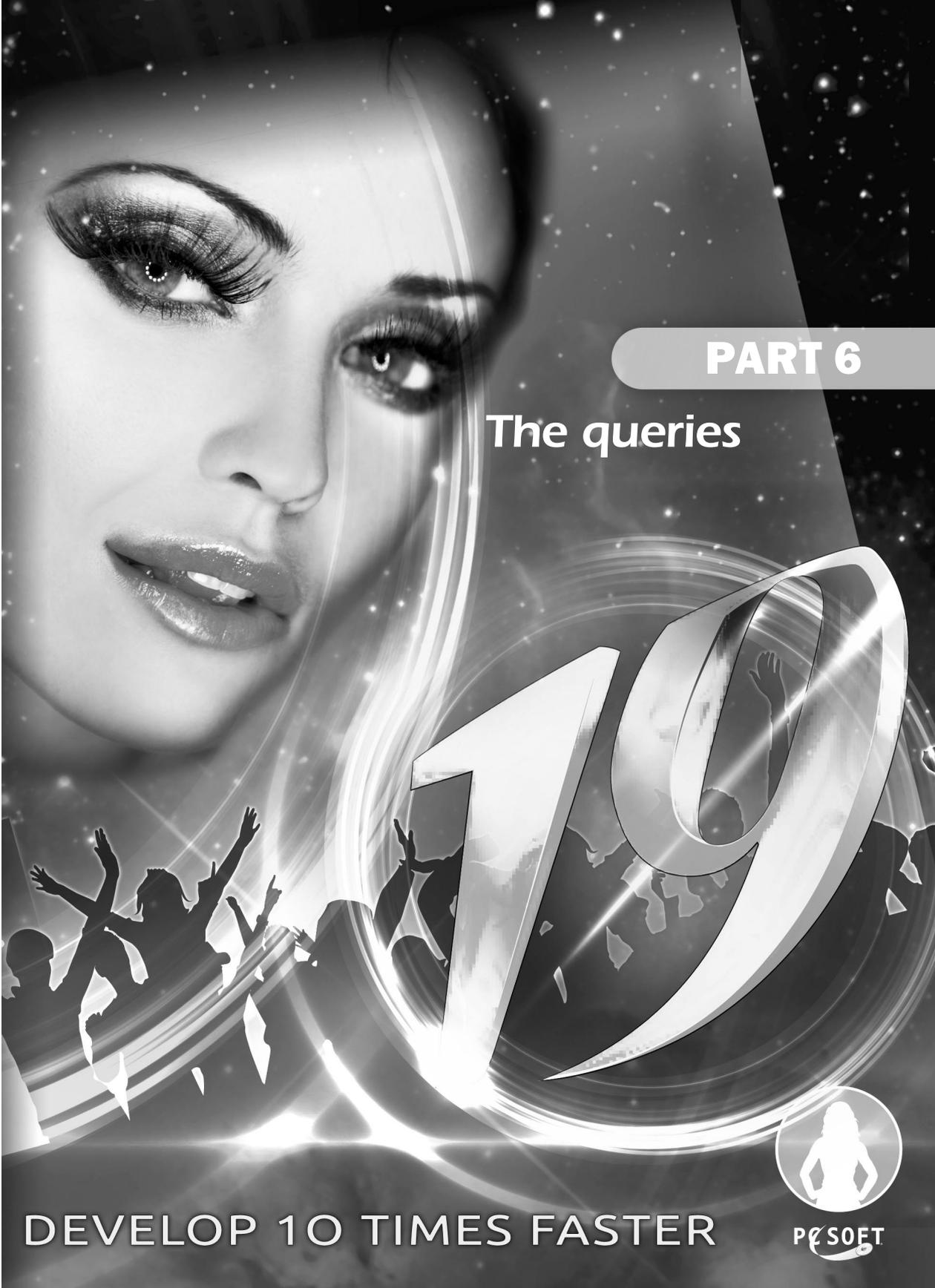
Notes

Installing a site that uses a HFSQL Client/Server HyperFileSQL requires:

- The setup of a HFSQL Client/Server server. This server must be accessible by the site.
- The setup of the data files on the server.

See the online help for more details.





**PART 6**

**The queries**

**10**

**DEVELOP 10 TIMES FASTER**





# LESSON 6.1. CREATE YOUR FIRST QUERY

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

---

- Operations performed in the query editor.
- Creating a query step by step.



Estimated time: 20 min

## Introduction

---

A query is used to define an interrogation on data files. The query editor allows you to create (without programming) the following types of queries:

- the list of classified ads in a city starting with the letter "P".
- the number of classified ads per city.
- the total of all prices proposed by the agencies.

And this, for all types of databases (HFSQL, Oracle, ...)!

We will now take a look at a simple example to explain how the query editor can be used. An advanced use of the query editor will be presented later in this tutorial.

In this lesson, you will learn how to:

- Create a simple query corresponding to a list with selection and a sort.
- Create a query used to perform a count.
- Create a multi-file query.

► To open the example project:

1. Close (if necessary) the current project to display the home window.
  2. In the home window, click "Tutorial" and select the "Running queries (Exercise)" project.
- Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Running queries (Exercise)".

## Your first query

---

We are now going to list the classified ads for the cities starting with the letter "P". We are going to select all the classified ads placed by people living in cities starting with the letter "P".

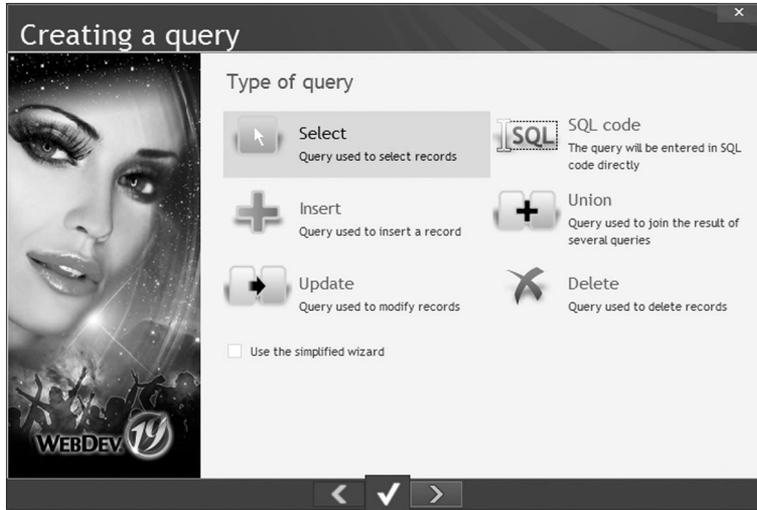
► Create a new query:

1. Click  among the quick access buttons.
2. Click "Query" in the wheel that is displayed. The wizard for query creation starts.

You can use the wizard for query creation or you can directly enter the SQL code of the query ("SQL code" option).

The wizard allows you to create the following types of queries:

- "Select (SELECT)": interrogation.
- "Insert": addition into a data file.
- "Update": modification in a data file.
- "SQL code": customized query in SQL code.
- "Union (UNION)": combination of queries.
- "Delete": deletion from a data file.

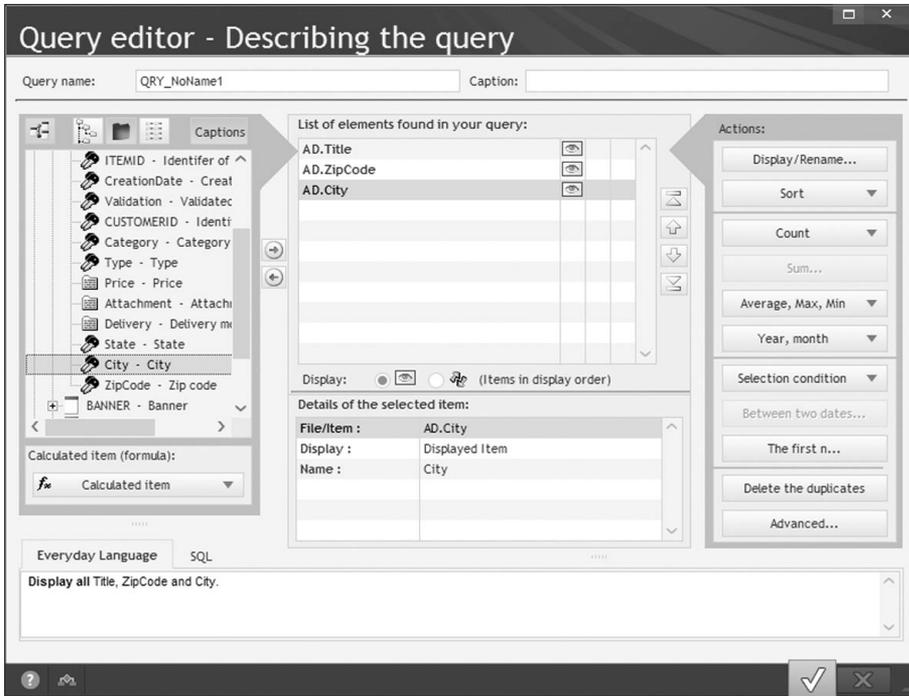
**Caution!**

The direct input of SQL code is intended for the users who are familiar with the SQL language!

See the online help (keyword: "SQL, SQL language") for more details.

In the following lessons, you will learn how to use the query editor by creating simple queries: selecting customers, performing a count, ... Then, you will have the ability to improve your knowledge by using all types of queries.

- ▶ To list the classified ads for a city starting with the letter "P", we must create a Select query. Go to the next plane.
- ▶ Select the items that will be included in the result of the query:  
On the left, in the list of items:
  1. Select the "AD" file and expand the tree structure.
  2. Double-click the "Title", "ZipCode" and "City" items. These items are displayed in the middle of the window.



- ▶ To list the classified ads for a city starting with the letter "P", we are going to apply a selection condition to the city.  
The selection condition is as follows: *"We want the list of ads coming from a city starting with the letter "P"*.
- ▶ Select the "City" item in the middle of the window. In the "Actions" on the right, click the "Selection condition" button then "New condition".  
The relevant item is: "AD.City".  
Select the following condition: "Starts with".  
Select "the value" and enter "P" in the "Value" control.

Describing a condition

This query will select the records for which:

The item

starts with

the value  
 the parameter

Value:

Comments:

- ▶ Validate.
- ▶ We are now going to specify the sort criterion. The result will be sorted by "Title".
- ▶ Select the "Title" item (in the middle). In the "Actions", click the "Sort" button and select "Sort on the selected item". The sort will be performed in ascending order:

Adding a sort

Choose the sort item or the sort parameter

Item to sort:

Parameter name:

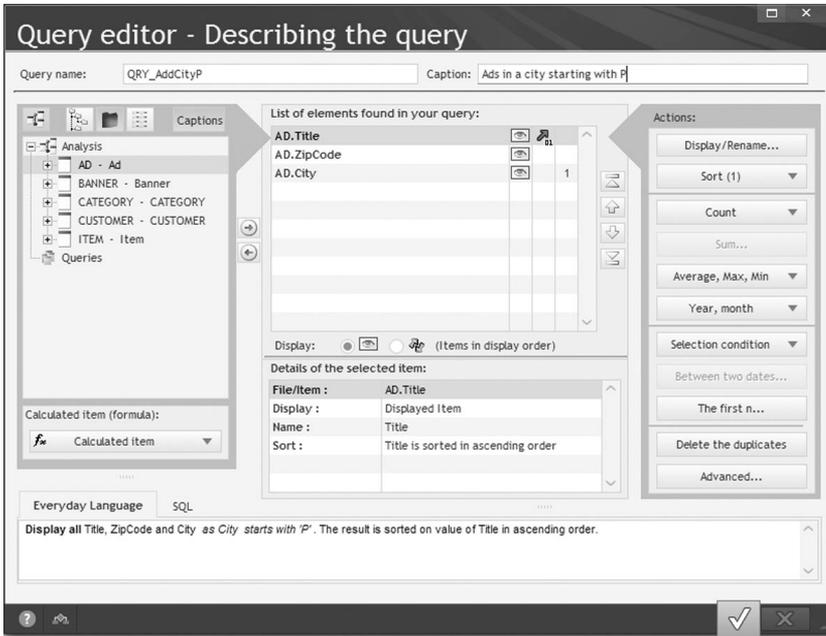
A comment can be added to this parameter...

Choose the sort order

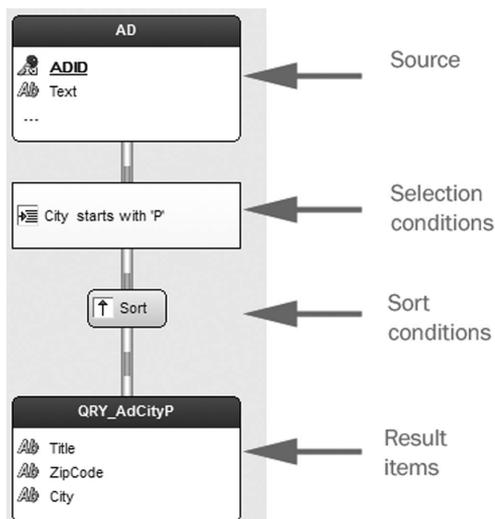
Ascending   
 Descending

- ▶ Validate. A red arrow is displayed on the right of the "Title" item. This arrow indicates that an ascending order is performed on this item.

- ▶ The query is created. We are going to name it and to save it.
  1. At the top of the screen, enter the name: "QRY\_AdCityP".
  2. Enter the caption: "Ads in a city starting with P".
  3. Validate.



The graphic representation of the query is displayed:



- Click  among the quick access buttons to save the query.

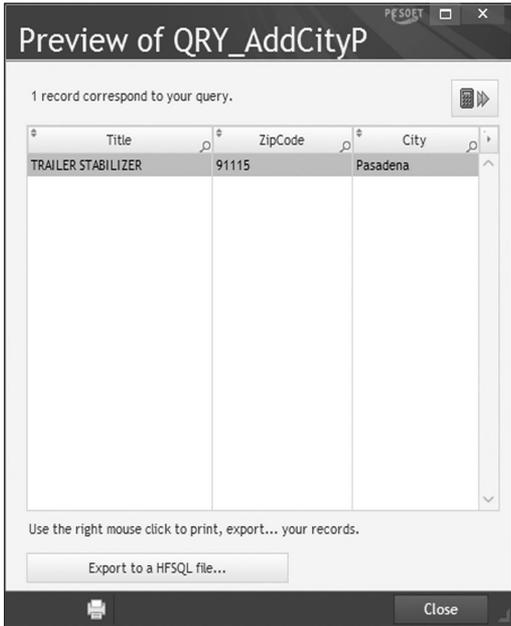
 **Notes**

Once the query is created, the AAD (Aided Application Development) may be triggered to specify that the query can be optimized.

**Query performance will be enhanced.**  
Click here to enhance performance of QRY\_AdvertSanFrancisco query.

We shall not discuss this feature in the tutorial. See the online help (keyword: AAD) for more details.

Now let's run the test of the query: click  among the quick access buttons. The result is displayed in a window:



1 record correspond to your query.

| Title              | ZipCode | City     |
|--------------------|---------|----------|
| TRAILER STABILIZER | 91115   | Pasadena |

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

Export to a HFSQL file...

Close

A popup menu is displayed when you right-click the table displaying the result of the query. The result can be exported to:

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

## LESSON 6.2. QUERIES WITH COUNT

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

---

- Operations performed in the query editor.
- Creating a query with count step by step.



Estimated time: 10 min

## Creating a query with count

We are now going to create a query used to perform a count. This query will count the number of occurrences for a value in a data file.

Let's count the number of classified ads placed per city. Which means counting the number of classified ads published per city.



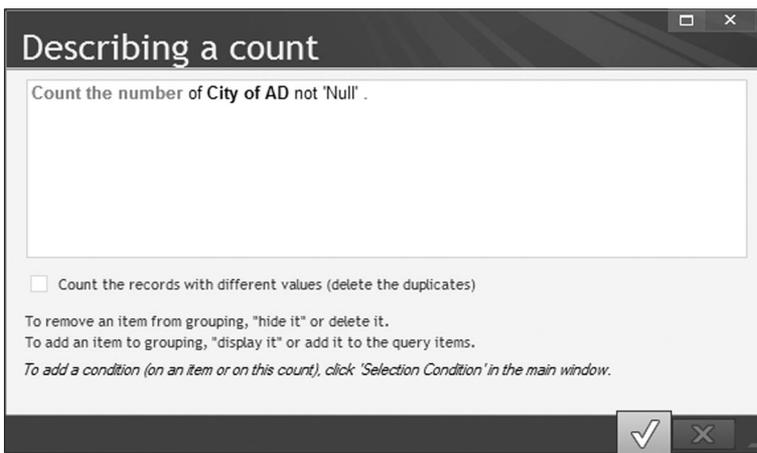
Notes

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

- ▶ To open the example project:
  1. Close (if necessary) the current project to display the home window.
  2. In the home window, click "Tutorial" and select the "Running queries (Exercise)" project. Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Running queries (Exercise)".
- ▶ Start the query editor:
  1. Click  among the quick access buttons.
  2. Click "Query" in the wheel that is displayed. The wizard for query creation starts.
  3. We are going to create a "Select" query ("Select"). Go to the next plane.

We are going to select the items that will be retrieved in the result of the query.

- ▶ To create a count item:
  1. In the list of items (on the left), double-click the "City" item (found in the "AD" file).
  2. Select the "City" item (in the middle).
  3. In the "Actions" (on the right), click the "Count" button then "Count on the selected item"

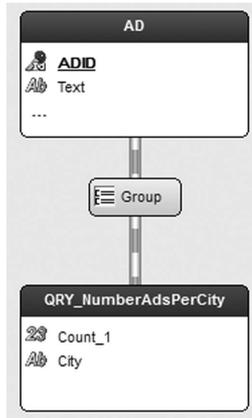


#### 4. Validate.

The "Count(AD.City)" item is displayed in the result of the query.

- ▶ We still want to display the city. In the list of items (on the left), double-click the "City" item (found in the "AD" file).
- ▶ The query is created. We are going to name it and to save it.
  1. At the top of the window, enter the name: "QRY\_NumberAdsPerCity".
  2. Enter the caption: "Number of classified ads per city".
  3. Validate.

The graphic representation of your query is displayed:



The "Count\_1" item contains the count of the above-mentioned records.

- ▶ Click  among the quick access buttons to save the query. Let's run the test of the query.
- ▶ Click  among the quick access buttons.

## LESSON 6.3. SUM QUERIES

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

---

- Operations performed in the query editor.
- Creating a "Sum" query step by step.



Estimated time: 10 min

## Creating a "Sum" query

We are now going to create a query used to perform a sum. We are going to perform a cumulative sum of a value found in a data file.

We are going to calculate the total of all the rates proposed for a classified ad per agency.

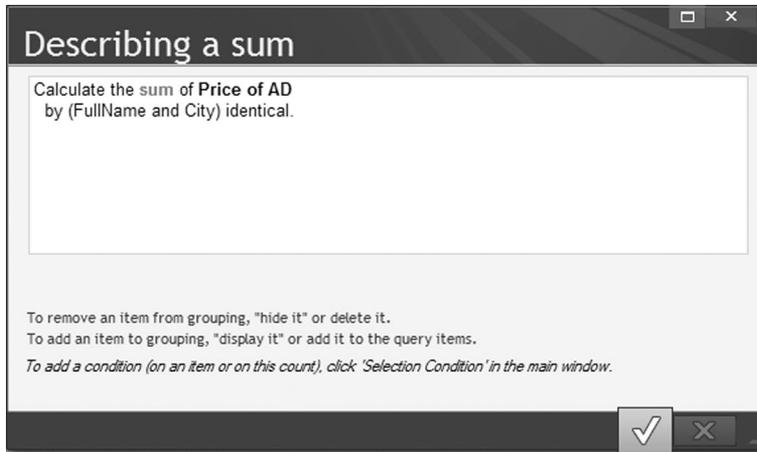


Notes

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

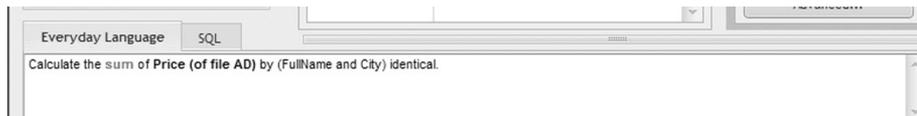
- ▶ To open the example project:
  1. Close (if necessary) the current project to display the home window.
  2. In the home window, click "Tutorial" and select the "Running queries (Exercise)" project.  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Running queries (Exercise)".
- ▶ Start the query editor:
  1. Click  among the quick access buttons.
  2. Click "Query" in the wheel that is displayed. The wizard for query creation starts.
  3. This query is a Select query. Select "Select". Go to the next screen.
- ▶ We are going to select the items that will be retrieved in the result of the query. We are going to display the name and city of the customer. On the left, double-click the "FullName" item and the "City" item (found in the "CUSTOMER" file).  
**Reminder:** the file of customers is named CUSTOMER.
- ▶ Then, we must select the item on which the **sum** will be performed.
- ▶ On the left, double-click the "Price" item (found in the "AD" file).
- ▶ To calculate the sum of the values of "Price":
  1. Select the "Price" item (in the middle).

2. In the "Actions" (on the right), select "Sum".



3. Validate. The sum of the "Price" items was added to the list of the query result.

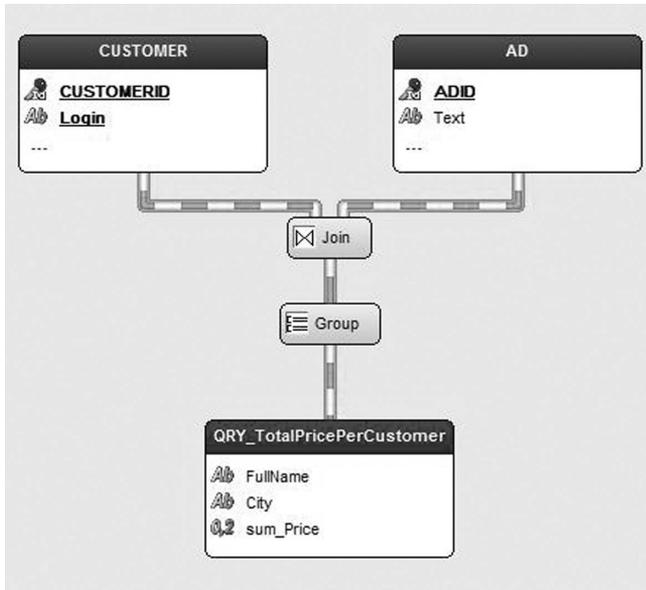
You will notice that the query editor of WebDev creates the queries in everyday's language (and in SQL language).



This method for generating queries enables you to check whether your query performs the proper operations.

- ▶ The query is created. We are going to name it and to save it.
  1. At the top of the window, enter the name: "QRY\_TotalPricePerCustomer".
  2. Enter the caption: "Total of prices proposed per customer".
  3. Validate.

The graphic representation of your query is displayed:



In this representation, two data sources (the "CUSTOMER" file and the "AD" file) are used and linked by a "join".

- ▶ Click  among the quick access buttons to save the query.
- ▶ Click  among the quick access buttons to run the test of the query.

## LESSON 6.4. THE SQL LANGUAGE

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

---

- The main commands of SQL language.



Estimated time: 15 min

## Source code of a query

In WebDev, a graphic representation is used to represent the query in the editor. We shall now discover the source code corresponding to a query. The language used to represent a query is the SQL language. This is just an overview of SQL, not a full course (entire books are devoted to SQL!).

Let's take a look at the SQL code of the query that was created in the previous lesson.

- ▶ To open the example project:
  1. Close (if necessary) the current project to display the home window.
  2. In the home window, click "Tutorial" and select the "Running queries (Exercise)" project.  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Running queries (Exercise)".
- ▶ Open the "QRY\_TotalPricePerCustomer" query. This query was created in the previous lessons. If this query was not created, open the "QRY\_TotalPricePerCustomer" query found in the answer.  
To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Running queries (Answer)".
- ▶ Display the popup menu of the query (right mouse click on the query graph) and select "SQL code".

```
SQL Code of QRY_TotalPricePerCustomer (server)
SELECT
    CUSTOMER.FullName AS FullName,
    CUSTOMER.City AS City,
    SUM(AD.Price) AS sum_Price
FROM
    CUSTOMER,
    AD
WHERE
    AD.CUSTOMERID = CUSTOMER.CUSTOMERID

GROUP BY
    CUSTOMER.FullName,
    CUSTOMER.City
```

The query in SQL language includes different keywords:

- the SELECT statement
- the FROM statement
- the WHERE clause
- the GROUP BY clause



Caution!

Make sure that you follow the sequence of commands. This sequence cannot be reversed.

## The SQL commands

---

### The SELECT statement

The SELECT statement is used to define the list of items and calculations that must be displayed in the result. The syntax is as follows:

```
SELECT NAMEITEM1, NAMEITEM2 ...
```

Each item can be renamed (we talk of alias) by using the AS keyword. For example:

```
CUSTOMER.CUSTOMERNAME AS NAME
```

In this example, the *CustomerName* item (found in CUSTOMER file) is renamed to *Name*. *ItemNameXXX* can be either an item associated with a data file, or a calculation formula.

### The FROM statement

The FROM statement is used to define the list of data files from which the items and calculations will be extracted. The syntax is as follows :

```
FROM FILE1 , FILE2 ...
```

*FileXXX* corresponds to one of the data files found in the analysis. Using **AS** is allowed (see "The SELECT statement").

### The WHERE clause

The WHERE statement contains the selection conditions as well as the join conditions. These conditions are used to filter the records coming from the database files. Only the records that match the conditions will be displayed in the result.

A join is used to associate a record found in a file with a record found in a linked file. For example, a join between the CUSTOMER and ORDERS files will be used to find out all the orders of each customer. The syntax is as follows:

```
WHERE CONDITION1 AND/OR CONDITION2
```

The AND and OR operators are used to perform a logical operation between two conditions.

- The AND operator indicates that the two conditions must be matched at the same time.
- The OR operator indicates that one of the conditions or both conditions must be matched.

For example:

```
WHERE DATE>="20030523" AND STATE=75
```

The WHERE statement is optional. If this command is not specified, all the records found in all the data files will be selected.

### The GROUP BY clause

The GROUP BY clause is used to specify the group items or the group formulas when performing a calculation. For example, in the query that was created beforehand, the calculation was performed on the total of all the rates proposed by the agents. Therefore, the rates were grouped by agent.

The syntax is as follows:

```
GROUP BY ITEM1, ITEM2 ...
```

*ItemXXX* corresponds to an item found in one of the data files or to a formula.

### The ORDER BY clause

The ORDER BY clause is used to specify the sort criteria for the query. The sort criteria will be used to sort the result of the query. The syntax is as follows:

```
ORDER BY ITEM1, ITEM2 ...
```

*ItemXXX* corresponds to an item found in one of the data files or to a formula.



Notes

The query editor automatically generates the SQL code, it's a good method for learning the SQL language!

## LESSON 6.5. USING QUERIES

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

---

- Using queries in the reports.
- Using queries in the tables.
- Using queries in WLanguage.



Estimated time: 5 min

## Using a query from a report

---

You have the ability to choose the data source when creating a report. However, the data source can be a query. Therefore, we could create a report based on the result of a query (see "The reports", page 297).

## Using a query from a table or from a loop

---

When creating a table or a loop, you have the ability to define the data source used to fill the control. This source can be a data file or a query.

## Using a query from WLanguage

---

A query can also be used like any data file.

You must :

- initialize the execution of your query with **HExecuteQuery**.
- when the browse is over, use **HCancelDeclaration** (optional, used to cancel the definition of the query).

Your browse can be performed as usual with FOR EACH for example:

```
// Initializes the query
HExecuteQuery(MyQuery)
IF ErrorOccurred THEN
    Error("Error while initializing the query"+ CR+HErrorInfo())
    RETURN
END
FOR EACH MyQuery
    // Process on the query record
    // read the next record
END
HCancelDeclaration(MyQuery) // Optional
```



**PART 7**

**The reports**

**10**

**DEVELOP 10 TIMES FASTER**



**PC SOFT**



# LESSON 7.1. INTRODUCTION

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

---

- The principle for printing from Internet
- What is a report?



Estimated time: 10 min

## Principles for printing on Internet

---

We should not talk about "printing" on Internet. Indeed, when a document must be "printed", a file is generated beforehand (HTML, PDF or XML) then it is transmitted to the browser. As soon as the file transfer is ended, the Web user decides whether the document must be printed or not.

However, a "print" remains available on the server. However, the document that is printed (on the server printer or on a shared network printer) will not be accessible to the Web user.

The printed or generated document is prepared and formatted by the report editor. The information found in the document can come from a database

Two types of print are available:

1. The direct print (on the server printer).
2. The generation of various documents (HTML, PDF, ...).

### Direct print

The direct print consists in printing on a printer directly. This printer is plugged into the server or accessible from the network.

The direct print with WebDev is recommended for Intranet or Extranet only. This type of print is used to print the logs for customer connections, to directly print the order on the printer of the sales department.



Notes

The direct print can also be used to send a fax from the Web server (see the documentation about *iPreview*).

### Generation of documents (HTML, PDF, XML), ...

Printing from HTML, PDF or XML documents consists in creating a document and in displaying it on the browser of the Web user. To create this document, all you have to do is use the report editor of WebDev. A formatted file can be "generated" by your WebDev site from a data source. The file can be in HTML, PDF, RTF or XML format.

The main benefit of this method is that the generated file can be transmitted to the browser. The Web user can print the document or store it on his computer.

Printing in a file can be used for an Internet site and for an Intranet/Extranet site (to transmit an order form in PDF format to the Web user, ...).

## Basic vocabulary

A **Report** is the name given to the representation of a print. A report is created in the WebDev editor. A document is created when running (or editing) the report.

WebDev proposes several types of reports:

- Blank
- Form
- Array
- Crosstab
- Treeview table
- Organizer or scheduler
- Mailshot
- Label
- Form
- Multicolumn
- Composite

A **Block** is the name given to a section of the report. A report includes a set of blocks.

|                             |  |               |              |
|-----------------------------|--|---------------|--------------|
| Start of Document           |  |               |              |
| Page Header                 |  |               |              |
| Start of Break 1 (FullName) | Full name: _____<br>Login: Chris<br>Zip code: 62702<br>City: Springfield |               |              |
| Start of Break 2 (ITEMID)   | Category: Associations<br>Identifier of AdvCateg: 0                      |               |              |
| Body                        | Identifier of Ad   | Creation date | VOLVO 440 GL |
| End of Break 2              | 344  | 09/05/2011    |              |
| End of Break 1              |  |               |              |
| Page Footer                 |  |               |              |
| End of Document             | Number of rows: = 999,999  |               |              |

The different blocks of a report:

- **Start of document:** The Start of Document block is printed once at the beginning of the report execution.  
In most cases, this block contains the title of the document as well as today's date and time.
- **Page header:** The Page Header block is printed on each new page. This block is run after the Start of Document block if it exists.  
The Page Header block contains information that must be printed at the top of each page, the title or a preset control indicating the page number for example.
- **Break header:** The Break Header block is optional. It is created if a break is requested on a sort argument. The Break Footer block necessarily exists if this block exists.  
This block is run whenever the value of the sort argument changes, before the execution of the first iteration of the Body block for a new value of the sort argument.
- **Body:** The Body block corresponds to the main block of the report. This block is necessarily found in each report. This block is printed as many times as the number of records to print.
- **Break footer:** The Break Footer block is optional. It is created if a break is requested on a sort argument. The Break Header block necessarily exists if this block exists.  
This block is run whenever the value of the sort argument changes, after the last iteration of the Body block for a value of the sort argument. In most cases, it contains the counter or total controls for the records corresponding to the sort argument.
- **Page footer:** The Page Footer block is run at the end of each page.  
The Page Footer block contains information that must be printed at the bottom of each page (counters and/or totals generated when creating the document or added thereafter, or the number of the current page according to the total number of pages for example).
- **End of document:** The End of Document block is run once at the end of the report execution.  
In most cases, the End of Document block contains counters and/or totals (generated when creating the document or added thereafter).

Other blocks are available but we won't go into details about them:

- the iteration block (the Iteration block is a free block, created manually in the report editor).
- body complement (the Complement block is used to automatically extend the print until it reaches the end of the page).

Some of these blocks are optional. They can be deleted when creating the report. Only the "Body" block is mandatory.

You also have the ability to print reports with bar codes (automatically generated by WebDev), reports with labels ...

# LESSON 7.2. YOUR FIRST REPORT

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

---

- Creating a report step by step.



Estimated time: 25 min

## Overview

A sample project containing the necessary data was created beforehand.

- ▶ To open the example project:
  1. Close (if necessary) the current project to display the home window.
  2. In the home window, click "Tutorial" and select the "Printing reports (Exercise)" project.
 Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Printing reports (Exercise)".



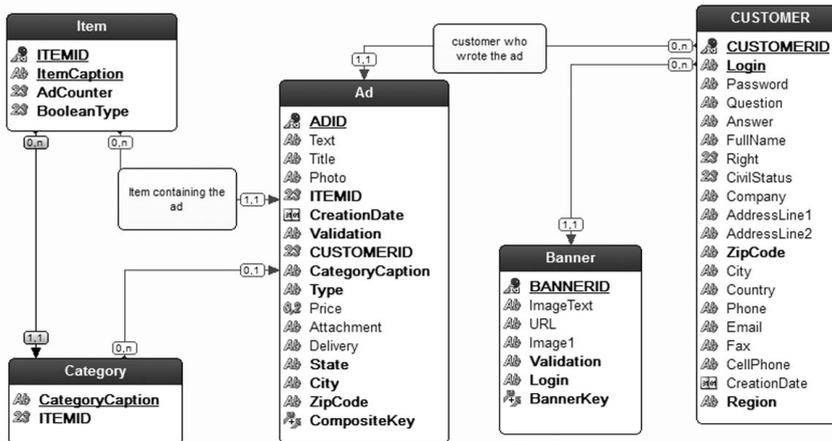
Notes

If you don't want to perform the operations of this lesson, a corrected project is available. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Printing reports (Answer)".

Simple data files have been used in our example:

- a CUSTOMER file
- an AD file
- an ITEM file
- a CATEGORY file
- a BANNER file

The logical modeling (MERISE) of the data used (called LDM) is presented below:



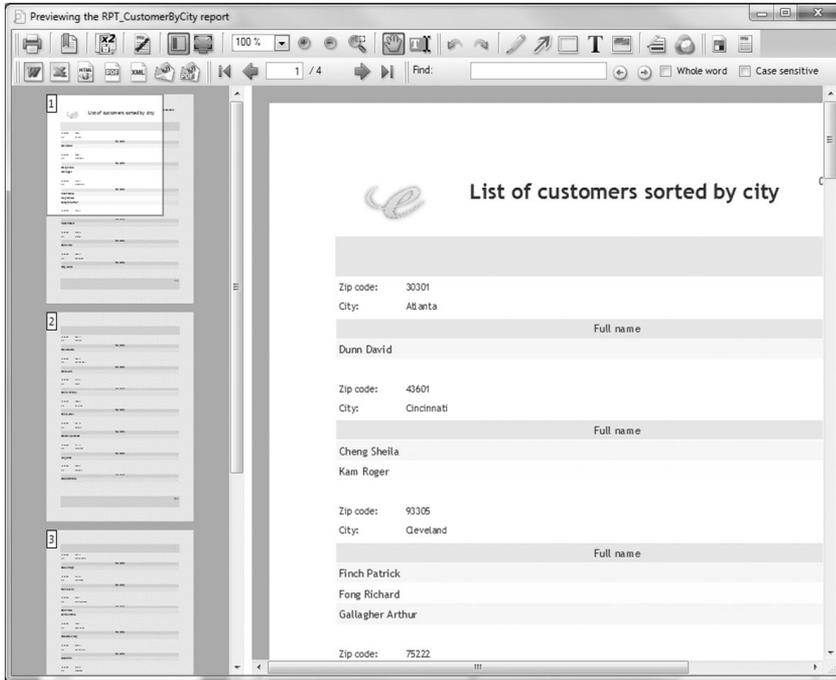
## The reports to create

- ▶ To learn how to use the report editor, we are going create the following examples:
  - List of customers (who placed an ad) sorted by city.
  - List of ads per customer.
  - Number of ads per year and per category.

## Creating your first report

### Creating the report: List of customers sorted by city

We are going to print the list of customers (who placed an ad) sorted by city. To do so, we are going to use a table report in order to clearly represent a list of data.



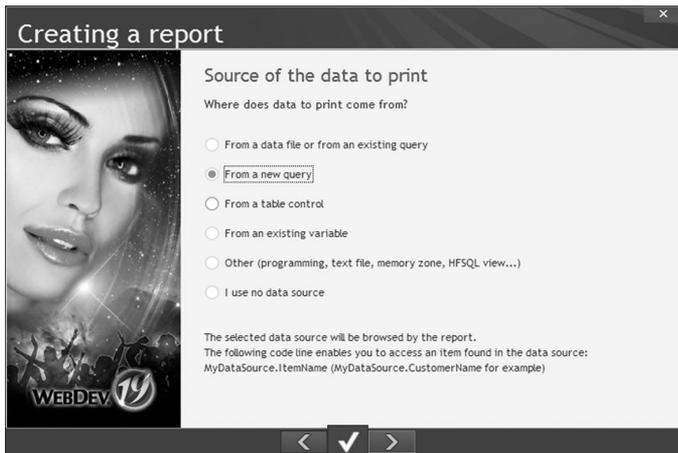
► To create a report:

1. Click  among the quick access buttons.
2. Click "Report" in the wheel that is displayed. The wizard for report creation starts and proposes several representations for the report:



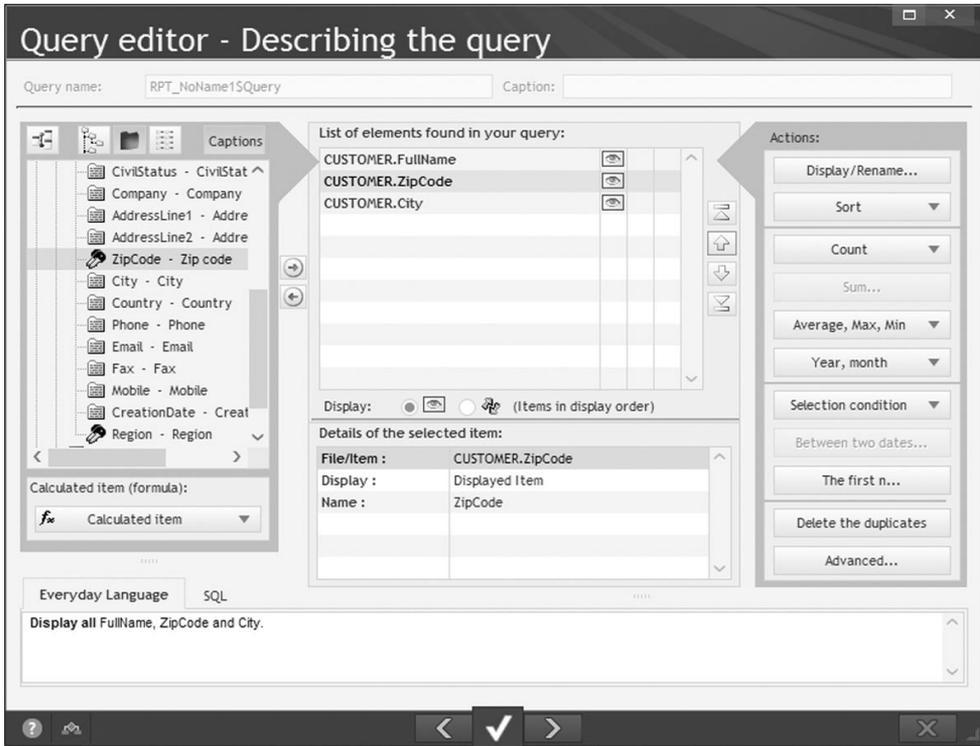
► Select "Table". Go to the next screen.

► The wizard proposes to choose the data source of the report:



► Select "From a new query" and go to the next screen. The query does not exist so let's create it. The wizard for query creation starts. This screen is used to select the files containing the data that will be printed. We are going to print the data coming from the CUSTOMER file.

- ▶ Click the "+" sign found to the left of the CUSTOMER file. Select the items that will be printed (via a double click):
  - FullName
  - ZipCode
  - City



- ▶ To sort by city, a sort criterion must be chosen:
  1. Select the sort item: click "CUSTOMER.City" in the middle section of the screen.
  2. In the right section, click the "Sort" button and choose "Sort on the selected item".

Adding a sort

Choose the sort item or the sort parameter

Item to sort: CUSTOMER.City

Parameter name:

A comment can be added to this parameter...

Choose the sort order

Ascending 

Descending 

3. Choose "Ascending" and validate. A red arrow is displayed on the right of the "City" item. This arrow indicates that an ascending order is performed on this item.
- ▶ Go to the next screen to validate the creation of the query. The next screen asks you to specify whether there is a break.

Creating a report

Breaks

Do you want to add some breaks into the report?

Yes

No

The breaks are used to group the records containing an item with the same value and to perform sub-totals.

The report must be sorted according to this item.

Break header ->

End of break (sub-totals) ->

Break header ->



Notes

### What is a break?

A **Break** is an operation used to group the records (or rows) according to one or more criteria. Caution, the records (or rows) will be printed.

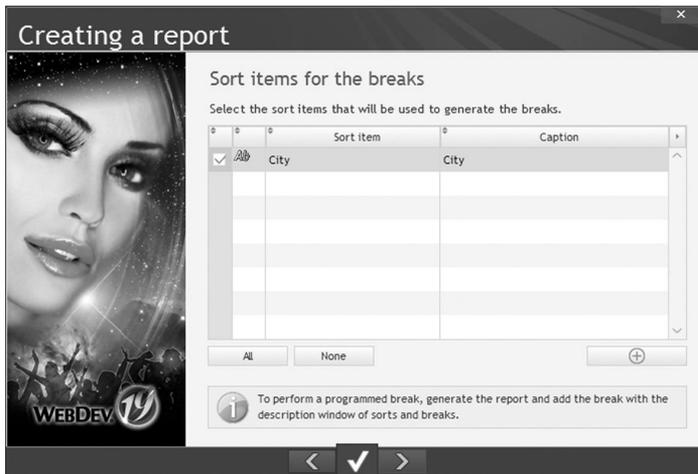
This break is **NECESSARILY** linked to a sort.

You will notice the presence of a break header and footer. The information common to the different rows is found in the break header.

The totals and the sums will be displayed in the break footer.

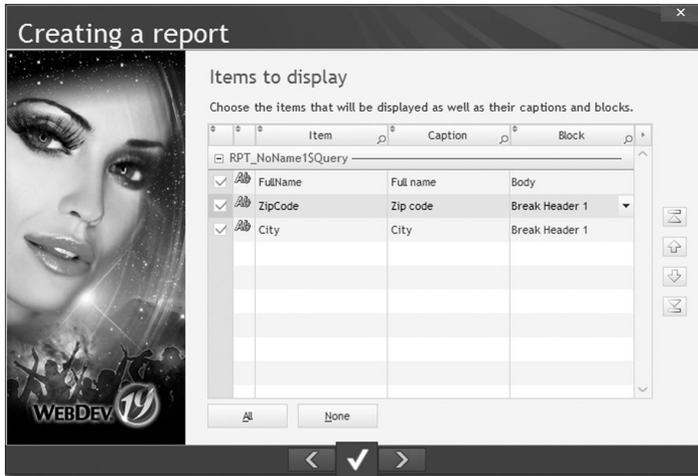
In our example, the break is performed on the city and it is used to group all the customers living in the same city.

- ▶ Answer "Yes" and go to the next screen.



- ▶ The break is performed on the city. Go to the next screen.
- ▶ You will now specify the order in which the items will be printed and how they will be distributed in the different blocks.  
In our example, the zip code and the city will be printed in the "Break header 1" block and the name in the Body block.

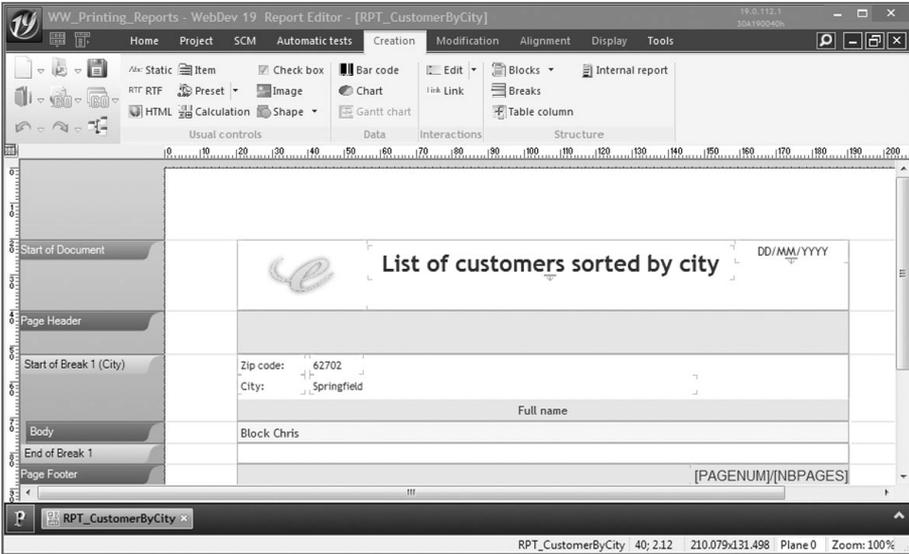
In the column on the right, select "Break header 1" for "ZipCode" and for "City".



The arrows on the right are used to modify the order of items. Go to the next screen.

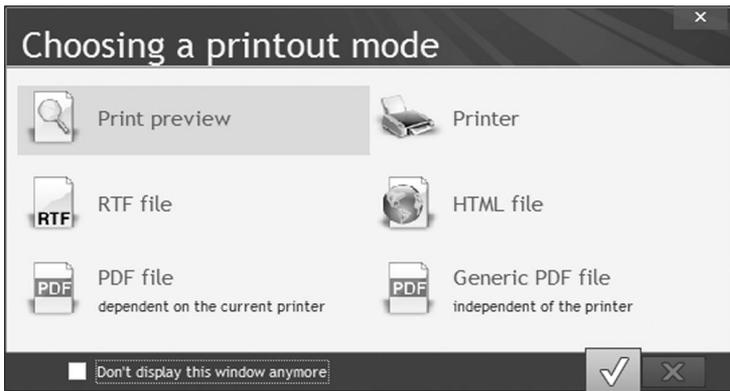
- ▶ This screen is used to define the layout of the report. We will keep the default values as well as the "Portrait" orientation.  
When choosing the print margins, don't forget to take into account the physical margins of the printer. The physical margins of the printer are margins in which no printing is allowed. Furthermore, the physical margins differ according to the type of printer. Go to the next screen.
- ▶ Select the "Elegant" skin template. Go to the next screen.
- ▶ Let's now give a name and a caption to the report and save it.
  1. Enter the name: "RPT\_CustomerByCity".
  2. Enter the title: "List of customers sorted by city".

3. Validate. The report is displayed in edit. Resize the title if necessary so that it is entirely displayed.

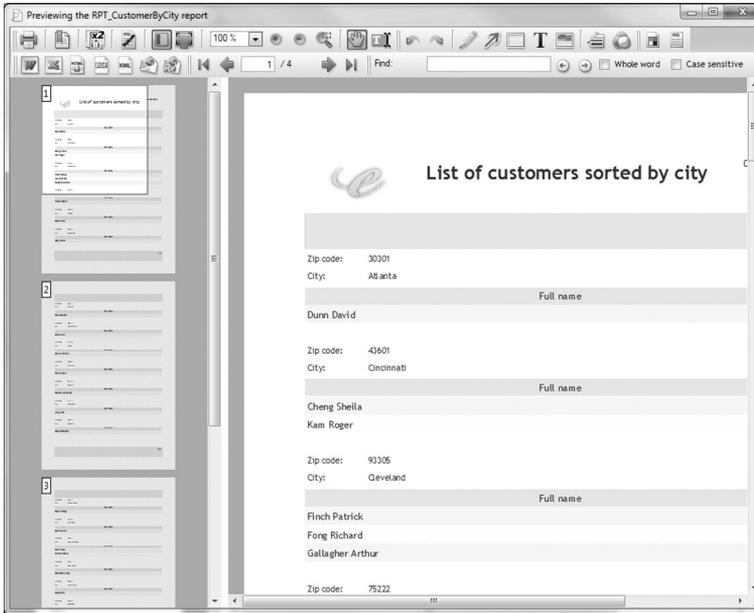


4. Save the report (  among the quick access buttons). The report is ended. Let's now run the test of the report.

- ▶ To run this report, click "GO" found among the quick access buttons. The print destination can be:

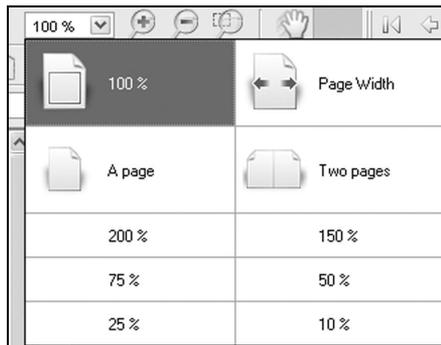


- ▶ Choose "Print preview" and validate.  
The report is displayed on the screen.



You can:

- print the current page or the entire document by clicking the printers.
- select a different zoom value:



Notes

In WebDev, the preview window can only be used in test mode from the editor when creating a report. The preview window will not be available when deploying your site or when running the test of your site.

## LESSON 7.3. STATISTICAL REPORTS

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

---

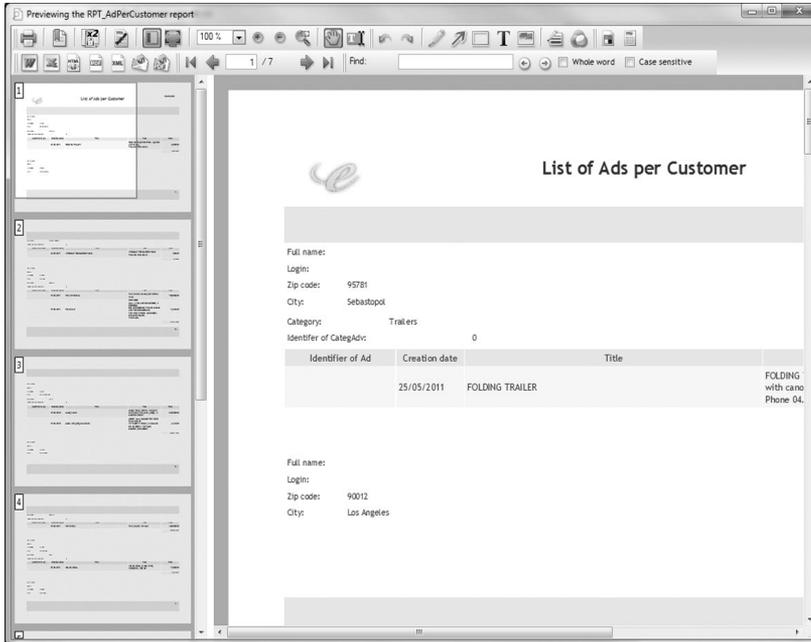
- Linking several data files in a report.
- Grouping elements in a report.



Estimated time: 30 min

## Overview

We are now going to edit the list of ads per customer (who placed an ad). This list will be sorted by customer name. We are going to print, for each customer, the ad and the details of the ad with a sub-total for each ad. The result should be similar to the representation below:



The report that will be created is using several data files as source.

**Reminder:** A sample project containing the necessary data was created beforehand.

- ▶ To open the example project:
  1. Close (if necessary) the current project to display the home window.
  2. In the home window, click "Tutorial" and select the "Printing reports (Exercise)" project.
 Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Printing reports (Exercise)".



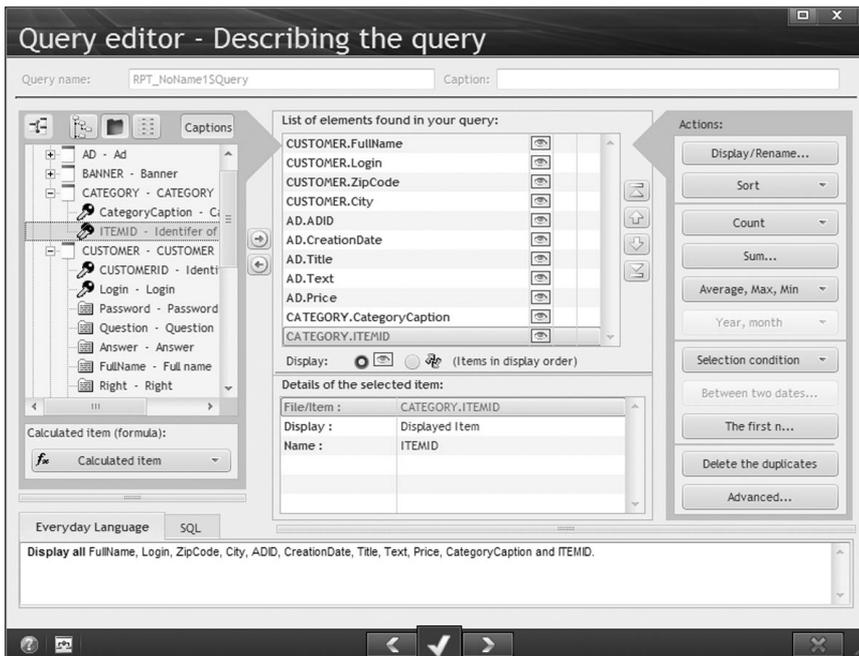
Notes

If you don't want to perform the operations of this lesson, a corrected project is available. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Printing reports (Answer)".

## Step by step

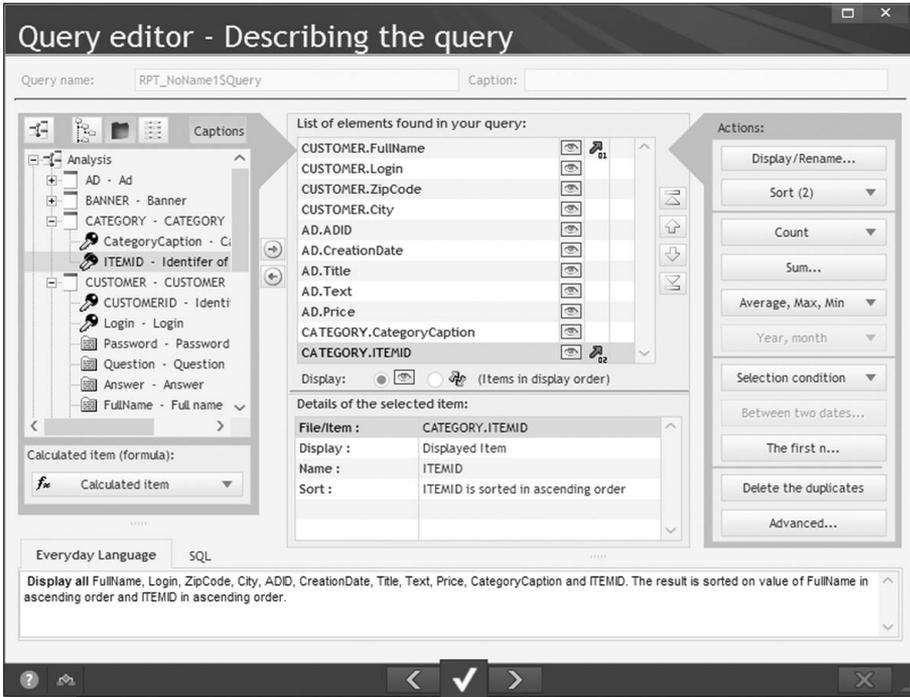
► To create a new report:

1. Click  among the quick access buttons.
2. Click "Report" in the wheel that is displayed.
3. Select "Table". Go to the next screen.
4. Select "From a new query". Go to the next screen. The query does not exist so let's create it. We are going to print the data coming from the CUSTOMER, AD and CATEGORY files.
5. Select the items that will be printed (via a double click):
  - The CUSTOMER file:
    - FullName
    - Login
    - ZipCode
    - City
  - The AD file:
    - ADID
    - CreationDate
    - Title
    - Text
    - Price
  - The CATEGORY file:
    - CategoryCaption
    - ITEMID



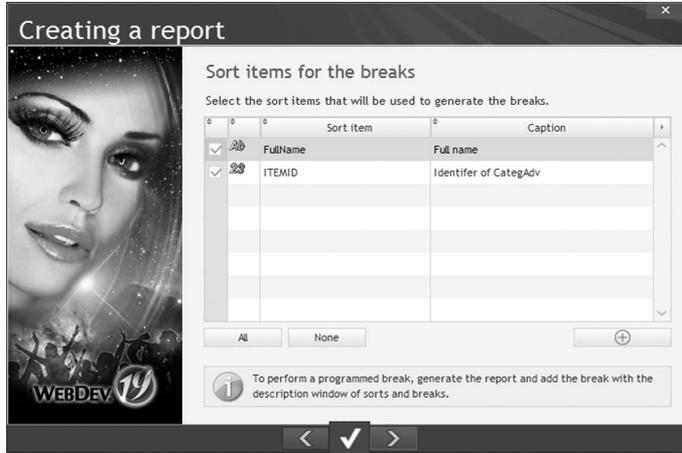
- ▶ The results will be sorted by customer name and by ad number. To do so, in the list in the middle:
  1. Click the "CUSTOMER.FullName" item. In the popup menu (right mouse click), select "Sort the item .. Sort in ascending order".
  2. Click the "CATEGORY.ITEMID" item. In the popup menu (right mouse click), select "Sort the item .. Sort in ascending order".

A red arrow must appear on the right of the "CUSTOMER.FullName" and "CATEGORY.ITEMID" items:



We are now going to validate this screen. Go to the next screen.

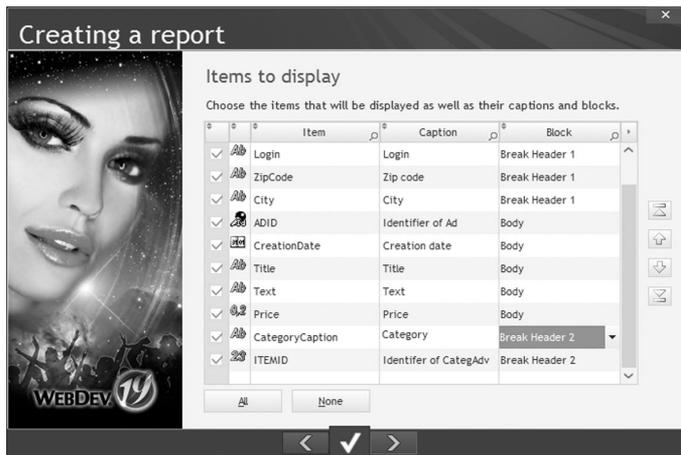
- ▶ The wizard for report creation proposes to create the breaks. Choose "Yes". Go to the next screen.  
By default, breaks are proposed on the "FullName" and "ITEMID" items. Indeed, these items correspond to the sort criteria that are already defined.



Let's confirm this choice. Go to the next screen.

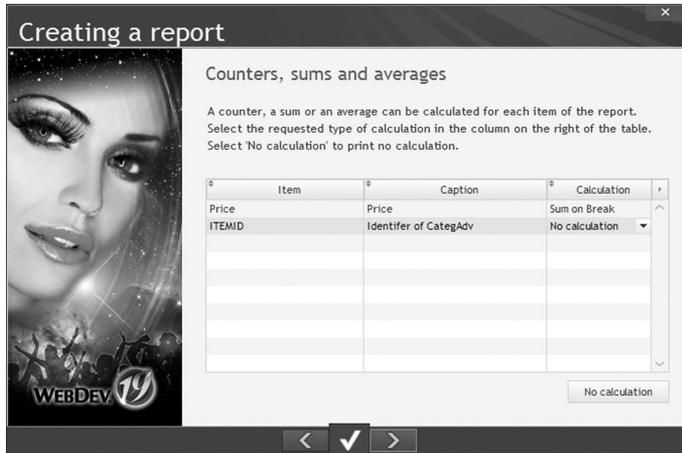
- ▶ We are now going to define the repartition of the items.
  - The information regarding the customer will be displayed in the break header 1: FullName, Login, ZipCode, City.
  - The information regarding the category of the ad will be printed in the break header 2: CategoryCaption, ITEMID.
  - The body block contains the details of each ad: ADID, CreationDate, Title, Text, Price.

The image below presents the repartition to perform:



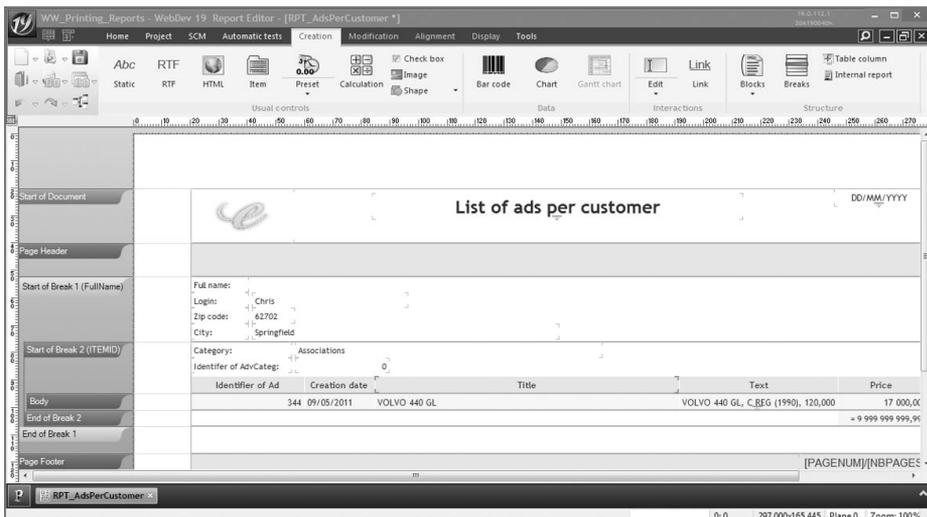
Go to the next screen.

- ▶ We must now specify whether totals or counts are found in the report. We are going to add the prices proposed by the customers for their ads. This sum will be included in a break.
  - We are going to keep the sum on the price.
  - We are going to delete the calculation on the identifier. Select the "ITEMID" line and in the "Calculation" column, select "No calculation".



Go to the next screen.

- ▶ Keep the parameters proposed by default for the page format. Go to the next screen.
- ▶ Choose a skin template. Go to the next screen.
- ▶ Give a name ("RPT\_AdsPerCustomer" for example) and a caption ("List of ads per customer" for example) to the report.
- ▶ Validate. The report being too big, select "Use the "landscape" mode" and validate the window. The report is displayed in the editor. Resize (if necessary) the title of the report, the title column and the text column.



► Save the report (  among the quick access buttons). Click "GO" to run the test.

The print preview displays the following report:

Previewing the RPT\_AdPerCustomer report

100%

1 / 7

Find: Whole word Case sensitive

### List of Ads per Customer

Full name:  
Login:  
Zip code: 95781  
City: Sebastopol  
Category: Trailers  
Identifier of CategAdv: 0

| Identifier of Ad | Creation date | Title           |
|------------------|---------------|-----------------|
|                  | 25/05/2011    | FOLDING TRAILER |

FOLDING with ca no Phone 04.

Full name:  
Login:  
Zip code: 90012  
City: Los Angeles

**Notes**

The preview window can only be used in test mode from the editor when creating a report. This option will not be available when deploying your site or when running the test of your site.

## LESSON 7.4. CROSTAB REPORTS

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

- Creating a crosstab report step by step.

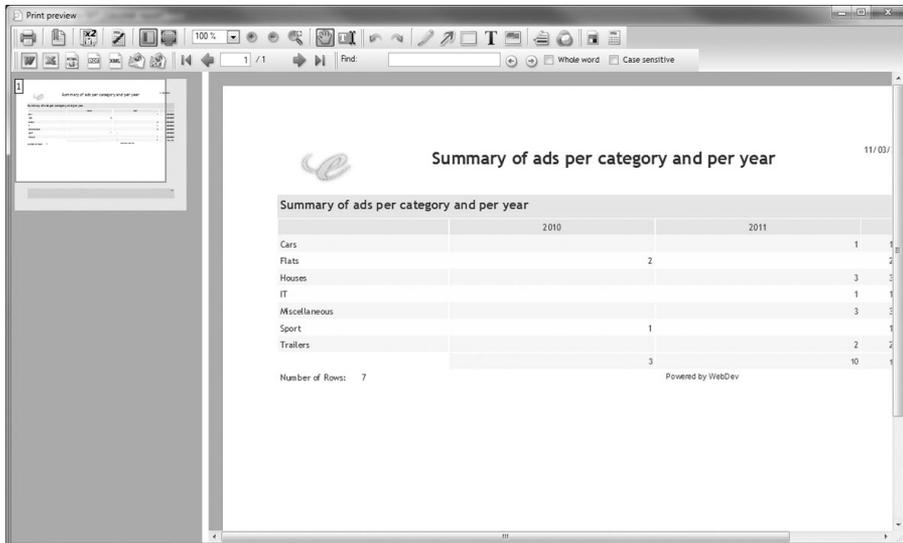


Estimated time: 20 min

## Overview

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

Example:



Unlike the other types of reports, the "Crosstab" report is always based on an **embedded query**. This query is created via the wizard for creating a "Crosstab" report. This query can be modified once your report is created.

**Reminder:** A sample project containing the necessary data was created beforehand.

► To open the example project:

1. Close (if necessary) the current project to display the home window.
  2. In the home window, click "Tutorial" and select the "Printing reports (Exercise)" project.
- Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Printing reports (Exercise)".



Notes

If you don't want to perform the operations of this lesson, a corrected project is available. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Printing reports (Answer)".

## Step by step

We are going to create a report used to list the number of ads per year and per category. The results will be sorted by year (horizontally) and by category (vertically).

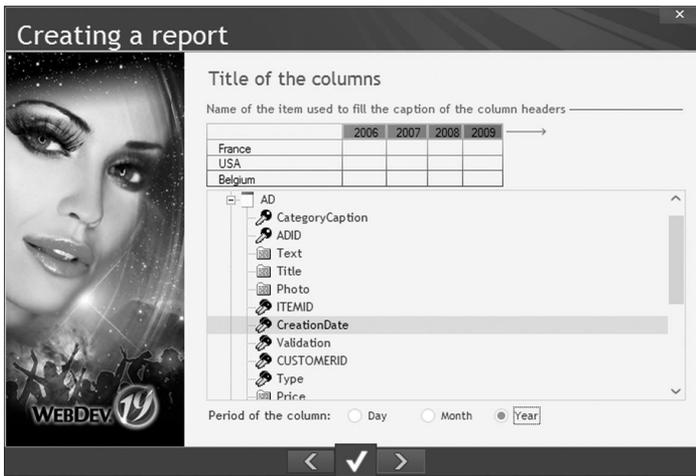
► To create a new report:

1. Click  among the quick access buttons.
2. Click "Report" in the wheel that is displayed.
3. Select "Crosstab". Go to the next screen.
4. The wizard proposes to choose the item that will be used as data source for the headers of columns.

Which means defining the first entry (row) of your crosstab.

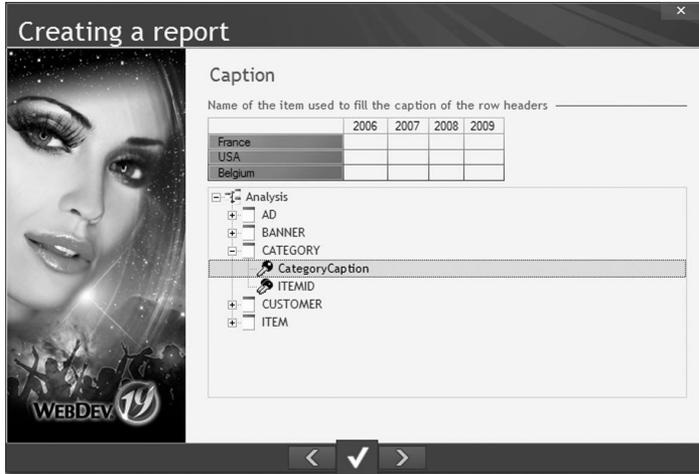
Select the "CreationDate" item found in the "AD" file.

The wizard proposes three choices (day, month and year):



5. Choose "Year". Go to the next screen.
6. The wizard asks you whether the dates must be "bounded". Don't check anything as all the ads will be selected. Go to the next screen.
7. The wizard asks you to choose the item that will be used as data source for the rows. This enables you to define the second entry (column) of your crosstab.

Select the "CategoryCaption" item found in the "CATEGORY" file.



Go to the next screen.

**8.** The wizard asks you to select the type of information that will be displayed in the cells of the crosstab.

As we want to display the number of ads per year and per category, we are going to ask for the number of records for each category and for each year.

Select "Number of records per Year and per CategoryCaption".



Go to the next screen.

**9.** The wizard proposes to calculate a sum per row and a sum per column. Accept this calculation. Go to the next screen.

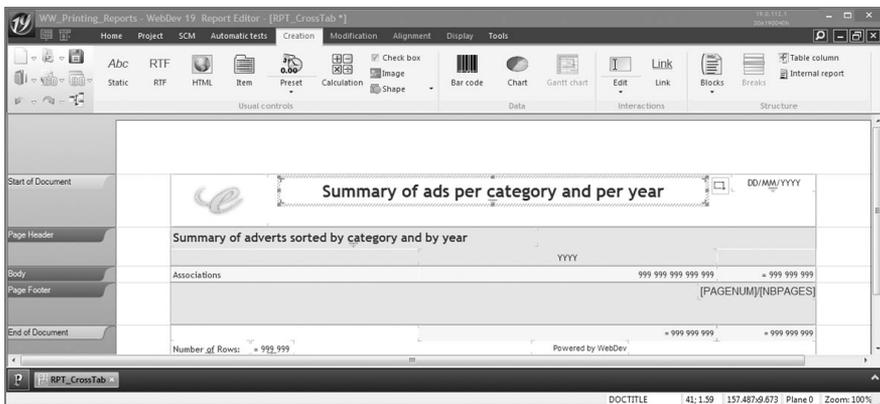
**10.** Don't change the page layout. Go to the next screen.

**11.** Select a skin template. Go to the next screen.

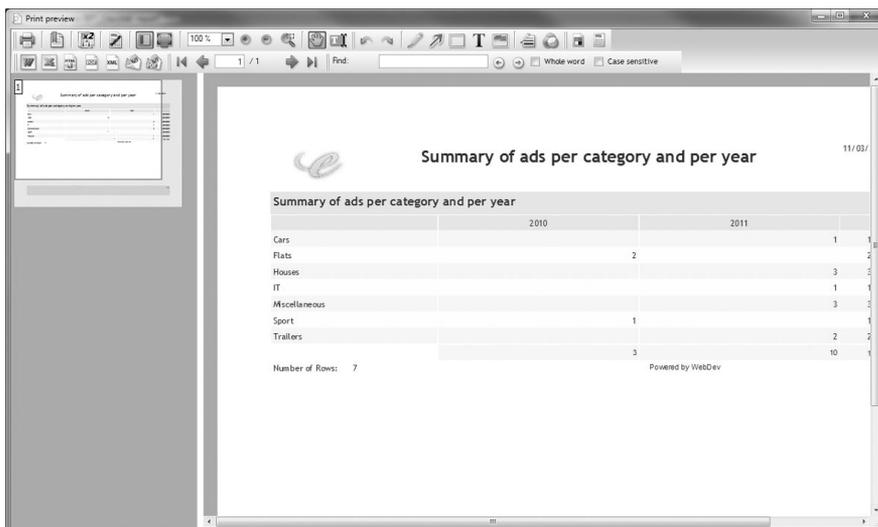
- ▶ Give a name to this report:
  1. Enter the name: "RPT\_Crosstab".
  2. Enter the title: "Summary of ads per category and per year".
  3. Validate.
  4. Select "Use the "Landscape" mode" and validate.
  5. The report is displayed in the report editor.

Resize if necessary:

- the title of the report
- the title of the page header.



- ▶ Save the report (  among the quick access buttons). Click "GO" to run the test. Caution: according to the data to print, the execution of a CrosTab report can take quite a long time.



You now know how to create a crosstab report. This type of report can be used for other purposes such as performing comparisons, calculating statistics, etc.

# LESSON 7.5. RUNNING REPORTS

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

---

- Running reports in WLanguage.



Estimated time: 10 min

## Running reports in WLanguage

The tests of the created reports have been run from the editor. We will now explain how to run a report from your site.



Notes

WebDev gives you the ability to run reports from a site and to program the report. You have the ability to add code lines to perform specific processes inside each report block (to perform calculations, to save a log, ...).

WebDev enables you to print your reports:

- on a server printer or on a printer accessible from the server. In most cases, this solution will not be used for Internet sites but for Intranet sites.



Caution!

The direct print on a server printer or on a printer accessible from the server **requires a specific configuration of the server** (rights, ...). See the online help (keyword: "Print") for more details.

- in a PDF, HTML or RTF file. This solution is the most commonly used (the generated file can be directly returned to the browser by *FileDisplay* for example).

The general syntax for running a report is as follows:

- Printing on the server:

```
// Direct print of Report_Table_POPULATION
iDestination(iPrinter)
iPrintReport(Report_Table_POPULATION)
```

- Printing in a file:

```
// Print the Report_Table_Population report
// in an HTML file
File is string = fDataDir()+ "\"+"POPREP_" + ...
                Today() + Now() + ".htm"
iDestination(iHTML, File)
iPrintReport(Report_Table_POPULATION)
FileDisplay(File, "text/html")
fDelete(File)
```

**Notes:** In this code:

- **fDataDir** is used to create the file corresponding to the report in the directory of HFSQL data files (this directory has write rights).
- **Today** and **Now** are used to create files with different names on the server. This allows you to identify the reports created by each Web user on the server.
- **fDelete** is used to delete the created file as soon as it is displayed.

### **iDestination function**

**iDestination** is used to select the output mode of the print. The syntax of **iDestination** is as follows:

```
iDestination (<Output Format>)
```

The <Output Format> parameter can take the following values:

- **iPrinter**: for a direct print on the current printer
- **iFile**: to print the report in a text file
- **iPDF**: to print the report in **PDF** format
- **iHTML**: to print the report in HTML format
- **iHTMLWithoutCSS**: to print the report in HTML format without style sheet
- **iRTF**: to print the report in RTF format
- **iXLS**: to print the report in an XLS file
- **iXML**: to print the report in an XML file
- **iFax**: for a direct print on a fax.

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

### **iPrintReport function**

**iPrintReport** enables you to specify the name of the report to run. The syntax of this function is as follows:

```
iPrintReport (<Report Name>)
```

Other syntaxes are possible for these two functions, see the online help (keywords: "Printing a report", "iDestination" and "iPrintReport") for more details.





**PART 8**

**Advanced  
programming**

**10**

**DEVELOP 10 TIMES FASTER**





# LESSON 8.1. WEB FEATURES

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

---

- The cookies.
- Upload and Download: transferring files.
- Securing the pages.
- Secure payment.
- Ajax.



Estimated time: 30 min

## Overview

---

This lesson presents several features specific to the WEB that can be included in a WebDev site:

- Cookie.
- Upload and Download.
- Securing pages.
- Secure payment.

### Practical example

For this lesson, we have created a project containing the data required to check the different features.

► To open the example project:

1. Close (if necessary) the current project to display the home window.
2. In the home window, click "Tutorial" and select the "WEB features" project.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "WEB features".

## The cookies

---

A cookie is an easy way to temporarily store an information on the computer of the Web user. This information can be retrieved later.

When the Web user visits the same site several times, the site "identifies" the user via the cookies. This allows you to avoid requesting details already provided during a previous visit and to propose custom pages.



Internet

A cookie has an expiration date (30 days after its creation by default). It is automatically destroyed by the browser of the Web user when its lifetime is exceeded.

**Caution:** the use of cookies is possible only if the browser of the Web users accepts the cookies.

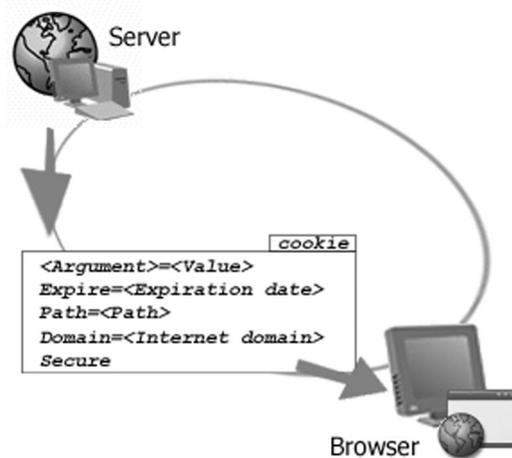
A cookie is used to store on the computer of the Web user various information such as the user name, the pages displayed by the user, the date of his last connection, the backup of his options, ...

This information, saved as cookies, will be read during the next connection to the site. Then, 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, ...

## What is a cookie made of?

A cookie is a text file stored on the computer of the Web user (in the Internet "cache" of the browser) during a specified duration. The cookie is created by the browser or by the server.



The following information is required to create a cookie:

- Name of the cookie, used by the site to identify the cookie
- Text of the cookie, corresponding to the information registered by the site: pages displayed, details supplied by the Web user, ...
- Expiration date after which the cookie is not valid anymore (it will be automatically deleted).
- Name of the Internet domain that created the cookie.



Notes

**Reminder:** the Internet domain is a specific section of the Internet address:  
[http://computer.\*\*domain.com\*\*/folder/page.html](http://computer.domain.com/folder/page.html)

## Managing the cookies with WebDev

Two WLanguage functions can be used to manage the cookies in your sites:

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

These functions can be used both in server code and in browser code:

- **in server code**, the cookie will be created on the computer of the Web user when displaying the next page. The cookie will be read during the next connection of the Web user.
- **in browser code**, the cookie will be directly created on the computer of the Web user during the call to **CookieWrite**. The cookie will be read immediately.

## Practical example

- ▶ To understand the operating mode of cookies:
  3. Run the test of the "WW\_WEB\_Features" project.
  4. Click the link named "The management of cookies". An information window is displayed, indicating that no cookie was found.
  5. Enter your name and click "Save".

During the next load of this page, the cookie will be sought and the name entered will be automatically displayed.
  6. Click the "Upload and download" link.
  7. Click the link named "The management of cookies" again. The cookie that was just created is automatically read and the corresponding controls are automatically filled.
- ▶ Go back to WebDev, open the "PAGE\_Cookies" page and take a look at the code of the button and page.
  - The load code of the page (browser code) reads the cookie via **CookieRead**.

```
sName is string
// Reads the "USERNAME" cookie
sName = CookieRead("USERNAME")

// The cookie is not empty
IF sName <> "" THEN
    EDT_UserName..Value = sName
    Info("The cookie was read")
ELSE
    Info("Cookie not found")
END
```

- The code of the "Save" button is used to create the cookie via **CookieWrite**.

```
CookieWrite("USERNAME", EDT_UserName, ...
            ListSelect (COMBO_ValidityDate))
Info("Information saved!")
```

## Uploading files

The upload consists in saving on the Web server a file accessible from the computer of the Web user. The file accessible from the computer of the Web user is "uploaded" to the server. It is loaded and saved on the server.



Internet

The upload is initiated by the user. You cannot force the upload of a file from the computer of the Web user to the server.

All types of files can be uploaded (images, video, archives, ...). WebDev allows you to propose the file upload:

- from an upload control (single-file or multi-file). This type of control uses the Flash technology.
- from an "Upload" edit control.



Internet

During an upload, you cannot specify the default directory in which the Web user will have to select a file. Furthermore, the default file type cannot be specified.

Some example for using the file upload:

- in an "Address Book" site, when a new contact is added into the address book: the user can select the photo of the contact on his computer. In order to be displayed regardless of the Web user, the image is uploaded onto the server.
- in a "Document management" site, each Web user can propose different types of documents: text, ... These documents are selected on the computer of the Web user and uploaded onto the server in order to be made available to all the Web users.
- ...

### Managing the upload with a single-file or multi-file upload control

To manage the file upload in your WebDev site, an Upload control (single-file or multi-file) must be created.

The necessary code is included in the control.

**Caution:** The code for receiving the uploaded files (code of the "Send" button) must be modified.

By default, the files are uploaded in a temporary directory of the WebDev Application Server. The reception code is used to handle each uploaded file. You have the ability to copy the file into a directory of the site, to rename it, to add it into a database, ...

## Practical example

- ▶ To understand the operating mode of the upload:
  1. In the "WW\_WEB\_Features" project, open the "PAGE\_UploadDownload" page.
  2. Run the test of the page.
  3. For a single-file upload, the "Send" link is used to select the file to upload.
  4. For a multi-file upload:
    - the Add button is used to select the files to upload.
    - the Send button is used to upload the files.
  5. Go back to the editor and take a look at the code of the buttons:
    - Send button for the single-file upload.
    - Add and Send buttons for the multi-file upload.

## The file download

---

The download is the reverse operation of the upload.

The download consists in saving on the computer of the Web user a file coming from the server. The file found on the server is downloaded to the computer of the Web user, then it is saved on the computer of the Web user.



Internet

The download is initiated by the server. You cannot force the destination directory of the file. A file picker is displayed on the computer of the Web user. Depending on the browser settings, the Web user will be able to choose the directory and the name of the downloaded file.

All types of files can be downloaded (images, video, archives).

**Note:** If the appropriate software is installed on the computer of the Web user, some files can be viewed in the browser directly (text or PDF files for example).

The file download can be used in the following cases for example:

- in a site for selling or renting DVDs. The trailer of a movie can be downloaded onto the computer of the Web user for example.
- in a "Document management" site. Each Web user can download different types of documents. These documents are selected on the server and downloaded onto the computer of the Web user.
- ...

## Managing the download

Several methods can be used to manage the file download:

- using a "Link" control of Other type, with the address of the file to download:



- using *FileDisplay* in server code.

## Practical example

► To understand the operating mode of download:

1. In the "WW\_WEB\_Features" project, open the "PAGE\_UploadDownload" page.
2. Run the test of the page.
3. In the "File download" section, click the different links used to perform a download.
4. Go back to the WebDev editor.
5. Take a look at the different links.
  - The first two links use *FileDisplay*. The download is defined by programming.
  - The last link is a custom link. The download is performed via an action defined in the description window of the link.

## Securing the pages

---

A site can propose specific pages to some of the Web users (back office, viewing the customer orders, ...).

In most cases, a password is sufficient to protect your pages. In some cases, to provide a better protection, the pages can be displayed via the **SSL** protocol.

### Protecting the access to your pages by a password

The access to a site or to a set of pages may be password-protected in the following cases:

- Protect personal data (but not sensitive data)
- Restrict the access to some pages (reserved to registered people or to the site administrator), ...

In most cases, an identification will be requested to the Web user (login and password).

For example, in a business site, an identified Web user will be able to see the history of his orders ...

### Practical example

- To see an example used to identify a Web user in a WebDev page:
1. In the "WW\_WEB\_Features" project, open the "PAGE\_Security" page.
  2. Run the test of this page.

This page is used to access a secure area intended for an administrator.

The following pages (if the identification was successful) will be protected by the TSL/SSL protocol (see "Securing the data and pages via TSL/SSL" later in this lesson).



Caution!

A TSL/SSL certificate must have been installed on your Web server.  
See the online help (keyword: "SSL") for more details.

3. Enter any user name and password, click "Enter" and see what happens.
4. Then, enter the "ADMIN" login and the "ADMIN" password (caution: the identification is case sensitive), and see what happens.

5. Go back to WebDev and take a look at the codes found in the page and in the "Enter" button.

```
// If the maximum number of attempts was exceeded
IF Attempts > 2 THEN
    STC_Attempts = "To access this area," + ...
    "you must identify yourself. No attempt left."
    Error("The number of attempts was exceeded.")
    // Makes the controls invisible
    IDENTIFICATION.Visible = False
    MySelf.Visible = False
    RETURN
END
// Checks the identity of the user
// The identity check could be performed
// via the data stored in a HFSQL data file
IF EDT_USERNAME <> "ADMIN" THEN
    // The user did not enter the proper user name,
    // increases the number of failed attempts
    Attempts += 1
    STC_Attempts = StringBuild(Text_Attempt, (4-Attempts))
    Error("Wrong user name or password.", "Please retry.")
ELSE
    IF EDT_PASSWORD <> "ADMIN" THEN
        // The user did not enter the correct password,
        // increases the number of failed attempts
        Attempts += 1
        STC_Attempts = StringBuild(Text_Attempt, (4-Attempts))
        Error("Wrong user name or password.", "Please retry.")
    ELSE
        // The user entered the proper user name
        // and the correct password
        PageDisplay(PAGE_SecureOK, EDT_USERNAME)
    END
END
END
```

This code is used to compare the input performed by the Web user with the login and password expected by the site.

If they match, the "secure" area is displayed in the browser of the Web user (**PageDisplay**). Otherwise, an error occurs and the Web user must perform his input again.

## Securing the data and pages via TSL/SSL

---

By default, the data exchanged between the computer of the Web user and the Web server is not protected. The confidentiality of data is not guaranteed.

Several systems can be used to ensure the confidentiality. A common system is 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/wd190awp/wd190awp.exe/connect/customers>).

### Implementing secure transactions via the TLS/SSL protocol

To implement secure transactions via the TSL/SSL protocol, you must comply with all the requirements linked to this protocol (purchase of a certified SSL key from a specific organization).

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

### Transactions secured by TLS\SSL in a WebDev site

Only the transfer of sensitive data must be secured: credit card numbers for example. Indeed, the secure mode is slower.

The secure mode is implemented when opening a new page via a button. All you have to do is use **SSLActive** in a browser code.

As soon as the secure page is opened, all the actions will be performed in secure mode (which means encrypted), regardless of the objects used (link, table, looper, clickable image, ...).

► To understand the use of **SSLActive**:

1. In the "WW\_WEB\_Features" project, open the "PAGE\_SecureOK" page.
2. Study the code of the "Enter" button.



Caution!

To run the test of the "PAGE\_SecureOK" page, an SSL certificate must be installed on your Web server.

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

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

## Secure payment

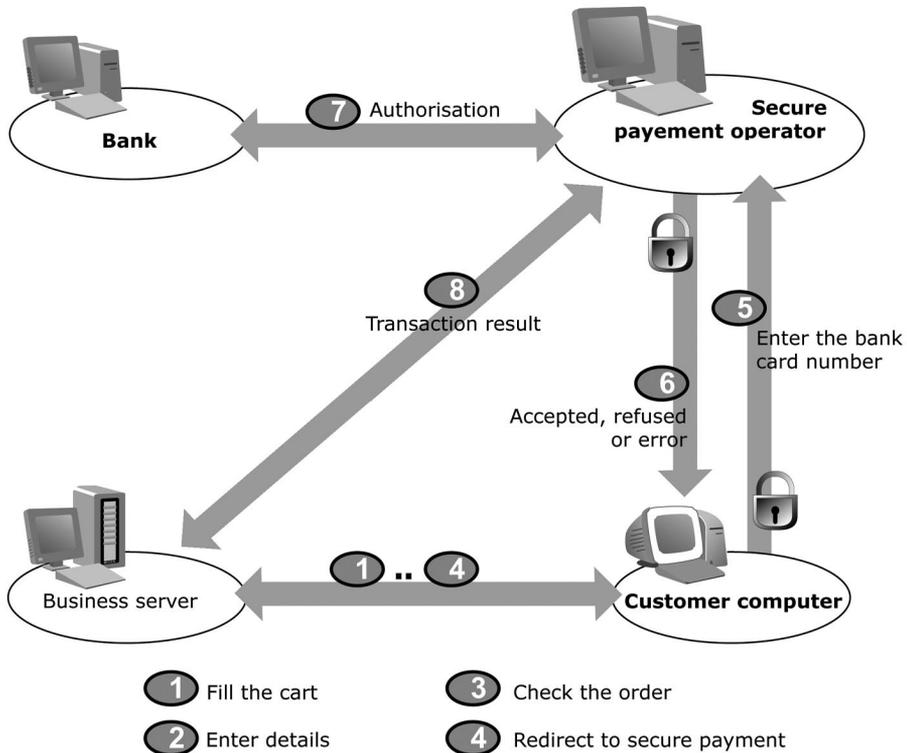
---

Most business sites allowing the Web users to perform online purchase use a system for online payment by credit card.

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

The data exchanged during this transaction must be secured (via the SSL protocol described in the previous lesson for example). A wide range of payment solutions is available (PayPal, Paybox, ...).

The principle for secure payment is as follows:



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

**5. to 7. Entering and checking the credit card number:** the Web user enters his credit card number in a page for secure payment. The transmission of data is protected via SSL to ensure the confidentiality of data. The business site is never aware of the credit card number of the Web user.

**8. Back to the business site:** the payment operator indicates to the business site whether the payment was validated, canceled or refused.



Notes

**Important:** the field of online payment evolves very quickly. Before implementing a payment solution, always check the latest solutions proposed by the different providers!

## System for secure payment in a WebDev site

### General procedure

In most cases, the following operations must be performed when implementing a solution for secure payment:

1. Request a development kit from the payment operator (PayBox, ATOS, SIPS, CyberMut, ...).
2. Contact the payment operator to establish a contract for remote sales. The bank supplies a vendor number.
3. Contact the payment operator to establish a contract specifying the parameters for remote sales.
4. Implementing the business site.

### Using the component supplied with WebDev: "PayBox component"

Several components for online payment are supplied with WebDev. They can be found in the "Components" directory of the setup directory of WebDev.

Each one of these components contains a use example.

The "PayBox component" component will be used in this tutorial. The component was already imported into our example. We won't go into details about including this component, this topic will be presented in "Internal components", page 382.

- ▶ To understand the operating mode of online payment via PayBox.
  1. In the "WW\_Web\_Features" project, open the "PAGE\_SecurePayment" page.
  2. Run the page test and perform the different operations.
  3. Go back to WebDev and study the code of the "Pay" button.

```
pPayBox . Payment ( EDT_AMOUNT , EDT_EMAIL )
```

The "PayBox Component" component establishes the connection to the payment operator.

We won't go into details about the different operations performed during an online payment (back to the WebDev page, cancelation, ...).

All these steps are explained in the "Component\_PayBox\_-\_Example" example available in the "Components\Component\_PayBox" sub-directory of the setup directory of WebDev.



Notes

The operating mode of WebDev with PayBox is given for information only. It is NOT a PC SOFT recommendation for this product.  
WebDev can be used with any provider of secure payment.

## Ajax

---

WebDev proposes the Ajax technology (WEB 2.0) for optimizing the display of Web pages.

### What are benefits of Ajax?

In traditional Web development, when the application wants to modify a single element in a displayed page (a price or a list of vehicles for example), the server must resend the entire page to the computer of the Web user.

This creates several side effects:

- it overloads the server because this one must send an entire page.
- it slows the server down because it must rebuild the page.
- it consumes bandwidth (the entire page content must be resent, which means several dozens or even hundreds of KB).
- it often generates a "refresh" effect on the browser of the Web user.
- the display can be slow down for the user because the browser must reinterpret and redisplay the entire page.

With the **AJAX technology**, essential to "WEB 2.0", "only" the modified data can be sent to the computer of the Web user, without refreshing the entire page.

There are several benefits:

- the server is less used, allowing to support an important number of connections on the same physical server.
- the information that circulates is less bulky.
- the transmission time is shorter.
- the display is immediate and without visual effect for the Web user.

### How to program AJAX in WEBDEV?

There's nothing special to program.

You automatically benefit from AJAX.

You continue to program in 5GL (WLanguage). WebDev takes care of everything!

An "AJAX" choice appears in the bar of the code editor. By default, the word "AJAX" is crossed out, which means that the process will be performed in standard WEB.



To enable AJAX on the process, all you have to do is click ... the "AJAX" button. The caption changes and becomes "AJAX enabled".



It is that simple!

### Practical example

For this section, the practical example is a training example that is supplied with WebDev.

- ▶ To open this example:
  1. Close the current project if necessary. The home window is displayed.
  2. In the home window, click "Open an example". The list of examples supplied with WebDev is displayed.
  3. In the search area, enter "Ajax". The training example named "WW\_Ajax\_Browse" is displayed.
  4. Double-click this example: the corresponding project is loaded.
- ▶ We are going to run the test of this example to see the difference between an Ajax process and a standard process.
  1. Run the project (  among the quick access buttons).
  2. Let's check the standard browse:
    - Use the arrows to browse the different products.
    - The page flickers whenever it is redisplayed. Indeed, the page is entirely refreshed.
  3. Now let's check the Ajax browse:
    - Use the arrows to browse the different products.
    - The products scroll smoothly. There is no noticeable flickering. Indeed, in Ajax, only the necessary page section is refreshed.

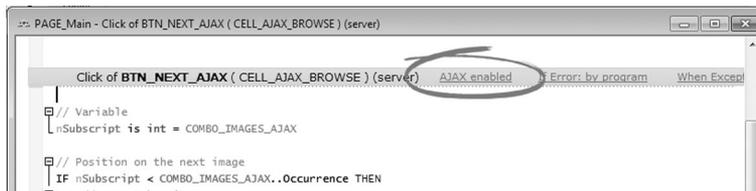
- Go back to the editor. If the following code is displayed in standard browse and in Ajax browse, the same code is used: .

```
// Variable
nSubscript is int = COMBO_TRADITIONAL_IMAGES

// Position on the next image
IF nSubscript < COMBO_TRADITIONAL_IMAGES..Occurrence THEN
  // Next subscript
  nSubscript++
  // Selects the image
  ListSelectPlus (COMBO_TRADITIONAL_IMAGES, nSubscript)
  // Refresh
  RefreshSelectedImage ()
END
```

The only changes are:

- the name of the combo box (makes sense you would say)
- and the "AJAX enabled" logo in the bar for click process:



One click and at the end, a smooth site.

## LESSON 8.2. AUTOMATIC MANAGEMENT OF ERRORS

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

---

- What is the automatic management of errors?
- Using the automatic management of errors



Estimated time: 10 min

## Overview

WebDev can manage the errors automatically. This feature helps you reduce the number of code lines while centralizing the management of errors.

The use of this feature also makes the code easier to read.

### Operating mode

Two operations are performed when an error is detected by a WLanguage function:

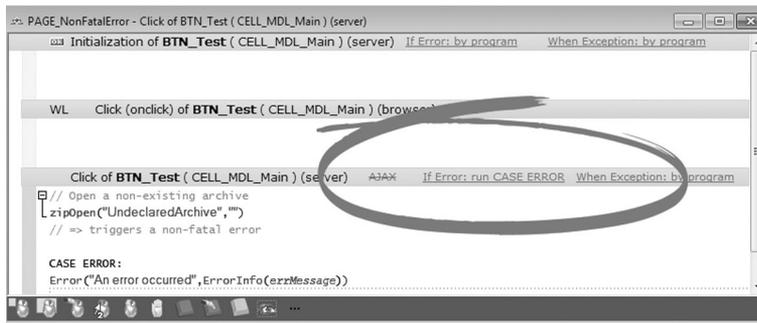
- a return value of error is returned by the function (**fOpen** returns "-1" if the specified file cannot be opened for example).
- the error is detected by WLanguage (the **ErrorDetected** variable corresponds to **True**); the details of this error can be retrieved by **ErrorInfo**.

This second operation can be automatically managed via the error management of WebDev.

### Implementation

The automatic management of errors can be configured:

- in the code editor: all you have to do is click the "If error: by program" link in the code editor:



- by programming with **ErrorChangeParameter**.

### Types of 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 project. 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 hardware problem (not enough disk space, loss of network connection, ...) or to important development problems (access to a non-declared file, use of non-existing controls, ...). A "fatal" error can also occur after a "non-fatal" error that was not processed properly. In this case, the project will be stopped.

The mechanism for managing the errors is used to manage these two types of error according to different methods so that you can specify behaviors adapted to the errors that occur.

## Using the automatic management of errors

---

- ▶ To open the example project:
  1. Close (if necessary) the current project to display the home window.
  2. In the home window, click "Tutorial" and select the "Automatic management of errors" project.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Automatic management of errors".
  
- ▶ This project presents the management:
  - of a **non-fatal error** (opening an archive that does not exist).

When running the code line that triggers the error, an error message is displayed to the Web user; it proposes to retry the operation, cancel the operation or stop the application.
  - of a **fatal error** (division by 0).

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.
  - of an **error on several levels** (process calling a procedure that opens an archive that does not exist).

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.
  
- ▶ Run the project test by clicking  among the quick access buttons.

## LESSON 8.3. MANAGING THE EMAILS

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

---

- Protocols for managing the emails
- The *EmailXXX* functions of WebDev.



Estimated time: 20 min

## Overview

---

Several WLanguage functions can be used to manage the incoming and outgoing emails in server code. Furthermore, WebDev allows you to find all the characteristics of an email:

- sender, recipients
- outgoing date, subject, message
- attachments ...

**Note:** the parameters for sending an email in browser code can also be configured by *EmailOpenMail*. See the online help (keyword: "EmailOpenMail") for more details.

WebDev proposes several methods for managing the emails:

- The **POP3 and SMTP** protocols (most frequently used method). These protocols are protocols for email management recognized by all the service providers. These protocols allow you to directly communicate with the server, available at your ISP or on your Web server.
- The **IMAP protocol**: this protocol for receiving emails allows you to leave the emails on the server so that they can be consulted from different messaging clients.
- The **"Simple Mail API"** (also called SMAPI or Simple MAPI): this mode for email management is used by most of the Microsoft applications, especially by Microsoft Exchange.

In this lesson, we will only present the management method via the POP3/SMTP protocols. Indeed, this mode is commonly used all over the world.

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

The functions for email management start with "Email".

## The POP3 and SMTP protocols: Principle

---

The POP3 protocol is used to receive emails.

The SMTP protocol is used to send emails

We won't go into details about the operating mode of these protocols.

To send or read messages via the POP3/SMTP protocols, you must:

1. Start an email session with *EmailStartSession*.
2. Send and read the messages.
3. Close the messaging session with *EmailCloseSession*.

To send and read the messages, WebDev proposes two methods for managing the messages:

- the Email structure. This structure is a preset structure of WLanguage.
- via an Email variable. This type allows you to handle several messages at the same time.

## Practical example

The "WW\_Advanced\_Programming.WWP" project contains a page used to send emails (most common case).

- ▶ Open the "WW\_Advanced\_Programming.WWP" project if necessary.
  1. Close the current project if necessary to display the home window.
  2. In the home window, click "Tutorial" and select the "Advanced Programming" project.  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Advanced programming".
- ▶ Open the "PAGE\_Email" page.
- ▶ Run the test of this page.
  1. Enter the characteristics of the email: SMTP server, sender address, text of the email, recipient address, ...
  2. Click the "Send" button according to the selected method:
    - "Send method 1" uses the Email structure.
    - "Send method 2" uses the Email variable.
  3. Check your favorite messaging software to find out whether message was sent.

## Sending an email: the different steps

---

### Starting an SMTP messaging session

The message is sent via the SMTP protocol. To start a messaging session, you can:

- use **EmailStartSMTPSession** and specify the address of the SMTP server to use. In our example, this function is used in the code of the "Send method 1" button. The following code is used:

```
IF NOT EmailStartSMTPSession("S1", EDT_SMTPServer) THEN
    Info("Unable to connect to the SMTP server")
    RETURN
END
```

- use **EmailStartSession** associated with an **EmailSMTPSession** variable. In our example, this function is used in the code of the "Send method 2" button. The following code is used:

```
MySession is EmailSMTPSession
MySession..ServerAddress = EDT_SMTPServer
IF NOT EmailStartSession(MySession) THEN
    Info("Unable to connect to the SMTP server")
    RETURN
END
```

## Preparing the email

To send a message with WebDev, all you have to do is specify the main information of an email. This information is found:

- in the preset Email structure of WLanguage. In our example, the "Send method 1" button is using this structure. The following code is used to fill the structure:

```
EmailReset ()
Email.Sender = EDT_Sender
Email.Subject = EDT_SUBJECT
Email.Message = HTMLToText (EDT_BODY)
Email.HTML = EDT_BODY
Email.NbRecipient = 1
Email.Recipient[1] = EDT_Recipient_Addr
IF AttachmentName<>" THEN
    Email.NbAttach = 1
    Email.Attach[1] = AttachmentName
END
```

- in the advanced Email variable. In our example, the "Send method 2" button is using this variable. The following code is used:

```
MyMessage is Email
EmailReset (MyMessage)
MyMessage..Sender = EDT_Sender
MyMessage..Subject = EDT_SUBJECT
MyMessage..Message = HTMLToText (EDT_BODY)
MyMessage..HTML = EDT_BODY
Add (MyMessage..Recipient, EDT_Recipient_Addr)
IF AttachmentName<>" THEN
    EmailLoadAttachment (MyMessage, AttachmentName)
END
```

Note: This example can be used to send an attached file. You have the ability to use several attached files. See the online help (keyword: "Sending an email") for more details.

## Sending the email

To send the email, all you have to do is use *EmailSendMessage*.

## Disconnecting

To disconnect, all you have to do is use *EmailCloseSession*.

## Other possibilities

---

It also allows you to:

- send attached files in your emails.
- perform mailshots. We recommend that you perform the mailshot operations via a WinDev application in back office (see "WinDev Back Office", page 446).

See the online help (keyword: "Email, Managing the emails") for more details.

# LESSON 8.4. OOP

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

---

- Concepts of Object-Oriented Programming.
- Examples of object declaration.



Estimated time: 30 min

## Concepts

---

The Object-Oriented Programming (OOP) is designed for better code reusability. The programs developed in OOP are structured: they include modules, each one of these modules being used to manage a feature of the program. These modules can easily be re-used in other programs. They contain a set of procedures and they encapsulate the data structure on which the procedures will act.

To use the "object-oriented" programming, you must declare the classes, the members and the associated methods.

### The classes

A **class** contains the description of a data structure (the members) and the procedures (methods) used to handle the members.

Therefore, a class defines a type of data and its behavior.

### The objects

A class is used to create **objects**. Each created object owns the members described in its class and it can be handled via the methods of its class. An object is defined as being an instance of the class.

When the class is declared, all you have to do is associate an object with a class in order for the object to be handled by all the methods of this class.

### The members

A **member** is a data (or parameter) of the object.

### The methods

A **method** is used to act on the object, to modify its members for example. A method is a procedure. Its operating mode is similar to the one of the standard procedures of WLanguage.

### Concept of inheritance

The **inheritance** is used to include the characteristics of an existing class (*base class*) in a new class (*derived class*). The inheritance allows you to create a new type of data from a known type in order to add features to it or to modify its behavior. Therefore, the base class will not be modified. A class can inherit from a class: it becomes a sub-class of this class.

A *derived class* inherits from the members and methods of its *parent class* (that can, itself, be a sub-class of another parent class), in addition to its own members and methods (and also from the members and methods of the first parent class if it is a multiple inheritance). There is no need to duplicate the members and methods of the parent classes.

### Constructor and Destructor

The notion of **Constructor** and **Destructor** is important because it allows for an automatic call to initialization methods when creating an object or when destroying it.

The Constructor method associated with a class is automatically called when declaring an object of the class.

The Destructor method associated with a class is automatically called when deleting the object (exit from the procedure where the object was declared).

## Data encapsulation

The data encapsulation is used to make sure that the data belonging to the object is not accidentally modified by functions (methods) external to the object.

This enables you to prevent the user of an object from accessing some or all of its members. The members whose access is not allowed are called private members.

They can only be accessed from the methods designed for this purpose in the class.

## Creating an object-oriented program

To create an object-oriented program in WLanguage, you must:

1. Describe the class and the members of the class.
2. Specify all the methods of the class.
3. Declare the objects by associating them with a class ("instantiate a class").

For the object-oriented programming, a project is required to handle the classes.

- ▶ Open the "WW\_Advanced\_Programming.WWP" project if necessary.
  1. Close the current project if necessary to display the home window.
  2. In the home window, click "Tutorial" and select the "Advanced Programming" project.  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Advanced programming".
- ▶ Open the "PAGE\_OOP" page and run its test.

We won't go into details about the syntax of O.O.P. We will present a simple example of object-oriented programming. See the online help and the programming guide of WebDev (keyword: "OOP, Class") for more details.

## Declaring a class

To create a class:

1. In the project explorer, select the "Classes" folder.
  2. Display the popup menu of this folder (right mouse click) and select "New class".
  3. In the window that is displayed, enter the name of the class and validate.
  4. The code of the class can be entered in the code editor.
- ▶ Study the code of the CFile class used in the example.
    1. In the project explorer, select the "Classes" folder.
    2. Open the "Classes" folder (to do so, click the arrow found in front of the folder name).
    3. Double-click the CFile class.
    4. The code of the class is displayed in the code editor. The declaration code of the class is as follows:

```
CFile is Class
  m_sName is string
  m_sExtension is string
  m_sFilePathOnServer is string
  m_sFilePathOnClient is string
END
```

"CFile" is the name of the class.

"m\_sName", "m\_sExtension"... are the members of the class.

## Describing the methods

To create a method:

1. Right-click your class found in the project explorer.
2. Choose "New method" from the popup menu.
3. In the window that is displayed, enter the name of the method and validate.
4. Enter the code of the method in the code editor.

► To display the FileDate method of the CFile class:

1. Click the arrow found on the left of the class name in the project explorer to display the methods of the class.
2. Double-click the name of the "FileDate" method.
3. The following code is displayed in the editor:

```
PROCEDURE FileDate()  
RESULT fDate(:m_sFilePathOnServer)
```

## Declaring and handling the objects

In the processes of the page, an object is declared at the same time as the other variables.

```
GLOBAL  
File1 is object CFile
```

To refer to a member of the "clMyFile" object, use the following syntax

```
<ObjectName>.<name of member>
```

► Display the code of the "Select" button (right-click the button and select "Code" from the popup menu). The object is handled as follows:

```
clMyFile is CFile(UPL_Upload)  
  
FILEINFO..Visible = True  
FILEINFO..State = DisplayOnly  
  
EDT_Name = clMyFile:m_sName  
EDT_Extension = clMyFile:m_sExtension  
EDT_ClientPath = clMyFile:m_sFilePathOnClient  
EDT_ServerPath = clMyFile:m_sFilePathOnServer  
  
EDT_Size = clMyFile:FileSize()  
EDT_Date = clMyFile:FileDate()  
EDT_Time = clMyFile:FileTime()
```

We won't go into details about OOP in this tutorial.



**PART 9**

**Advanced project  
management**

**DEVELOP 10 TIMES FASTER**





# LESSON 9.1. DASHBOARD

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

---

- Overview
- View mode
- Dashboard options



Estimated time: 20 min

## Overview

The project dashboard is an essential element for managing the WebDev projects. The project dashboard gives an overall view of the progress status of a project.

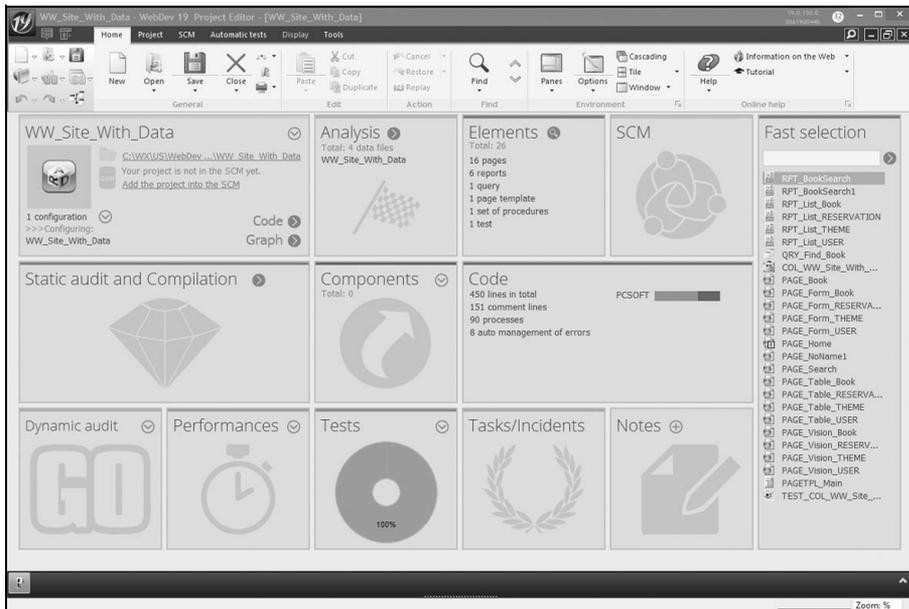
Some features of the dashboard were already presented in part 2.

We are now going to study it in details and see how it can interact with the Control Centers.

### Example

To handle the dashboard, we will be using the project named "WW\_Site\_With\_Data". If this application was not created beforehand, a corrected version is available: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Site with data (Answer)".

## The different dashboard elements



The project dashboard includes several Widgets. Each Widget displays various information about the project.

For example, the "Performances" Widget is used to start the performance profiler or to open the last performance report.

The "Static audit and Compilation" Widget is used to quickly see if the project contains compilation errors.

You have the ability to enable (or not) a Widget, to add one or to delete one at any time.

Let's present some Widgets found in the dashboard.

## "Project" Widget

This Widget is used to quickly view the characteristics of the project:

- Project found or not the SCM (Source Code Manager). We will see how to use it in another lesson.
- Number of configurations found for this project. The arrow button allows you to select a configuration or to create a new one.

A link is used to open the project directory.

The "Code" option is used to access the code of the project.

"Graph" option is used to display the graph of the project.

You can go back to the dashboard at any time by using  (found on the "Project" pane, in the "Project" group).

## "SCM" Widget

This Widget is used to find out the status of the project found in the SCM.

## "Static audit and compilation" Widget

This Widget is used to quickly view the number of compilation errors found in the project and to start the static audit of the project.

The static audit is used to get a status of your project. It allows you to:

- **detect the dead code**, which means list the unused procedures of your site.
- **detect the orphan elements**, which means lists the unused project elements (page used for test when developing the site for example).
- **detect whether an optimization of queries is required**: the static audit informs you if your project is using queries that can be optimized by a simple modification of the analysis (adding a composite key for example).

## "Runtime" Widget

This Widget is used to start a dynamic audit of the project. The dynamic audit is used to perform an audit during the test or the execution of the site. You can find out:

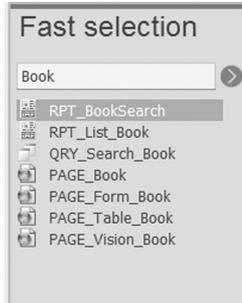
- the memory footprint of the different elements used by the site.
- the images not found
- ...

## "Performances" Widget

This Widget is used to start the performance profiler (that will be presented in another lesson). The performance profiler is used to check and optimize the execution time of your application.

## "Fast selection" Widget

This Widget is used to easily find a project element. All you have to do is type some letters found in the name of the sought element. The selection is automatically performed.



To find the PAGE\_Book page: type "Book" and make your choice. A double click on the name of the element allows you to open it in the associated editor.



Tip

You want to find an element without being positioned in the dashboard? All you have to do is press [CTRL] [E] ... and the same features will become available.

## The Widgets in relation with the Control Centers

WebDev is supplied with several Control Centers. The Control Centers are used to control a set of features. You are already familiar with the HFSQL Control Center, that is mainly used to control the data files found in the HFSQL Client/Server sites.

The Project Monitoring Center is also available in WebDev. This center is used to manage the projects, from design to distribution and even maintenance.

Therefore, you have the ability to include in the dashboard a Widget used to find out:

- the number of tasks currently in progress on the project. These tasks are referenced in the Project Monitoring Center. These tasks correspond to:
  - Features that must be added to the project
  - Forthcoming development, ...
- the number of requests currently found on the project. These requests are referenced in the Quality Control Center.

## LESSON 9.2. SCM

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

---

- Overview
- The Source Code Manager
- Using the Source Code Manager



Estimated time: 30 min

## Introduction

The development of a large IS system requires the participation of several developers. These developers must work on a single WebDev project while sharing the resources (pages, classes, ...).

WebDev proposes a Source Code Manager ("SCM") used to share the source codes of different projects between developers and to find out the full history of the modifications performed (in the code, in the interface, ...).

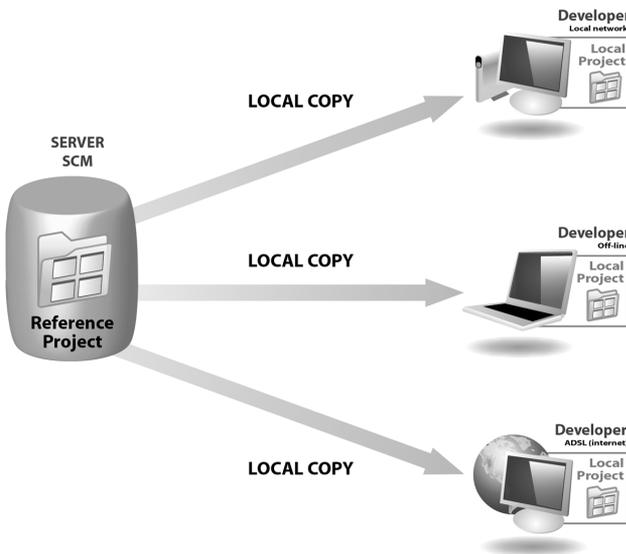
## SCM (Source Code Manager)

### Principle of SCM

The Source Code Manager is used to store and share the projects and their elements.

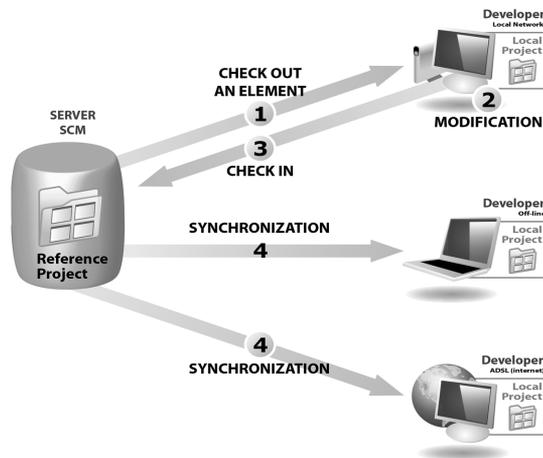
The principle is as follows:

- A reference version of each one of your projects is found on a server. All these versions are called "SCM database".
- Each developer has a local copy of the different projects on his computer.



- Whenever a developer wants to modify a project element (page, report, query, ...), he notifies the SCM that he is becoming the temporary owner of this element. To do so, this element will be checked out from the SCM database by the developer.
- This developer gets exclusive rights on this element: all the requested modifications can be performed on this element.
- The other developers are still working on the copy of the reference version of this element (found in the SCM database).
- Once the modifications have been made by the developer, the checked-out element is checked back into the SCM database.

- The other developers are automatically notified of this check-in operation. They can now update their local copy.



The SCM supports teamwork and it allows you to find out the history of all the modifications. The SCM can also be used to manage and control the elements shared between several projects.

### Creating the SCM database

To share a project via the Source Code Manager, a SCM database must be created. This SCM database must be created **once only** on a server.

This SCM database can be created:

- when installing WebDev.
- when creating a project that uses the SCM.
- when importing a project into the SCM.
- in the SCM administrator directly.

- ▶ Our SCM database will be created when a project is imported into the SCM (next step).



Notes

We advise you to perform backups of the SCM database on a regular basis. To do so, connect as administrator to the tool for SCM management and select "Tools .. Management .. Full database backup".

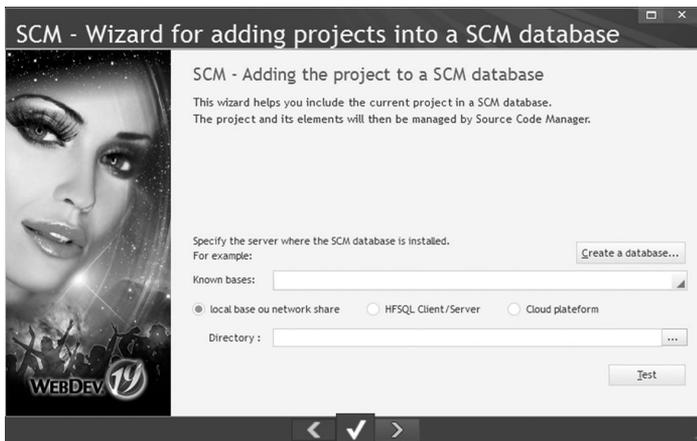
## Including a project in the SCM

To use an existing project with the Source Code Manager, all you have to do is include this project in the SCM database.

- ▶ We are now going to include the "WW\_Pages\_and\_controls.WWP" project in the SCM database:
  1. Close the current project if necessary. The home window is displayed.
  2. In the home window, click "Tutorial" and select the "Pages and controls" project. The project is loaded.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Pages and controls". The project is loaded.

3. On the "SCM" pane, in the "Project" group, click "Add the project". The wizard for adding projects into the SCM starts:



The SCM database was not created yet. We are going to create one.

Note: We are going to create a "local" SCM database (on the development computer). The operating mode would be similar for a network SCM database.

4. Click the "Create a database" button. The SCM database can be in HFSQL Classic format (local or network) or in HFSQL Client/Server format.

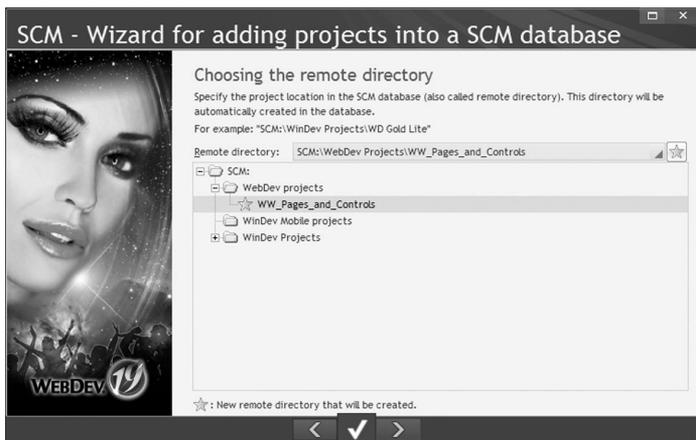
We are going to create a SCM database in HFSQL Classic format.



Notes

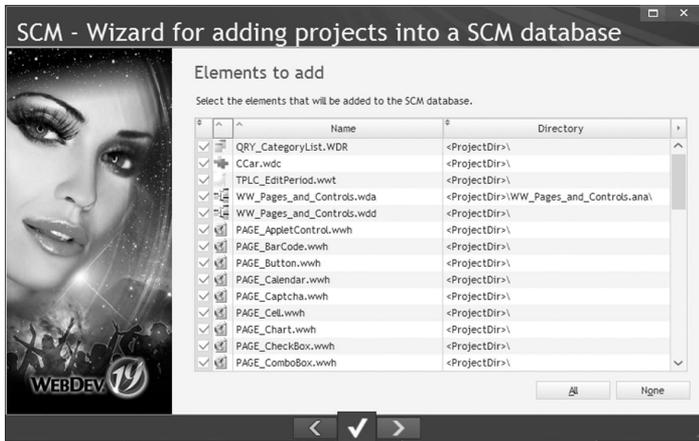
If the SCM database is in HFSQL Client/Server format, this SCM database can be used remotely.

5. Keep the "Creating a database in network share mode" option selected and specify the directory of this SCM database ("C:\My Projects\My SCM database" for example). Validate the creation of the SCM database ("Create the database" button). The SCM database is now created. We are going to include our project in this SCM database.
6. Go to the next screen.
7. The wizard proposes to place the project in the "WebDev projects" sub-directory of the SCM database.



Accept this location. Go to the next screen.

8. The wizard asks you to select the project elements that must be added into the SCM database.



We want to add all the project elements. Go to the next screen.

9. The wizard asks you to select the project dependencies that must be added into the SCM database. These dependencies correspond to all the external elements required by the project (images, style sheets, ...).

We want to add all the project dependencies. Go to the next screen.

10. Validate the inclusion of the project in the SCM. The project and all its elements have been added into our SCM database.



Notes

#### Sharing the project elements

When the projects that share the same resources (same analysis, same windows, ...) are included in the SCM, the relevant elements can be shared between the different projects. Therefore, the same element is checked in once only into the SCM.

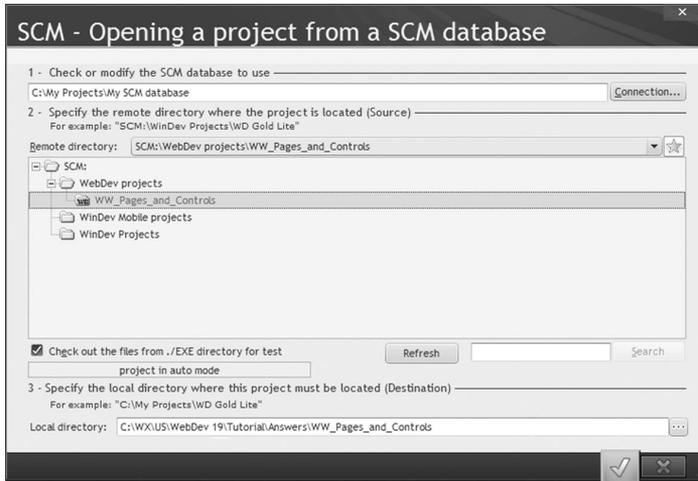
## Opening a project from the SCM

In our example, the project is included in the SCM and it can be used directly.

**In a real case, in order for other developers to work on a project found in the Source Code Manager, they must retrieve a copy of this project locally.**

To do so, the following operations must be performed:

1. Open your project from the Source Code Manager: on the "Home" pane, in the "General" group, expand "Open" and select "Open a project from the SCM".
2. In the next screen, specify the local directory if necessary:



**Note:** If the project was already opened from the SCM, the SCM proposes to open the project as usual or to overwrite the content (to retrieve the entire project).

This operation must be performed once only by each developer who is using the project.

The developer who added the project into the Source Code Manager (you in this case!) has no operation to perform.



Notes

The next openings of a project managed by the SCM are identical to the openings of a project not managed by the SCM: all you have to do is open the project (".WWP" file) corresponding to the local copy.

## Configuring the SCM

The SCM must be configured before you start working on the project elements found in the SCM.

The important points of this configuration affect:

- The check-out mode of the project elements.
- The check-out mode of the project (WWP file).

### Check-out mode of the project elements

When working on the elements of a project found in the SCM, the element must be checked out from the SCM database before it can be modified, then it must be checked back in once the modifications have been performed. Therefore, the modified element becomes available to all the SCM users.

**WebDev proposes two modes for checking out the project elements:**

- **the standard mode:** if you display a SCM element that is not checked out, a panel indicates that this element must be checked out before it can be modified. The element can be checked out immediately (check-out button found in the panel).
- **the automatic mode:** if you try to modify an element that is not checked out, the SCM automatically proposes to check out this element. Once the check-out is validated, the element can be modified.

Note: this mode is not recommended when using the SCM with a slow Internet connection.

**In this tutorial, we will be using the standard check-out. This check-out mode is enabled by default in the SCM.**

**Check-out mode of the project**

The Project file (WWP file) contains the different options used to configure the project (initialization code of the project, list of linked elements, name of the first project page, ...).

WebDev proposes two modes for managing the project:

- **Master/Guest mode:** Only the master can modify the project and apply these modifications to the SCM database. The master can also check back in all the elements to create the executable and the setup program. The modifications made to the project by the guests will not be taken into account by the SCM database.



Notes

The Master/Guest mode is recommended when the SCM is used by a single developer.

- **Automatic mode:** The project file is checked out only if the action performed requires this check-out (regardless of the user). Once the action was performed on the project, the project file is automatically checked back in.



Notes

The "Automatic" mode quickly becomes essential for the large projects shared by several developers.

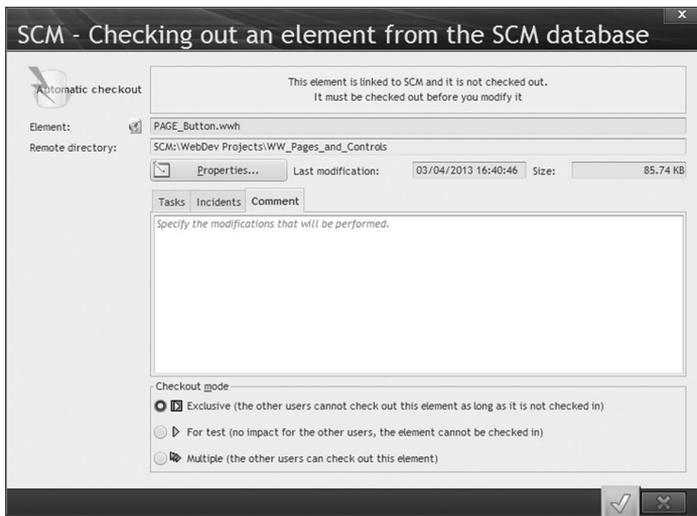
**In this tutorial, we will be using the automatic check-out.**

- ▶ To check whether the automatic check-out is enabled, on the "SCM" pane, in the "Project" group, expand "Master/Guest" and make sure that "Manage the project check-out automatically" is checked.

## Checking out an element

We are now going to work with the SCM. To modify a project element, this element must be checked out.

- ▶ We are going to check out the "PAGE\_Button" page:
  1. Select the "PAGE\_Button" page in the project explorer.
  2. Double-click the element to open it in the page editor.
  3. Display the code of the "Value of a control" button ("Code" from the popup menu).
  4. We are going to enter a comment in front of the "Info ..." line. Position the cursor and press the ENTER key.
  5. The window for automatic check-out is displayed:



Notes

### Check-out: Specific feature of the code editor

Regardless of the mode for checking out the project elements (automatic or standard), a modification made to the code editor triggers an automatic check-out of the element where the modification is performed.

6. Three check-out modes are available:
    - **Exclusive** (recommended mode): no one else will be able to check out this element until it is checked back in. The element can be checked out for test only.
    - **For test**: the element can be modified but the modifications will not be checked back in.
    - **Multiple**: the element can be checked out by the other users. In this case, the differences between the different versions of the element can be viewed when the element is checked back in. This mode is reserved to specific cases and to experienced developers.
  7. The page will be checked out in exclusive mode. Keep the "Exclusive" option checked.
  8. Enter a comment ("Adding a comment" for example). This comment will be useful for the other developers.
  9. Validate the check-out.
- The page is checked out. The code can be modified.

## Modifying the checked-out element

The method for modifying a checked-out element (GUI, code, ...) is the same as the method for modifying an element in a project not managed by the SCM.

However, the modifications performed on a checked-out element will not be visible to the other developers.

If another developer runs the checked-out element, the element that is currently found in the SCM database will be used.

This enables you to modify an application while keeping a steady version in the SCM database.

► Modify the checked-out page.

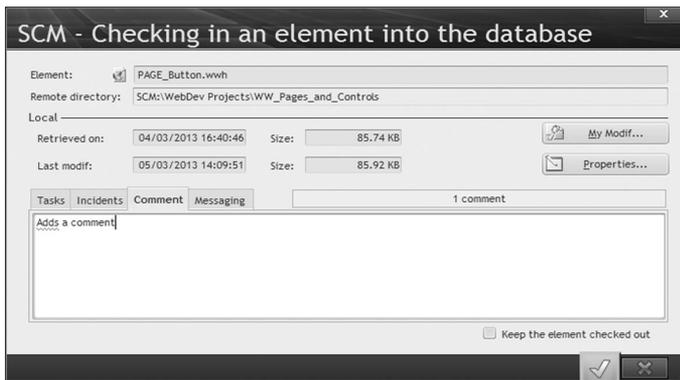
1. Enter the following comment: "// Display a message". Close the code window.
2. Change the location of controls, modify the captions, add controls or code, ...
3. Save your page (CTRL + S).

► Check your modifications.

## Checking the checked-out element back in

Now that the modifications have been made, we are going to check the page back into the SCM. Then, your modifications will be accessible to the other developers.

- On the "SCM" pane, in the "Current element" group, click the "Check in" button. The following screen is displayed:



This screen is used to:

- find out the modifications performed by comparing the element found in the SCM database with the checked-out element ("My Modif" button).



Notes

### Merging code

An element can be compared to one of its earlier versions. This allows you to compare the code in order to retrieve a section of code that was "lost" or accidentally deleted by another developer.

- access the history of the element found in the SCM database ("Properties" button).
- enter a comment about the modifications performed. By default, WebDev proposes the comment that was entered during the check-out.
- send a message to the other developers.

- check in the modifications made to the element while keeping the element checked out ("Keep the element checked out").



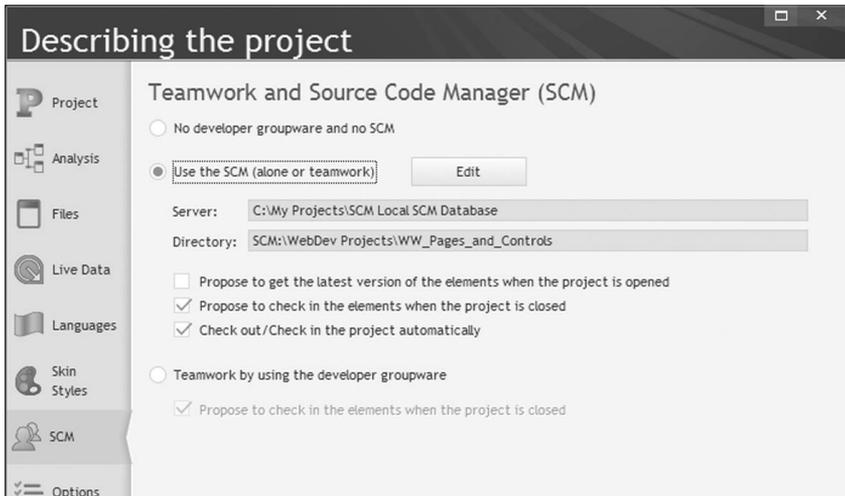
Notes

If you are using the Control Centers, the current task can be ended when the element is checked back into the Source Code Manager. This feature is useful to follow the monitoring of the tasks, the corrections of bugs, ...

- ▶ Validate the check-in.

## Synchronizing the project

Several options can be used to configure a project handled by the SCM. These options are grouped in the "SCM" tab of the project description (to display the project description, click "Description" on the "Project" pane).



These options are as follows:

- *Propose to get the latest version of the elements when the project is opened.*  
When opening a project found in the SCM database, this option proposes to retrieve the latest version of the project elements.  
By default, the latest version of the elements is automatically retrieved.
- *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.

## Off-line mode (or mobile mode)

The SCM allows you to work in off-line mode (or mobile mode).

This mode allows a developer who is using a laptop computer to continue to work on a project found in the SCM database while being disconnected from the SCM database.

The principle is straightforward:

- before the disconnection, on the "SCM" pane, in the "Other actions" group, expand "Remote work" and select "Disconnect for a mobile use".
- during the reconnection, on the "GDS" pane, in the "Other actions" group, expand "Remote work" and select "Reconnect and synchronize". Then, all you have to do is check back in the modified elements.

In mobile mode, two solutions are available for checking out elements:

- No element is checked out from the SCM. The other developers will be able to work on the same elements as you while you are working in off-line mode. \When you reconnect to 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. Nobody can use the element while you are working in off-line mode.

## SCM administrator

The SCM administrator enables you to handle the different projects included in the Source Code Manager.

It allows you to:

- manage the SCM databases (creation, connection to a SCM database).
- manage the files and directories found in a project of the SCM database (add, delete, rename files and directories).
- manage the different files found in the SCM database (check-in, check-out, share,...).
- start some tools (options, maintenance, ...).
- show the history of an element.
- show the status of the elements.
- perform backups.
- grant rights to the different SCM users.
- list the projects in which you are taking part in order to dissociate from them (if necessary).

- ▶ Start the SCM administrator: on the "SCM" pane, in the "SCM database" group, click "Manage". All the project elements are listed in the administrator.

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

## Disconnecting from the SCM

In the rest of this tutorial, we are going to use the "WW\_Pages\_and\_controls" project. To simplify the operations, we advise you to disconnect from the SCM:

1. Display the project description of the project: on the "Project" pane, in the "Project" group, click "Description".
2. In the "SCM" tab, select "No developer groupware, no SCM".
3. Validate the description window of the project.

# LESSON 9.3. CONTROL CENTERS

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

---

- Overview
- The Project Monitoring Center
- The other Control Centers



Estimated time: 20 min

## Overview

---

WebDev proposes several tools to follow the development of a project, from design to maintenance. These tools are called the Control Centers.

The HFSQL Control Center was presented in a previous chapter. This Control Center (redistributable) is mainly used to manage the HFSQL Client/Server sites deployed.

In this part, we are going to present the Control Center linked to the development of an application: the Project Monitoring Center

This center is using a specific database. This database can be:

- a HFSQL Classic database: the path of this database was specified during the first WebDev startup. This database is installed by default in a sub-directory of WebDev 19.
- a HFSQL Client/Server database.

This information can be modified at any time from the WebDev options:

1. In the "Home" pane, expand "Options" and select "General options of WebDev".
2. In the "User" tab of the WebDev options, click the "Parameters of Control Centers" button.

## The Project Monitoring Center

---

The Project Monitoring Center is the heart of the organization. The Project Monitoring Center is used to organize and schedule a project, from design to delivery. It allows you to define the tasks that must be performed in a project, the developers working on it, the durations, the progress status, ...

To check the Control Centers, we will be using our application named "WW\_Site\_With\_Data".

- ▶ Start the Project Monitoring Center: on the "Tools" pane, in the "Control Centers" group, click "Project Monitoring".

**Notes**

The Project Monitoring Center helps you manage and schedule any project or task (development, business or other).

The first thing to do is to define the working "Project". In our case, this project corresponds to a WebDev project but it could be any other project.

1. Create a new project ("Management .. Manage the projects", "+" button).
2. Enter:
  - the name of the project ("Tutorial" for example)
  - its description ("WW\_Site\_With\_Data" for example)
  - the type of application: Internet/Intranet

- the path of the project.

In our case, specify a project path outside SCM (for example, path of the "WW\_Site\_With\_Data.wwp" project, in the "Tutorial\Answers\WW\_Site\_With\_Data" sub-directory of WebDev).



Note

If the project is available in the SCM (Source Code Manager), you have the ability to enter the SCM database and the remote directory corresponding to the project.

- a group
  - the associated pool (a new one can be created).
  - the color associated with the project (used to easily identify the different projects).
3. Validate. The screen used to select the contributors of your project is automatically displayed. Keep yourself and validate.
  4. Close the window for project management.
- You now have the ability to define tasks on the project. We are going to create a task in order for our project to become a multilingual project:
1. Select "Tasks .. Create a new task".
  2. Specify the following characteristics:

**Saving a new task**

General information

Title: Tutorial: WW\_Site\_With\_Data project     Associated with the project: Tutorial  
 Personal tasks (outside project)

Id: 0

Status: In progress    Contributor: FP

Type: Development task

Dates and durations

Creation date: 3/6/2014    Deadline:

Fixed task

Estimated Time: 0 Minutes    Edit the history

Progress bar: 0%

Document

Details    Links    Elements of the project    Source Code Manager (SCM)

>> Created on 3/6/2014 by FP:  
 Multilingual management

Buttons:

3. Validate. The task appears in the task list. The task is in "In progress" which means that it starts immediately.

You can create as many tasks as necessary. These tasks represent your schedule. These tasks can also be fixed tasks (a meeting at a given day and time for example) or recurring tasks (a meeting every Friday for example).

To enable a task, all you have to do is select "Start this task" from the popup menu of the task list. The use of the Project Monitoring Center is straightforward: as soon as you want to perform a task, you select this task in the Project Monitoring Center and you specify that you "start this task" (from the popup menu of the task list). If your project is linked to a WebDev project, the corresponding task is enabled as soon as the project is opened in WebDev.

## Managing your time

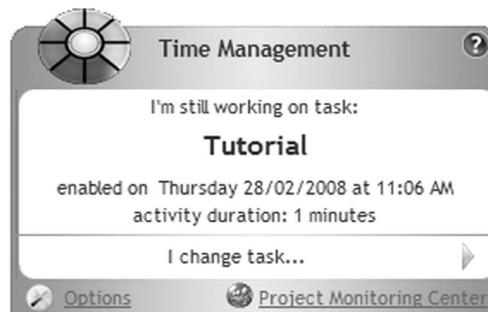
Time management is extremely important. Where is time spent in a day? How to find out without adding constraints, without requiring fastidious time keeping from team members and without making people feel like somebody is watching over their shoulder?

Entering the time spent on the tasks is an interesting feature of WebDev. This feature is linked to the use of the Project Monitoring Center.

**The principle is straightforward:** the task list is entered in the Project Monitoring Center.

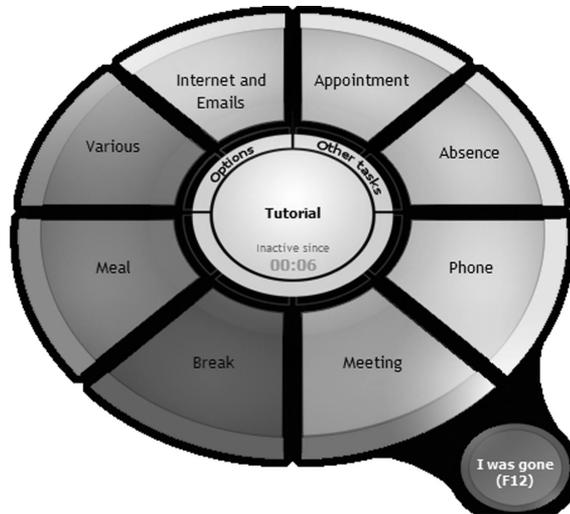
- ▶ To enable the help for time management:
  1. In the Project Monitoring Center, select "Parameters .. Options of Project Monitoring Center".
  2. In the "Help for time management" tab, select "Enable the help for time management".
  3. Validate.

The current task is called back at regular time intervals.



The user can validate the current task or change his current task if necessary. Furthermore, if the computer is idle for a set amount of time, a "task wheel" is displayed.

This task wheel enables you to select the interrupting task corresponding to the time spent while the wheel was displayed.



### Configuration

The time management can be enabled/disabled in the window for configuring the time management. This window can be displayed:

- in the Project Monitoring Center ("Parameters .. Options of Project Monitoring Center").
- in WebDev: to do so, on the "Project" pane, in the "Other actions" group, expand "Time management" and select "Options for time management".

**Note:** WebDev also includes the Quality Control Center. This center is used to enter the different problems that occurred on a project and to follow their process.

# LESSON 9.4. BUSINESS RULES

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

---

- Overview
- Creating a business rule
- Validating a business rule



Estimated time: 20 min

## Overview

---

WebDev allows you to manage the business rules.

A business rule is used to define a specific operating mode or to specify a particular process. For example: the calculation of a specific VAT rate, the rules for changing the status of a customer, the formula for calculating the shipping cost, ...

A business rule can be simple or complex, and it can affect one or more elements found in one or more projects.

### Example

To see the benefits of the business rules, we are going to create a business rule on the "WW\_Site\_With\_Data" project that was created in part 2 of this tutorial.

If this application was not created beforehand, a corrected version is available. To open this corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Site with data (Answer)". The project is loaded.

## Application on a real example

---

### Creating a business rule

- ▶ Open the "WW\_Site\_With\_Data" project and display the "PAGE\_Book" page.
- ▶ We are going to modify this page in order to manage the books whose entry date is recent (less than a year). Indeed, a book whose entry date is recent will be highlighted via a flashing image.
  1. Create an Image control and position it beside the "Entry date" control.
  2. Display the description of the Image control.
  3. Modify the name of this control: this image is named "IMG\_Image1".
  4. Via the "Catalog" button, select a GIF animation ("New" for example). Validate the selected image and its integration in the site directory. In the "GUI" tab of the control, indicate that the control is invisible.
  5. Validate the description window of the control.
  6. Enter the following code in the exit code with modification of the "Entry Date" control:

```
IF DateDifference(EDT_EntryDate, Today)<365 THEN
    IMG_Image1.Visible=True
END
```

This code is used to display the flashing image only if the entry date of the book is less than one year.

- ▶ We are now going to create a business rule on this page.

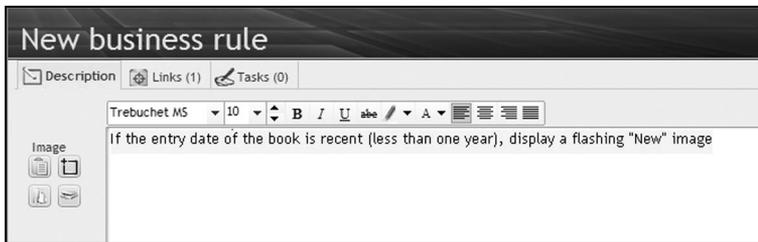


Note

A business rule is used to define a specific operating mode or to specify a particular process. For example: the calculation of a specific VAT rate, the rules for changing the status of a customer, the formula for calculating the shipping cost, ...

In our case, the business rule will be applied to all the books whose entry date is less than one year. In this case, a specific process must be performed: display an image indicating that it is a new book.

1. Display the pane of business rules: on the "Home" pane, in the "Environment" group, expand "Panels" and select "Business rules".
2. By default, the new business rule will be created on the selected element: select the "Entry date" control.
3. Click the "+" button: the screen for entering the business rule is displayed.
4. The description of the rule is as follows: "If the entry date of the book is recent (less than one year), display a flashing "New" image".



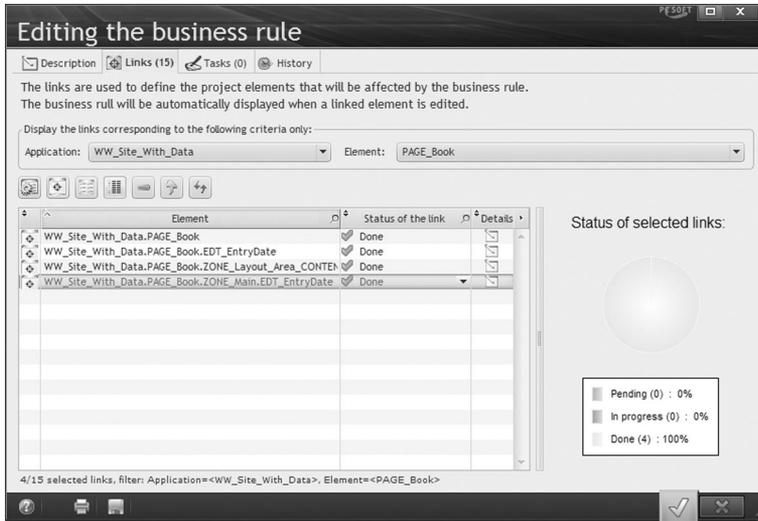
5. We are now going to define the elements onto which the business rule must be applied. Click the "Links" tab.
6. We are going to define an automatic link: the rule will be applied as soon as an element that fulfills the conditions is created. Click the "Conditions of automatic links" button (  ). A new window is displayed.
7. We are going to define the selection conditions of the project elements. Select "The current project only" for the "Automatically find the elements in:" option.
8. We are going to define the selection conditions of the business rule. Click "+" and specify the following information:



The rule will affect all the page controls linked to the "Book.EntryDate" item. Validate. The selection condition is displayed in the window for defining the conditions.

9. Validate this window.
10. Validate the window for "Defining the conditions of automatic links".

**11.** In the displayed list, all the links are "Pending". The business rule can be marked as "completed" for the "PAGE\_Book" page. To do so, select each element in the list and choose "Done" in the "Status of the link" combo box.



**12.** Validate the creation of the business rule.

**13.** Save your page.

## LESSON 9.5. INTERNAL COMPONENTS

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

---

- What is an internal component?
- Creating an internal component step by step.
- Distributing and using an internal component.



Estimated time: 30 min

## Overview

---

An internal component is used to group several project elements. This grouping is mainly used to share elements between different projects (via the Source Code Manager).

When an internal component is included in a project, all the elements of the component are included in the project. The public elements can be handled in the editor directly. Furthermore, the internal component can be debugged from the project that is using it.

The projects that use an internal component have access in the WebDev editor to the name of the objects, procedures or methods made visible by the creator of the component.

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

How do I proceed? Don't change anything, create your pages, procedures, classes. Then, choose the option for creating a component and that's it!

A component can contain source code, pages, an analysis, data files, etc!



Notes

The following methods can be used to share "code":

1. The sets of procedures
2. The classes
3. The supercontrols (via the use of a dictionary)
4. External components
5. Internal components

Let's present some cases in which the components can be useful.

### Teamwork

A developer creates and maintains a component that is made available to the other developers.

### The databases accessed by several projects

When the same database is accessed by several projects, you often find inconsistencies in the database due to modified or old source code. By grouping the operations for accessing the database (at least in write mode) in an external component, a single source code must be checked and maintained; therefore, the risks of database inconsistency are reduced.

### The processes used in several projects

You will often find more or less complex processes used in several projects. These processes can be re-used via "sets of procedures" or "classes". In this case, the modifications may not be applied to the different projects, making the elements no longer compatible between themselves. Using components prevents such out-of-sync modifications, preserves the compatibility between projects and simplifies the common modifications.

Furthermore, the PRIVATE keyword enables you to make sure that your source code remains confidential at all levels of the component. When your component is re-used in another WebDev project, the source code cannot be edited but the documentation regarding the use of the functions (procedures for example) will be displayed!

## The ability to distribute a feature or set of features

The components are used to implement a feature or a set of features. Other WebDev developers will be able to include these features in their own projects.

### Multi-product component

A component can be designed to operate in:

- a WinDev application
- a WebDev application
- a WinDev Mobile application
- the three types of applications.

## Step by step

### Step 1: Creating an internal component

We are going to create an internal component used to display news in a site. This component includes:

- an internal page that contains the different controls used to display and browse the news.
- an analysis that describes the data file containing the news.

To avoid having to develop the code required for the component to operate, all the necessary elements have been grouped in a project named "WW\_Internal\_Component". We are going to use this project to create our internal component. A new project will be created later to use this internal component.

► To open the example project:

1. Close the current project if necessary. The home window is displayed.
2. In the home window, click "Tutorial" and select the project named "Internal Component (Exercise)".

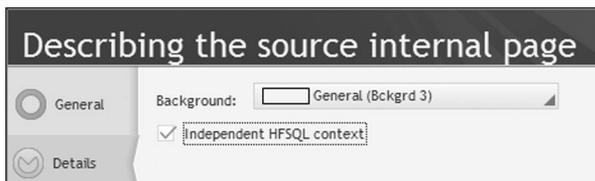
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Internal component (Exercise)".

3. The project is loaded

► We are going to run the test of the internal page of this project:

1. Display the internal page named "IPAGE\_News" (double-click its name in the project explorer).
2. Display the description of the internal page ("Description" from the popup menu of the internal page).

In the "Details" tab, the "Independent HFSQL context" box is checked.



This option means that this page uses a browse and specific data. In our case, the internal page browses the data found in the "News" file.

3. Validate the description window of the page.
4. Run the test of the page (  among the quick access buttons).

5. Click the different buttons to browse the news.

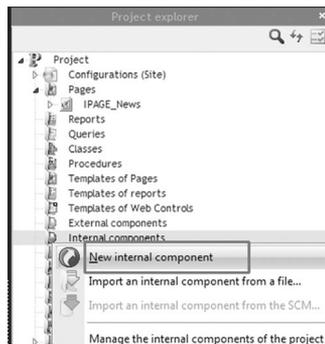


6. Stop the test and go back to the editor.

We are now going to create our internal component.

► To create an internal component:

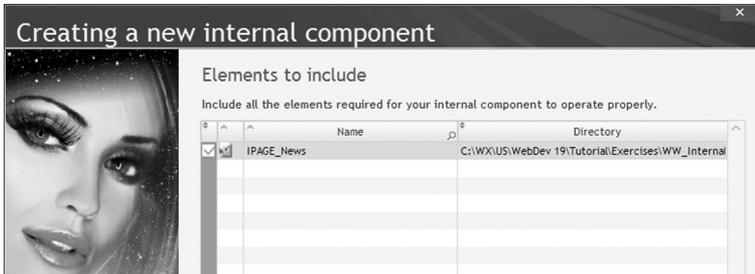
1. In the project explorer:
  - Select the "Internal components" folder.
  - Display the popup menu (right mouse click).



- Select "New internal component".
  - The wizard for creating an internal component starts.
2. Go to the next wizard plane.
  3. Identify your component:
    - Enter the name: "MyInternalComponent".
    - Enter the caption: "MyInternalComponent component".

4. Go to the next screen.

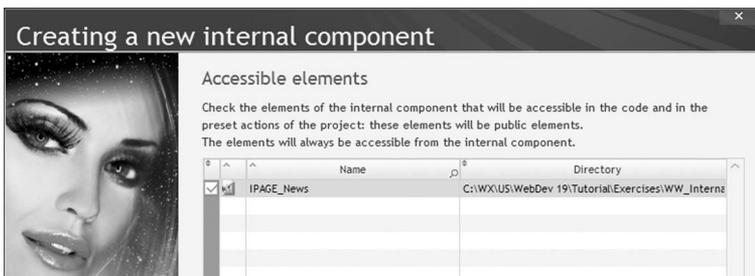
Select the elements that are included in the component. In our case, the internal page named "IPAGE\_News" must be selected.



5. Go to the next screen.

The wizard asks you to select the component elements that will be accessible from the client project.

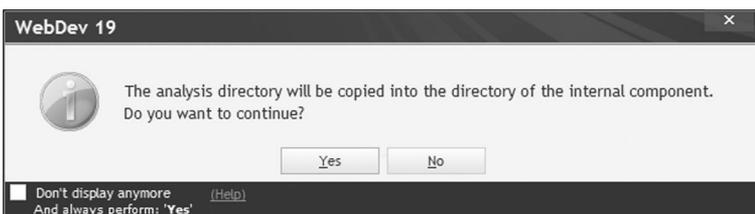
In our example, the internal page is accessible:



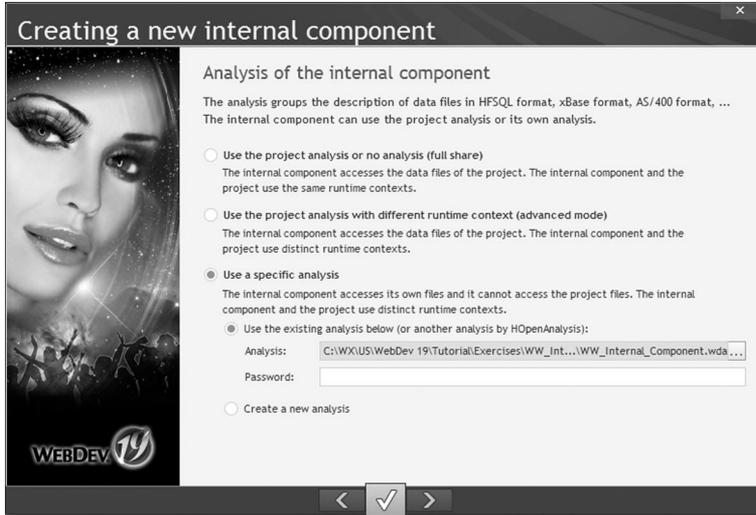
6. Go to the next screen. This screen is used to specify the mode for managing the component data. In our case, the internal component uses its own analysis.

Select "Use a specific analysis", then in the "Analysis" control, select the analysis corresponding to the current project ("WW\_Internal\_Component.wda" file found in the "WW\_Internal\_Component.ana" sub-directory of the project).

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

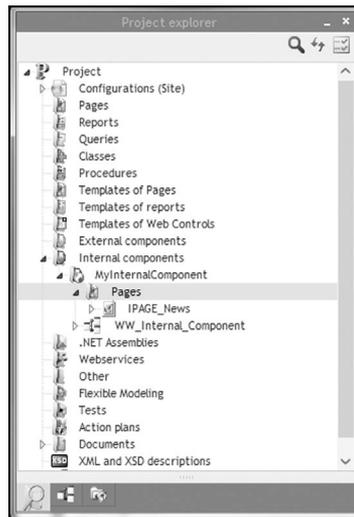


Accept



7. Go to the next screen. WebDev is going to create the internal component in a specific directory of your project.

In the project explorer, the "IPAGE\_News" page is no longer found in the list of project pages. On the contrary, the internal component that was just created is listed in the "Internal Component" folder, along with its different elements: analysis and internal page.



## Step 2: Using the internal component

Once created, your internal component can be used in any other WebDev project. Let's see how this component can be re-used.



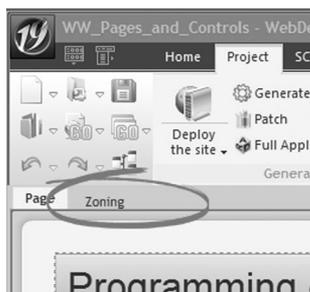
Notes

In this example, we will see a "direct" use of the internal component. To share resources, we recommend that you use the internal components via the Source Code Manager (SCM). See the online help (keyword: "Internal component") for more details.

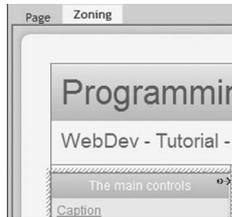
In our example, we are going to import the internal component into the "WW\_Pages\_and\_controls" project.

- ▶ Open the "WW\_Pages\_and\_controls" project.
  1. Close the current project if necessary. The home window is displayed.
  2. In the home window, click "Tutorial" and select the "Pages and controls" project. The project is loaded.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Pages and controls".
- ▶ Include the internal component in the project.
  1. On the "Project" pane, in the "Project" group, expand "Import" and select "Import an internal component .. From a file".
  2. Select the "MyInternalComponent.wci" file (found in the "WW\_Internal\_Component\MyInternalComponent" sub-directory of the WebDev tutorial).
  3. The internal component is included in the project.
- ▶ In the Windows explorer, copy the data files (NewsItem.fic, NewsItem.ndx and NewsItem.mmo) found in the "Exe" sub-directory of the "WW\_Internal\_Component" project to the "Exe" sub-directory of the "WW\_Pages\_and\_controls" project.
- ▶ To use the internal component, we are going to create an "Internal Page" control in the page template to include the internal page of the component. To do so:
  1. In the editor, display the page template named "PAGETPL\_TUT" (double-click its name in the project explorer, "Page templates" category).
  2. We are going to move the menu to the left of the page to insert the internal page of the component. To do so, we are going to use the features of the zoning:
    - Click the "Zoning" tab at the top of the page.



- The red lines identifying the areas appear in the editor.
- With the mouse, click the red line above "The main controls".



- Keep the mouse button down and move it to the bottom: the area is increased.



3. Click the "Page" tab to go back to edit mode.

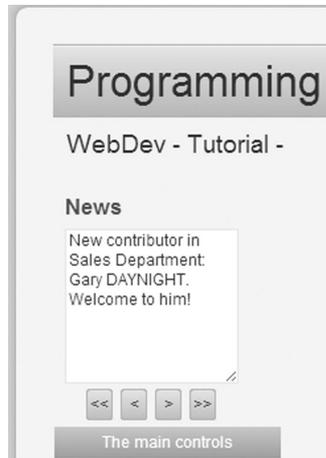
► To create the "Internal Page" control:

1. On the "Creation" tab, in the "Containers" group, click "Internal page".
2. In the page template, click in the area that was just created. The "Internal Page" control is automatically created. Resize the control to occupy the entire space.



3. Display the control description ("Description" from the popup menu).
4. In the list of available internal pages, select the internal page coming from the component: "IPAGE\_News".
5. Validate.

- ▶ To center the internal page:
  1. Select the Internal Page control.
  2. On the "Alignment" pane, in the "Centering and distribution" group, click "Center in the parent".
- ▶ We are going to run the test of the page template (  among the quick access buttons). The news are displayed in the page at run time.



- ▶ To apply the modifications performed in the page template to all the site pages:
  1. Click the icon  in the orange "Page template" area at the top of the screen.
  2. The list of relevant pages is displayed.
  3. Validate this window.
  4. Run the test of the project. The news are included in all the pages.

# LESSON 9.6. MULTI-CONFIGURATION

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

---

- Overview
- Using the same project for different configurations



Estimated time: 20 min

## Overview

---

You have created a site and you want to create a component from some project pages? Your project contains procedures that could be used in a Web service?

How can I avoid the multiplication of projects? A single project and several destinations, that's the challenge that was resolved by the project configurations.

The project configurations are used to create several different "targets" from the same project. At any time, you have the ability to choose the requested configuration and to generate in a single operation all the elements for all the configurations of a project.

A practical example? We are going to test the project configurations on our full application.

## Creating a project configuration

---

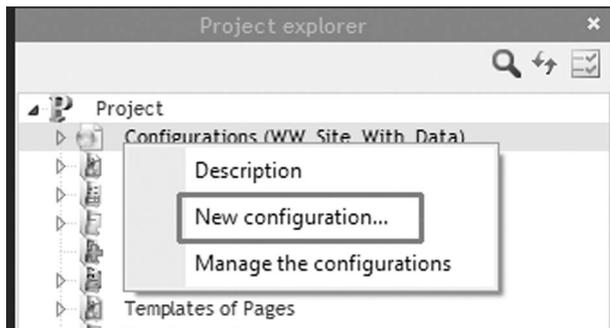
We are going to create a project configuration in order to create a component from the "My\_Library" project.

► To create the project configuration:

1. Close the current project if necessary. The home window is displayed.
2. In the home window, click "Tutorial" and select the "Site with data (Answer)" project. The project is loaded.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Site with data (Answer)".

3. In the "Project explorer" pane, select "Configurations (WW\_Site\_With\_Data)".
4. Select "New configuration" from the popup menu.



5. The wizard for creating a project configuration starts.

6. In the general information of the project configuration, specify the name (test component), the description (Test) and the type of generation (External component for our example).



Go to the next screen.

7. Specify the products in which the component will be used (WebDev).

Go to the next screen.

8. Specify the elements that will be included in the configuration. For our example, include the "PAGE\_Book" and "PAGE\_Search" pages. Go to the next screen.

9. Validate the creation of the configuration.

10. The "Test component" configuration is automatically selected in the "Explorer" pane.

Select the "WW\_Site\_With\_Data" configuration and select "Enable this configuration" from the popup menu.

## Managing the project configurations

To define all the option of the project configurations, on the "Project" pane, in the "Project configuration" group, click "Manage the configurations".

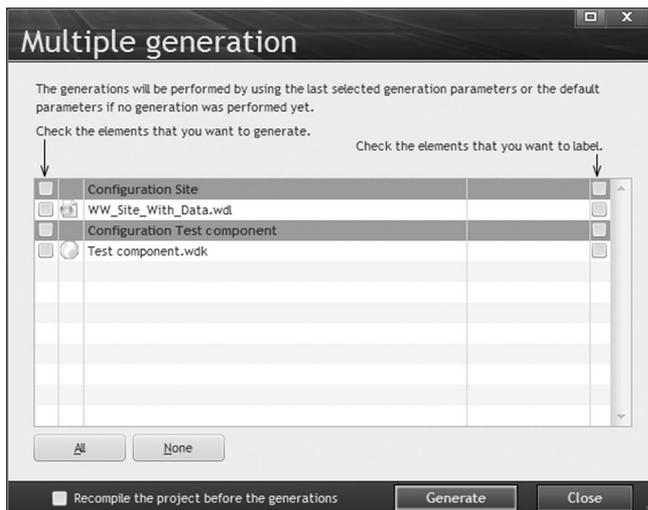
You have the ability to:

- Create a new configuration or delete an existing configuration.
- Generate a configuration
- Configure the test mode
- Find out the description of the main element found in the configuration
- Configure the actions before and after generation (copy the element into a specific directory for example).

## Multiple generation

You know how to generate a configuration; however, after a modification, it is often necessary to generate the elements corresponding to all the configurations defined on the project. To do so, on the "Project" pane, in the "Generation" group, expand "Generate the configuration" and click "Multiple generation".

All you have to do is select the elements to generate. This generation will be performed in a single operation.



# LESSON 9.7. THE DEBUGGER

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

---

- Overview
- Debugging a project element
- Debugging a project
- Advanced options



Estimated time: 20 min

## Overview

---

When developing a full site (WW\_Site\_With\_Data project in part 2), several tests have been run: tests on pages, tests on queries, tests on reports, ...

An error occurs and you still don't know why? WebDev enables you to go even further in the tests, by allowing you to follow the execution of a program step by step: you view and you validate all the code lines that are run via the debugger.

When your site is run, the debugger enables you to find out the values of the different variables used, to run some processes, to re-run code lines. You even have the ability to modify code lines during the debug operation.

The debugger can be used during the development (to find out the reason of a problem during the development step) or at run time: you have the ability to debug a site started on the computer or on another computer, a component, ...

Interested in these features? Let's check these features directly.

## Debugging a page

---

### Starting the debugger

To run the test of the page, click the GO icon (or press [F9]).

Several methods can be used to debug a page:

- 1st method: You want to start the debugger during the test of your application, from a specific action: all you have to do is press [CTRL] + [PAUSE].
- 2nd method: You know which section of code triggers the error in your page, you want to start the debugger from a specific code line: you can implement a breakpoint or use the STOP statement.

For example, the debugger will be started on one of the pages found in our site developed in part 2:

1. Open the "WW\_Site\_With\_Data" project. If the application was not developed, a corrected project is available. To open this corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Site with data (Answer)".
2. Open the "PAGE\_Book" page.
3. Display the code of the button for exact-match search (BTN\_ExactMatch).
4. With the mouse, click the yellow column found before the first code line. A red dot is displayed: it is a breakpoint.

```
Click of BTN_ExactMatch ( CELL_TPL_Main ) (server)  AJAX  If
● {HReadSeekFirst(Book, BookID, COMBO_Title)
  IF HFound(Book) THEN
    FileToPage()
  END
```

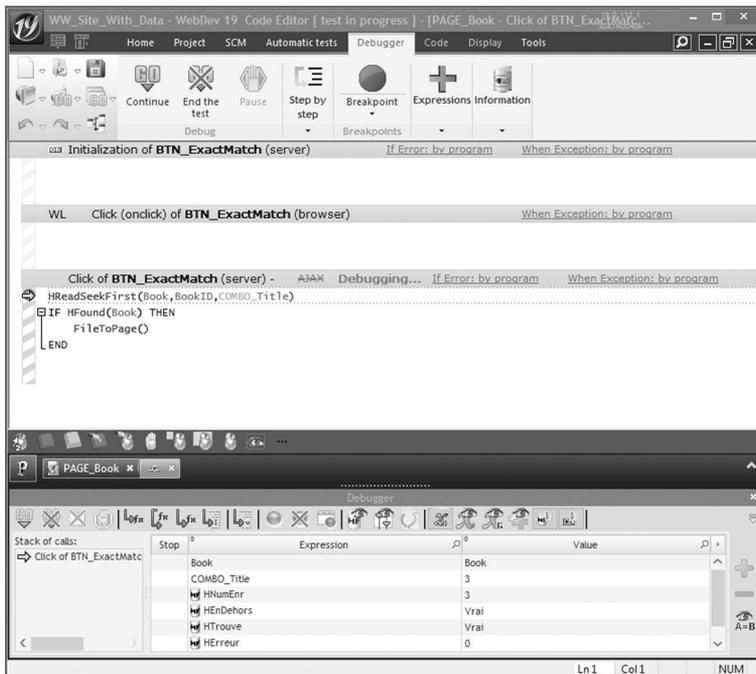
In test mode, the debugger will be enabled as soon as the lines containing a breakpoint are run.



Notes

The STOP keyword and the breakpoint have the same effect. Like the breakpoint, it is ignored in the executable.

5. The breakpoint being implemented, let's now run the test of our page (  among the quick access buttons).
6. Select an element in the combo box.
7. Click the "Exact-match search" button.
8. The code editor is displayed in debug mode.



The "Debugger" pane appears in the lower section of the screen. This pane displays two distinct sections:

- **the stack of calls:** This section is used to find out the hierarchy of the processes displayed in the debugger. In our example, we are currently debugging the click process on the "BTN\_ExactMatch button".
- **the list of expressions to evaluate:** By default, the main variables used in the code are displayed in this section. You have the ability to add variables to follow their evolution.

Let's study the first debugging lines:

1. In the code editor, the yellow arrow indicates the line that will be run.
2. Press [F7]. The line is run. The yellow arrow moves to the next line.
3. Position the mouse cursor on the "END" line.
4. Press [F6]. All the lines are run up to the cursor.

We are now going to study the values of the variables used:

- The variables handled in the code are automatically viewed in the "Debugger" pane. Their value is automatically displayed.
- To find out the value of a variable found in the code, the variable must be hovered by the mouse cursor: the value is displayed in a tooltip.
- To add an expression into the "Debugger" pane, select the expression in the code editor and select "Add the expression to the debugger" from the popup menu of the code editor.

When the use of the debugger is no longer required, you can:

- press [F5] to continue the application without using the debugger.
- press [SHIFT + F5] to stop the test. In this case, the code editor will be re-displayed in standard mode on the code currently run.

The main topics of the debugger have been presented here. Several other options are available, such as the auto-stop expressions, the code modification while debugging, ... See the online help (keyword: "Debugger") for more details.

## Trace window of the debugger

---

To debug a project, you also have the ability to use **Trace**. This function is used to display the requested information in a "trace window". You can for example display the value of a variable, a message to find out the way taken by the application, ...

During a GO (of the project, page, ...), the trace window is automatically displayed when **Trace** is reached. This window disappears when the test is ended. However, the information displayed in the trace window can be retrieved via the "Trace of debugger" pane.

This pane displays the information displayed during the test with "Trace". A click performed on one of the lines found in the trace pane allows you to automatically access the corresponding code line.



Notes

An error occurs during the test of your application? You didn't have time to write down the information displayed on the screen? Don't worry, this information was stored by the trace pane. A double click performed on the error message allows you to display the detailed information.

## Debugging a project

---

You know how to debug a page but you also have the ability to debug the entire project. In this case, specific parameters can be defined: on the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Configure the test mode".

The WebDev debugger is a powerful tool, used to debug several cases. You have the ability to debug:

- a component, from a project that uses the component for example.
- a WebDev dynamic site (on the current computer or on a remote computer)
- a site already run (on the current computer or on a remote computer), ...

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

# LESSON 9.8. THE PERFORMANCE PROFILER

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

---

- Overview of the performance profiler.
- Using the performance profiler.



Estimated time: 20 min

## Overview

The performance profiler enables you to check and optimize the execution time of your site.

The performance profiler should be used to optimize your project (before it is distributed for example).

### The principle is straightforward:

You run the test of your site. During this test, the performance profiler keeps track of all the actions and processes run.

At the end of the test, the performance profiler displays:

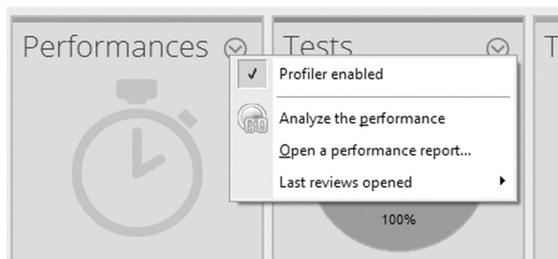
- the 10 most time consuming processes
- all the actions performed in the site whose test was run, sorted by duration (from the longest one to the shortest one).

You have the ability to select a specific process in order to analyze the reasons for its duration.

## Using the performance profiler

The performance profiler can be started:

- **from the dashboard:** In the "Performances" Widget, expand the arrow button and select "Detailed Go of project".



- **from the WebDev editor:** on the "Project" pane, in the "Audit and performances" group, click "Analyze the performance".

In this case, the project is automatically run in test mode. You can handle your site 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 result of the analysis. This result is saved in the format of a WPF file.

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

|                      |  |
|----------------------|--|
| <b>ProfilerStart</b> | Starts "collecting data" for the performance profiler. |
| <b>ProfilerEnd</b>   | Stops "collecting data" for the performance profiler.  |

In this case, only the code found between **ProfilerStart** and **ProfilerEnd** is analyzed. The corresponding WPF file can be opened: on the "Project" pane, in the "Audit and performances" group, expand "Analyze the performance" and select "Open a performance report".

By default, the performance profiler saves the statistics performed on the code of the application in a <Project Name>.WPF file. To change this name, click the "Save as..." button in the performance profiler.

To open a specific statistical file (a file created by programming for example):

1. On the "Project" pane, in the "Audit and performances" group, expand "Analyze the performance" and select "Open a performance report".
2. Specify the path and name of the statistical file.

The list of the last statistical files opened can be found in the menu opened by the "Analyze the performances" option.

## Optimizing a process with the performance profiler

### Reading the result of the performance profiler

The performance profiler presents the result of the analysis in several tabs:

- the "Summary" tab presents the 10 longest processes.
- the "Mapping" tab presents a graphical view of the main processes.
- the "Details" tab presents all the processes run during the test of the application (from the slowest one to the fastest one).

The following information is displayed for each process run:

| Function  | Total time | Internal time | Nb of calls | Time 1 call | code %  | Parent         |
|---|------------|---------------|-------------|-------------|---------|----------------|
| Click of BTN_Print (server)                         | 2 s 317 ms | 2 s 317 ms    | 1           | 2 s 317 ms  | 0.5 %   | PAGE_Table_Boc |
| !PrintReport()                                      | 2 s 245 ms | 2 s 245 ms    | 1           | 2 s 245 ms  |         | PAGE_Table_Boc |
| Menu Selection of ZONE_Layout_Area_BRW.MENU_M...    | 918 ms     | 258 µs        | 1           | 918 ms      | 0.0 %   | PAGE_Home_ZO   |
| PageDisplay("PAGE_Table_Book")                      | 917 ms     | 864 ms        | 1           | 917 ms      |         | PAGE_Home_ZO   |
| Connecting to the 'WW_Site_With_Data' site          | 141 ms     | 14 ms         | 1           | 141 ms      | 9.9 %   |                |
| Click of BTN_Modify (server)                        | 128 ms     | 13 ms         | 1           | 128 ms      | 0.1 %   | PAGE_Table_Boc |
| PageDisplay("PAGE_Home")                            | 127 ms     | 109 ms        | 1           | 127 ms      |         |                |
| Click of BTN_VALIDATE (server)                      | 117 ms     | 79 ms         | 1           | 117 ms      | 0.5 %   | PAGE_Form_Boo  |
| PageDisplay("PAGE_Form_Book")                       | 115 ms     | 77 ms         | 1           | 115 ms      |         | PAGE_Form_Boo  |
| HModify()   | 77 ms      | 77 ms         | 1           | 77 ms       |         | PAGE_Form_Boo  |
| sending the 'PAGE_Table_Book' page to the browser ( | 53 ms      | 53 ms         | 1           | 53 ms       | 100.0 % |                |
| PageDisplay("PAGE_TABLE_BOOK")                      | 38 ms      | 18 ms         | 1           | 38 ms       |         | PAGE_Form_Boo  |
| FileDisplay()                                       | 31 ms      | 31 ms         | 1           | 31 ms       |         | PAGE_Table_Boc |
| !Destination()                                      | 24 ms      | 24 ms         | 1           | 24 ms       |         | PAGE_Table_Boc |
| sending the 'PAGE_Table_Book' page to the browser ( | 20 ms      | 20 ms         | 1           | 20 ms       | 100.0 % |                |
| sending the 'PAGE_Home' page to the browser (interr | 17 ms      | 17 ms         | 1           | 17 ms       | 100.0 % |                |
| TableSelect()                                       | 13 ms      | 13 ms         | 1           | 13 ms       |         | PAGE_Table_Boc |
| sending the 'PAGE_Form_Book' page to the browser (  | 7 ms       | 7 ms          | 1           | 7 ms        | 100.0 % |                |
| Global Declarations of PAGE_Form_Book (server)      | 5 ms       | 5 ms          | 1           | 5 ms        | 0.3 %   | PAGE_Form_Boo  |

The following elements are displayed:

1. Total Time: Execution time of the function.
2. Internal time: Execution time due to the engine.
3. Nb calls: Number of calls made to the function (procedure or process).
4. Time 1 call: Time for running a call to the function (procedure or process).
5. % Code: Percentage of time spent in the process of the function or procedure (developer code that can be optimized).
6. Parent: Element that contains the process.

### Choosing a process to optimize

The process to optimize is chosen according to several criteria:

- the execution time of the process. The longest processes must necessarily be optimized.
- the percentage of time spent in the process of the function or procedure. The higher this percentage is, the greater the number of processes that can be optimized in the code.

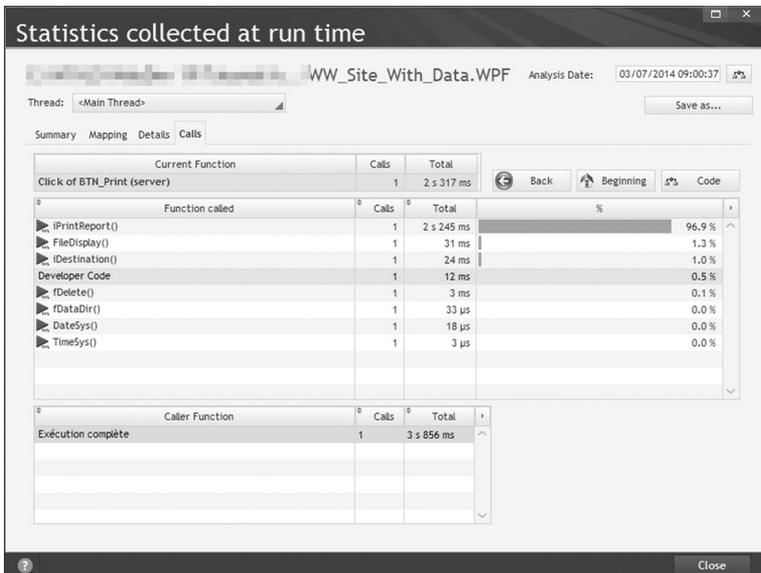
## Optimizing a process

Once the process to optimize is chosen, the performance profiler enables you to find out the details of the operations performed.

To display the details of the operations performed in a process:

1. Select the function to optimize in the "Details" tab.
2. Display the details of this function:
  - double-click the function.
  - click the "Calls" button.

The following tab is displayed:



This tab is divided into three sections:

- The selected function (with the number of calls to this function and the total processing time).
- The list of functions called by the selected function.
- The list of functions calling the selected process.

The list of functions called is used to improve the search for the process to optimize. Indeed, you have the ability to view the different functions called as well as the processing time for each function.

All the processes containing a WLanguage code (named "Developer code") can be optimized.

A double click performed on one of the functions found in this list ("Click on Validate" for example) allows you to view the details of the processes called by this function.

### Notes:

- The "Internal process of runtime engine" caption corresponds to the execution time of the function or procedure (for a WLanguage function for example). This time cannot be reduced and it cannot be optimized.

- The "Developer code" caption corresponds to the execution time of the code of the function or procedure (excluding the calls to sub-functions). This time can be reduced and it can be optimized.
- To quickly view the code of the current process, click the "Code" button. The profiler remains opened and the current code can be modified.
- The "Previous" button is used to go back to the calling function.
- The "Start" button is used to go back to the process at the beginning of the application.

# LESSON 9.9. 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 project, HTML pages, ...).



Estimated time: 10 min

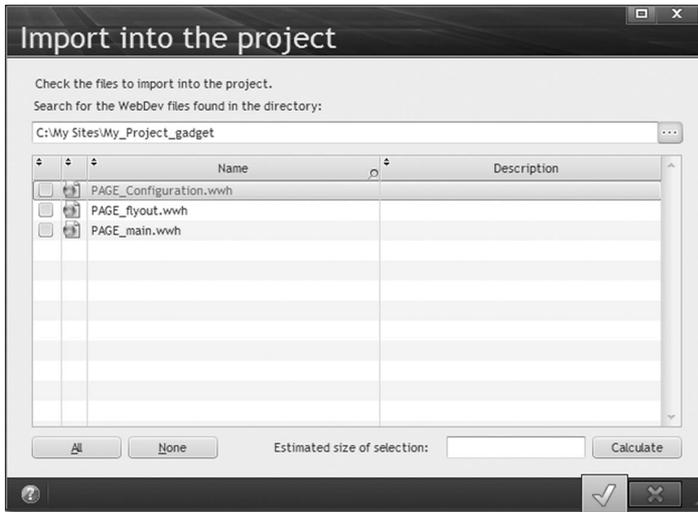
## Importing elements

You have the ability to import existing WebDev elements into your project. All types of WebDev elements can be imported:

- pages, reports
- classes, components
- procedures, ...

► To import existing elements into the current project:

1. On the "Project" pane, in the "Project" group, expand "Import" and select "WebDev elements and their dependencies...".
2. Click "..." and select the directory containing the elements to import (the directory must contain WebDev elements).
3. Validate. WebDev returns the list of elements that can be imported (the sub-directories are ignored).



4. Select the elements to import and validate.

The elements (and all the files used by these elements: images, ...) are now included in the project.



Notes

The "Calculate" button (found in the import window) is used to calculate the size of the selected elements along with their dependencies.

## Exporting elements

You also have the ability to export elements from your project to another directory for example. These elements can be re-used in other projects.



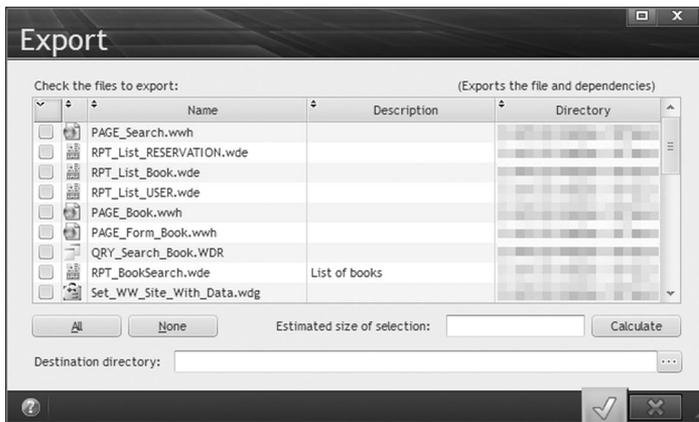
Caution!

Exporting elements is a lot more than a simple copy of elements. It is actually the only safe method for transferring a project element along with **all** 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...".

In the window that is displayed, select the project elements that will be exported.



2. Specify the name of the destination directory (or select it with the "..." button).

3. Validate.

The elements are exported to the specified directory. These elements are still available in your project.



Notes

The "Calculate" button (found in the import window) is used to calculate the size of the selected elements along with their dependencies.

## Specific import operations

---

### Importing a WinDev project

More and more WinDev applications must be changed into WebDev sites. This feature is now automatic.

WebDev enables you to import a WinDev window or a full WinDev project into a WebDev project.

During this import:

- The windows are changed into pages.
- The codes are changed into server codes.
- The elements with no equivalent in WebDev are imported as comments or as separate elements.

### To import a WinDev project into a WebDev project:

1. On the "Project" pane, in the "Project" group, expand "Import" and select "A WinDev or WinDev Mobile project". The wizard for importing a WinDev/WinDev Mobile project starts.
2. Select the WinDev project to import. This project will not be modified and a new WebDev project will be created.
3. Specify the name and location of the WebDev project to create.
4. If an analysis is linked to the project, specify its name, its location, and whether it must be used by the WebDev project.
5. Specify the elements that will be shared between the WebDev project and the WinDev project (common elements such as Reports, Classes, ...). If the elements are shared, they will not be copied into the WebDev project.
6. Validate. The WinDev project is changed into WebDev project.

### To import WinDev elements into a WebDev project:

1. Open the WebDev project into which the element must be imported.
2. On the "Project" pane, in the "Project" group, expand "Import" and select "WinDev or WinDev Mobile elements". The import wizard starts. Select the WinDev project containing the elements to import or select the WinDev elements to import. Go to the next screen.
3. Specify the elements that will be shared between the WinDev and WebDev projects. These elements will not be copied into the WebDev project.
4. Validate. The specified elements are automatically imported into the current WebDev project.

### Importing an HTML page

A page found in a non-WebDev site is useful? You want to retrieve its interface? Nothing's easier. WebDev enables you 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!



Notes

The feature for importing HTML pages is a great help given to the developers of WebDev sites. This feature is not designed to "suck up" the Internet sites. However, some display differences may remain in the HTML page before and after the import.

To import an HTML page into a WebDev project:

1. Open your WebDev project (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 the HTML page. This page can be a file found on your computer ("..." button) or an Internet address (http://www.mywebdevsite.eu/mapping.html for example). The HTML page is displayed in the "Page preview" frame.
4. Validate. A new WebDev page has been created.
5. Save the page. The page is automatically added to the list of project elements.

## LESSON 9.10. UML AND 3-TIER

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

---

- What is the UML?
- The different types of diagrams supported by WebDev.



Estimated time: 5 min

## What is the UML?

---

UML (Unified Modeling Language) is a concept for "modeling" processes.

## The diagrams managed by WebDev

---

### Class diagram

The class diagram is used for modeling the structure of a system and relationships between the different elements found in this system. It is mainly used in OOP.

In WebDev, via a UML class diagram, you have the ability to automatically generate the classes and the methods that derive from it.

Via "reverse engineering", you can then display the relationships between the different classes that were created.

### Use case diagram

Once the specifications have been defined, the use case diagram is used to establish in a simple and visual way the behavior of your project (correspondence between the implementations of user requests by the developers, ...). This type of diagram is sometimes called a "sequence diagram" (without specific chronology).

No application can be generated from this type of diagram. This type of diagram is only used when implementing the project management.

### Object diagram

An object diagram presents a set of objects and their relationships at a given time.

An object diagram is used to show a context (before or after an interaction between objects for example).

### Component diagram

A component diagram describes the physical and static architecture of a computer application. For example: source files, libraries, executables, ...

### Activity diagram

An activity diagram represents the behavior of a method or the flow of a use case.

### Sequence diagram

A sequence diagram represents the chronological order of the messages sent and received by a set of objects.

### Collaboration diagram

A collaboration diagram presents the structural organization of the objects that send and receive messages.

### State-transition diagram

A state-transition diagram presents a sequence of states that an object goes through during its lifecycle. It is used to describe the changes of states for an object or for a component.

A state is defined by its duration and by its stability.

A transition represents the instantaneous change from one state to another one.

A transition is triggered:

- by an event.
- automatically when no triggering event is specified.

### Deployment diagram

A deployment diagram presents the physical layout of the hardware devices (nodes) used in a system as well as the association between the executable programs and these devices.

We won't go into details about the UML language. See the online help (keyword: "UML") for more details.

## 3-tier

---

### What the 3-tier is?

The development in "3-tier" architecture is now simplified with WebDev.

The 3-tier architecture is designed to separate the 3 standard layers of an application: GUI, Processes and Data.

An application will include 3 independent layers:

- a presentation layer
- a process tier
- data access layer.

Specific functions (APIs in standard languages, advanced WLanguage functions) allow these 3 layers to communicate between themselves.

The reason for separating the layers is to make the layers independent in order to simplify the maintenance and future upgrades of the application (change of database system, transfer from a graphic environment to another one, ...).

This provides better security because the access to the database is allowed via the Processes layer only.

It also optimizes the teamwork and the multi-target development.

## How do I implement the 3-tier?

The dialog between the different layers is performed via classes or structures that are automatically generated by WebDev.

### 1. You choose the data sources (data files or queries that will be handled in 3-tier).

The different files and queries are created in the data model editor and in the query editor.

### 2. WebDev automatically builds the class diagram that corresponds to the classes and structures that will be generated.

To do so, in the data model editor, on the "Analysis" pane, in the "Analysis" group, expand "Generation" and select "Generate the UML diagram corresponding to the analysis".

You also have the ability to create, in the UML editor, a class diagram corresponding to the analysis:

- Create the UML model: Click  among the quick access buttons. In the wheel that is displayed, hover "Architecture" and click "UML".
- In the wizard, select the "Class diagram" option. Go to the next screen.
- Select "Build the class diagram corresponding to the analysis and to the queries".
- Choose the type of code to generate (procedural or object-oriented code).
- Create and save your diagram.

### 3. The classes and structures are generated in WLanguage.

On the "Project" pane, in the "Other actions" group, expand "UML modeling" and select "Generate the code". The sets of procedures and/or the classes are automatically generated.

### 4. It can be "improved" by developing your own methods in these classes.

The synchronization between the classes and the diagram is performed in real-time.

You also have the ability to modify the analysis and to synchronize the UML diagram with the analysis.

### 5. These classes or structures allow the "Process" layer and the "Presentation" layer to communicate between themselves.

## Handling the different layers

- For the "Access to data" layer:

The standard functions for accessing the data can be handled: read, write, queries, transactions, ...

- For the "Processes" layer:

Specific WLanguage functions can be used: **FileToMemory** and **MemoryToFile**. These functions are used to fill the members of the class from the items found in the corresponding data files (and conversely).

- For the "Presentation" layer:

The generated classes can be handled directly, no need to worry about the structure of the database.

For the communication between layers:

Depending on the communication protocol between the 2 layers (component, Web service, HTTP, WebDev site, ...), the classes can be serialized in XML or in binary format (**Serialize** and **Deserialize**).

# LESSON 9.11. FLEXIBLE MODELING

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

---

- Principle
- Operation



Estimated time: 20 min

## Principle

---

The modeling editor is used to represent all the organizations (existing ones or forthcoming ones). The modeling editor allows you to represent the organization of email management in your company, the organization of contacts with the customers, ...

A modeling example was created to help you discover the features of the modeling editor. This modeling represents part of the ERP (Enterprise Resource Planning) in a fictitious company called Precilia.

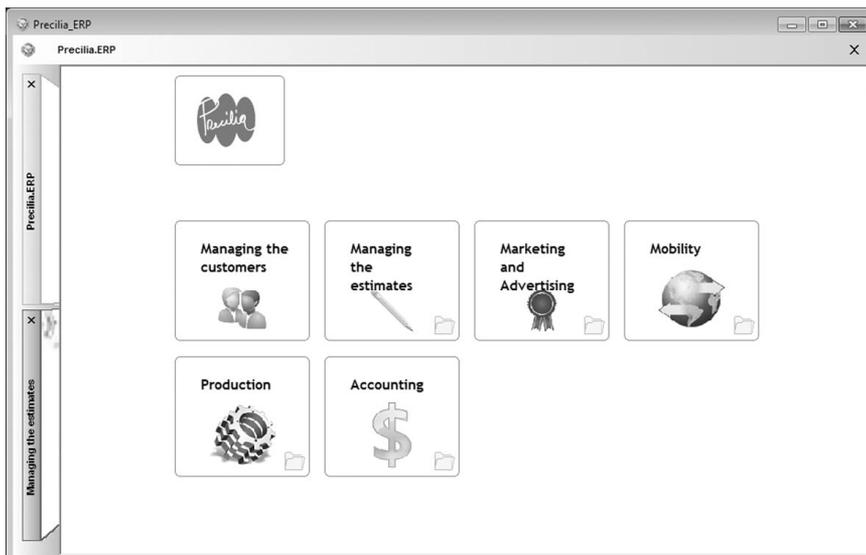
## Operations

---

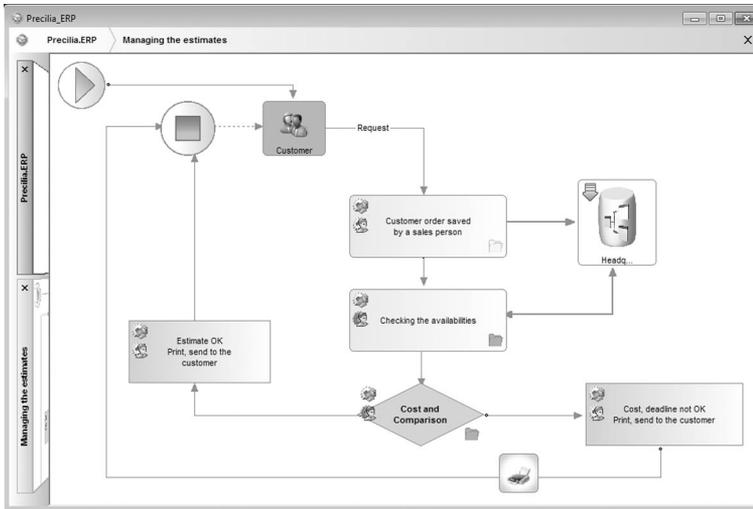
- ▶ To open the example project:
  1. Close the current project if necessary. The home window is displayed.
  2. In the home window, click "Tutorial" and select the project named "Flexible modeling". The project is loaded.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Flexible modeling".
- ▶ Open the "Precilia\_ERP.wem" file (double-click the modeling in the "Project explorer" pane). Several major fields are presented in this modeling: customer management, estimate management, marketing, ...

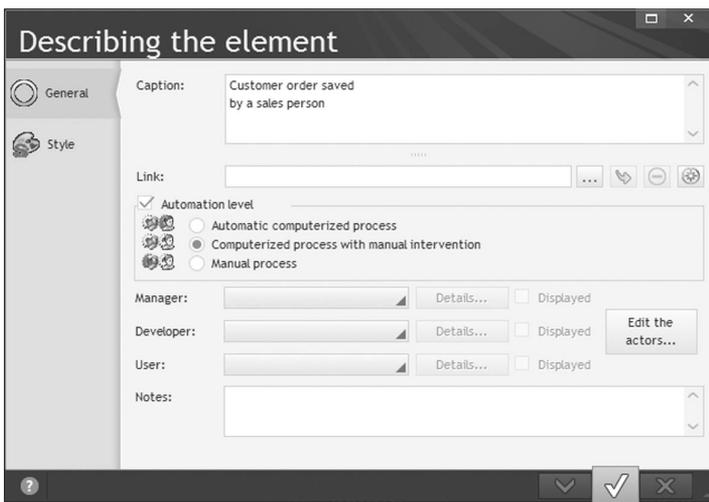
The icon  indicates the presence of a sub-modeling. Double-click this icon if you want to access this sub-modeling.



- ▶ In our example, open the sub-modeling corresponding to the management of estimates (double-click the folder). You will discover the different steps for managing the estimates:



- ▶ Let's take a closer look at the "Headquarter" element. An arrow is found in the top left corner of this element. This arrow indicates a link. The application analysis is automatically opened when you double-click the arrow.  
This is a main feature of the modeling editor: any element found in the modeling can be linked to an element of your project (window, code, ...).
- ▶ All the elements found in the modeling have their own characteristics. To display these characteristics, select "Description" from the popup menu of the element.  
For example:



This window is used to specify:

- the link between the element and an external file (analysis, window, ...)
- the automation level of the element: process performed automatically, manually, ...
- the actors taking part in this process (managers, users, developers, ...). This information can come from the contributors described in the Project Monitoring Center.

See the online help for more details.

## LESSON 9.12. MULTILINGUAL

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

---

- What is a multilingual site?
- Creating a multilingual site step by step.



Estimated time: 20 min

## What is a multilingual site?

---

A multilingual site is a site that can be run in different languages (English, French, German or any other language).

Therefore, the same site can be used in several languages. How is it possible?

That's what we shall see in this lesson.

We are going to handle a project that can be run in English or in French, according to the choice made by the Web user.

The main steps of a multilingual project are:

- Choosing the project languages.
- Localizing the analysis.
- Localizing the project elements (pages, reports, controls, ...).
- Localizing the messages found in the source code.
- Localizing the WLanguage functions.
- Programming the change of language in the site.

We are going to apply these different steps to the "WW\_Multilingual" project. This project, available in French, will be translated in English.

## Choosing the project languages

---

The first step consists in defining the languages supported by the project.

### Opening the example

► To open the example project:

1. Close the current project if necessary. The home window is displayed.
2. In the home window, click "Tutorial" and select the project named "Multilingual site". The project is loaded.

Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Multilingual site".

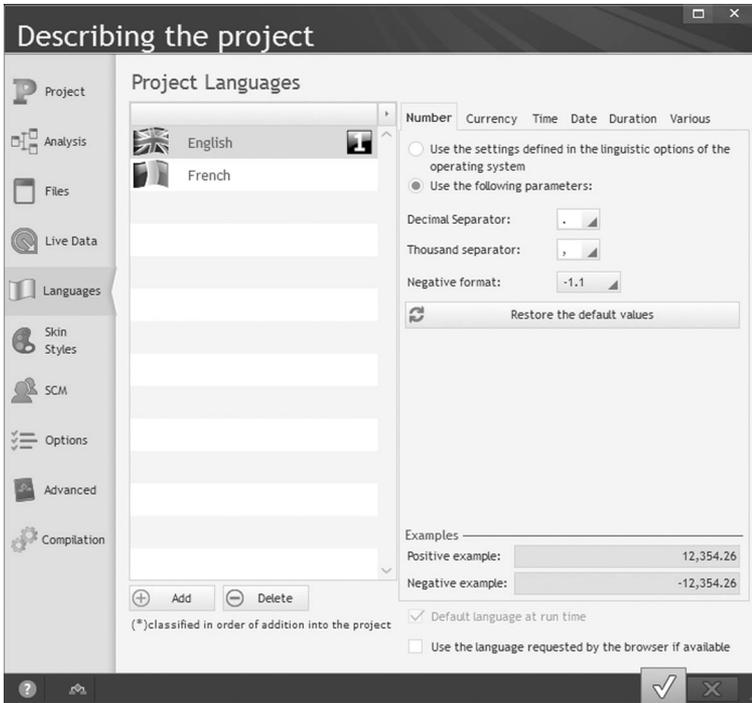
The associated analysis is "WW\_Multilingual". This information can be checked in the "Analysis" pane.

### Configuring the project

► The first operation consists in choosing the project languages.

1. Display the project description if not already done: on the "Project" pane, in the "Project" group, click "Description".
2. Click the "Languages" tab. Our application will support French and English. Check whether these 2 languages are displayed.

- ▶ We are going to configure the linguistic options of the project in this same window. These linguistic options affect the numbers, the currencies, the dates, ... for the selected language. Let's see an example:



1. Click the "English" language.
  2. Click "Date".
  3. The "Use the following parameters" option is selected by default. This options allows you to customize the date format used as well as the translation of days and months. If you choose "Use the parameters defined in the Windows options", the linguistic options of Windows for the current computer will be used.
  4. Keep the selected option.
- ▶ Validate.



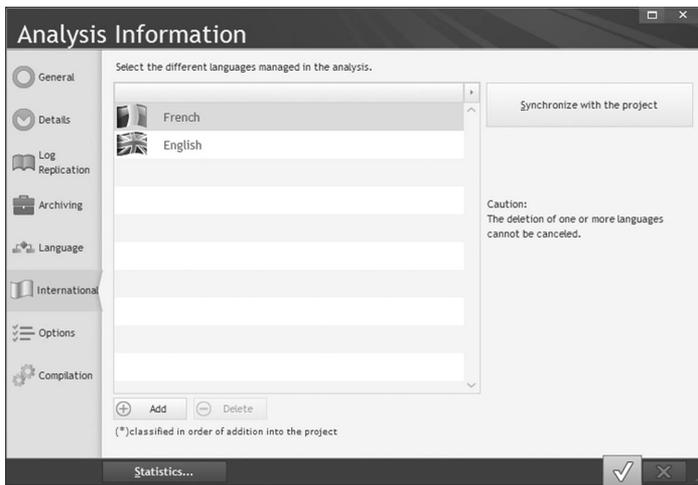
Notes

In the linguistic options, you have the ability to choose the text direction for the language ("Various" box, "Text direction" option). This allows you to create interfaces with a language written from right to left.

## Localizing the analysis

By default, an analysis is created in a language and it cannot be translated. However, some information can be entered in several languages (notes in the documentation, shared information, ...). By default, the controls created from the analysis items have the item caption specified in the analysis. If a caption was specified in the shared information of the item, this caption will be used when creating the control.

- ▶ To support several languages in an analysis:
  1. Display the data model editor: click  among the quick access buttons of the WebDev menu.
  2. In the analysis description ("Analysis description" from the popup menu), select the "International" tab.



3. Specify the languages supported by the analysis and validate.



Notes

The "Synchronize with the project" button is used to automatically retrieve the languages defined in the associated project.

- To enter the shared information in several languages:
1. Display the description of the items found in a data file ("Description of items" from the popup menu of the file).
  2. For each item, display the parameters of the control linked to the selected item (shared information). To do so, click the double arrow at the bottom of the screen.



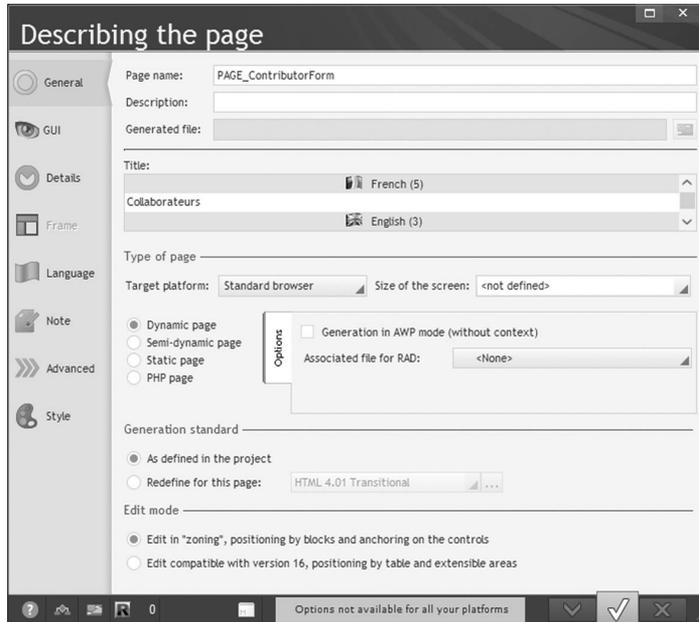
3. Validate the description of the file.
4. Close the data model editor.

## Localizing the project elements

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

Let's take a look at the characteristics of a page (and page controls).

- ▶ Open the "PAGE\_ContributorForm.WWH" page.
- ▶ Display the description of the page ("Description" from the popup menu of the page). Select the "Language" tab: the two languages selected in the project are displayed. Select the "General" tab: the title of the page must be translated.



Close this window.

- ▶ In the description of a page control, several elements must be taken into account during the translation:
  - The caption of the control (as already seen)
  - The image associated with the control. If this image contains a text, you must specify the images that will be used for each language (via ).
  - The alternative text ("Details" tab of the control description). This text is used if a display problem occurs (for an image for example).
  - The text of the tooltip ("Help" tab of the control description).
  - The information text displayed when the control is empty ("Content" tab of the edit controls).

The same type of information must also be translated for:

- all the controls found in the pages,
- the pages,
- the reports,
- the controls found in the reports,
- the text messages found in the code editor.

### How do I translate this information?

Several methods can be used to translate this information:

- a direct translation performed in the different editors.
- a translation performed via an external tool (WDMSG and WDTRAD)

### Direct input of translations

The translations are entered in the interface of the product. For example, the caption of the "Retour" link becomes "Back" in English. All you have to do is open the description window of the control and enter the corresponding translation in the requested language.

The translation can also be performed in the editor directly.

- ▶ To translate the elements of a page (including the menu):

1. Open the "PAGE\_ContributorForm" page.
2. On the "Display" pane, in the "Options" group, expand "Language displayed" and select the language to view in the editor.

The different controls and the menu options are displayed in the selected language. If no translation corresponds to the selected language, the captions of controls and menus are displayed in French.

3. Enter the captions in the selected language.

- ▶ 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 the translation.
- The supported languages.

4. Once the translation parameters are defined, you have the ability to use the button  to use the software defined for the translation.

## Special cases

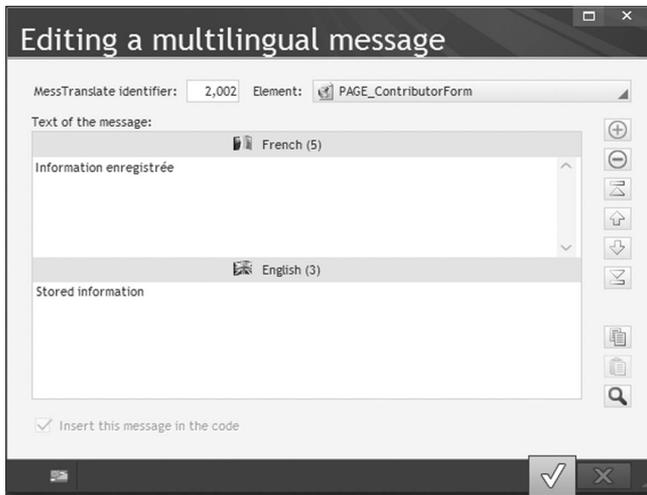
### Direct translation of programming messages

All the messages found in your program can also be entered in several languages.

| ■ Info("Stored information")

To translate this type of message, on the "Code" pane, in the "Languages" group, expand "Translate the strings" and select "Translate the messages".

The following window is displayed:



This window enables you to translate all the messages found in your program into all the project languages.

A number is assigned to each message.

When a translation is entered for the message, the icon  is displayed in the code editor.

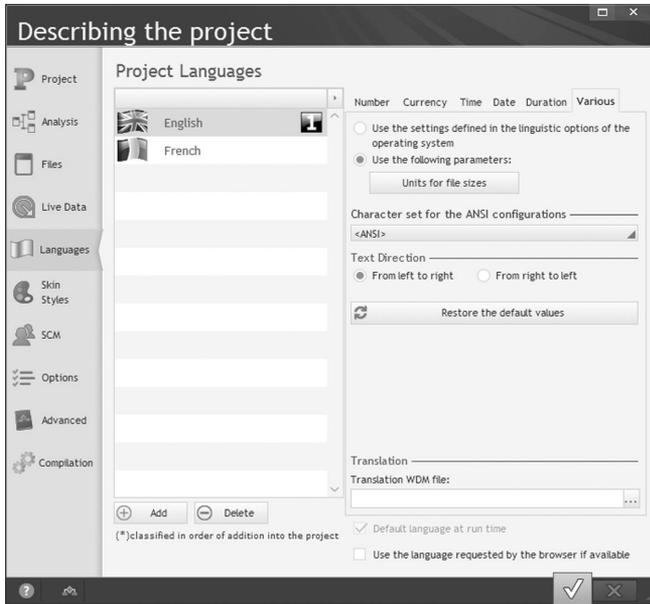
| ■ Info("Stored information" )

### Translating the result of WLanguage functions and the error messages of the framework

Various information and messages are found in the WebDev framework. For example, the names of the days and months used by the functions for date management come from the WebDev framework. To translate one or more libraries of this framework, you must use WDINT (not supplied with WebDev).

This program is used to get a file whose extension is WDM. To use this file in your project:

- you have the ability to use **LoadError**.
- the file can be included in the project description ("Languages" tab, then "Various" tab of the relevant language, "Translation WDM file" option).



Contact our Sales Department for more details about **WDINT**.

### Translation with WMSG and WTRAD

A tool named **WMSG** (not supplied with WebDev) is used to

- check out all the messages found in a project (caption of controls, code message, title of pages, ...) in order to translate them,
- check in the translated messages.

The messages to translate are checked out in text format, that can be configured to be used by the most common translation tools.

WMSG is also supplied with a tool for computer-assisted translation, WTRAD. WTRAD is used to enter all the translations for the multilingual information found in a project.

Contact our Sales Department for more details about **WMSG** and **WTRAD**.

## Programming the change of language

---

By default, the project is run in the first language defined for the project, in the "Languages" tab of the project description.

**Nation** is used to change the language.

In most cases, selecting a language in a site is performed via a drop-down list box or via a click performed on a button. As soon as a language is selected by the Web user, the site is loaded in the selected language.

- ▶ In the "PAGE\_ContributorForm" page, two links are used to modify the site language: "French" and "English". The code associated with the "French" link is as follows:

```
Nation (nationFrench)
ChangeCharset (charsetAnsi)
PageUse (CurrentPage ())
```

This process requires some explanations:

- **Nation** is used to modify the runtime language of the project.
  - **ChangeCharset** is required to manage the languages that use a specific character set.
  - **PageUse** is used to reload the page passed in parameter (**PageCurrent** is used to find out the name of the current page). The change of language performed by **Nation** is immediately taken into account.
- ▶ Run the site test and change the display language of the site.

## Managing the Unicode format and specific character sets

---

WebDev proposes two management modes:

- **the single-charset mode:** If your site supports languages that use specific character sets (Greek, Korean, ...), the character set used by all the screen fonts can be changed by programming (**ChangeCharset**). This character set can also be used for the data saved in the HFSQL files. See the online help (Keyword: multilingual) for more details.
- **the full support of Unicode:** This mode is used to manage the non-Latin character sets. In this mode, **ChangeCharset** is useless: the change of character set is automatically performed according to the language options defined in the project description. You also have the ability to mix different character sets.

The mode for managing the Unicode format is configured in the options of the current configuration ("Unicode" tab in the description window of the configuration).

**Note****Important**

If you choose to switch your projects to UNICODE, the ANSI/Unicode cohabitation generates constraints: the data must be translated when switching from an encoding system to the other one. The exchange of text strings must be adapted to take into account the new memory representation of the strings (in Unicode, 1 character is coded on 2 bytes while in ANSI, it is coded on 1 byte): calling APIs, reading or writing on disk to exchange data, reading or writing sockets containing strings, ...

If you choose to switch your projects to Unicode, a wizard measuring the impact of switching to Unicode is triggered on the project during the switch to Unicode.



**PART 10**

**Deploying  
sites**

**10**

**DEVELOP 10 TIMES FASTER**





# LESSON 10.1. DEPLOYING A DYNAMIC SITE

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

- Installing a WebDev Application Server - 10 connections.
- Creating the setup.
- Test of the site.



Estimated time: 20 min  
(time for installing the WebDev application server: about 30 min)

## Overview

---

When a WebDev site is developed, it must be deployed on a Web server in order to become available. That's what we are going to do.

Several methods can be used to deploy a dynamic WebDev site:

1. 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.
2. Remote deployment from the development computer (by FTP).
3. Remote deployment from a management computer (by FTP) via a "deployment package".
4. Deployment via the test hosting service of PC SOFT.
5. Deployment in PC SOFT Cloud.

When installing WebDev, you have the ability to install the test version of WebDev Deployment. This allows you to test the deployment of your dynamic sites.

To simplify the deployment operations and to allow you to directly test the deployment of your Web site, we will :

- Install WebDev Application Server (10 connections) on a Windows computer. This application server must be installed on a computer other than the development computer. WebDev Development must not be installed on this computer.
- Deploy the site remotely (by FTP).



Notes

If you already have the parameters for accessing the server on which your site will be deployed, there is no need to install WebDev Application Server (10 connections). The setup can be directly performed on the server by FTP.

## Required configuration

---

A specific configuration is required for the computer that hosts the test application server.

The following elements must be installed on the computer:

- a Web server (IIS or Apache)
- an FTP server
- the WebDev Application Server - 10 connections

See a specific documentation for more details.

## Installing the "WebDev Application Server - 10 connections"

The setup program of "WebDev Application Server - 10 connections" is available:

- from our Web site: [www.windev.com](http://www.windev.com)
- on the setup DVD of WebDev: start "Menu.exe" and select "Install a WebDev Application Server - 10 connections".

The setup steps are as follows:

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



Notes

You also have the ability to deploy on a Linux platform.

3. Select the setup path of the application server ("c:\WebDev 19" by default). Validate the creation of the directory.
4. Validate the next plane without modifying it.
5. Keep "Host the sites in earlier versions" and go to the next screen.
6. Validate the setup.

Note: The IIS Web server is automatically installed if no valid Web server is detected by the setup (in case of failure, the Apache Web server will be proposed for setup).

7. Select the virtual Web servers on which the WebDev application server must be installed. The default Web site is sufficient. Validate.
8. Keep the selected options and check "Start the WebDev 19 administrator". Validate.
9. The administrator is automatically started.



Notes

The PDF file named "WebDevDeployment.pdf" (accessible from the "Start" menu) is installed with the WebDev application server (10 connections). This file may help you solve the configuration problems linked to the access rights on the server.

## Configuring the WebDev account for deployment

In the WebDev administrator:

1. Select the "Setups/Accounts" tab.
2. Click the "Accounts" button. The WebDev account manager is started.
3. For the tests, we will be using the ADMIN account that is already created. In real use, we recommend that you to create specific accounts.
4. Select the "ADMIN" account and click "Edit".

5. In the "General" tab, specify the following values:

- Number of connections allowed for all the ADMIN sites: 50 (number greater than 0)
- Maximum number of sites allowed for ADMIN: 50
- Maximum number of Webservices allowed for ADMIN: 50
- Directory of FTP connection: Directory into which the FTP account will connect to transfer the site pages. Specify "C:\WB Tests\FTP" for example. Then, this directory will be declared on the FTP server.
- Home directory of the sites: Directory that will contain the site pages. Specify "C:\WB Tests\sites" for example
- Home directory of Webservices: Directory that will contain the Webservices of the site. Specify "C:\WB Tests\webservices" for example. "
- Home directory of the data files: Directory that will contain the data files of the site. Specify "C:\WB Tests\data" for example.

6. Validate and accept (if necessary) the creation of the different directories.

7. Close the account manager.

### Creating the FTP alias

The following operations explain how to create the FTP alias for IIS. The operations that must be performed for the other servers are presented in the online help.

To create the FTP alias:

1. Start the Internet service manager ("Start .. Run .. Inetmgr.exe" for IIS).
2. Select the default FTP site. Create a new virtual directory. The name of this directory is "TEST" and its path is the directory specified in the WebDev administrator ("C:\WB Tests\FTP" in our case). The virtual directory must have read/write rights.

### Creating a Windows user account

A specific Windows user account must be created on the deployment computer. This user account must have the same name as the alias that was created beforehand. In our case:

- the name will be: test
- the password will be: test

This user account must have the rights to write into the "Account" directory of the application server. This directory was installed in "C:\WebDev 19\Account" by default.

## Deploying your dynamic site

We are going to deploy the "WW\_Site\_With\_Data" project that was created in part 2 of this tutorial. If this project was not created, a corrected version is available.

- ▶ To open this project in WebDev:
  1. On the "Home" pane, in the "General" group, expand "Open" and select "Open a project".
  2. Select the "WW\_Site\_With\_Data" project.

If this application was not created, open the corrected project:

1. Close the current project if necessary to display the home window.
  2. In the home window, click "Tutorial" and select the "Site with data (Answer)" project.
- Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Site with data (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). To access it, on the "Project" pane, in the "Generation" group, expand "Deploy the site" and select "Deploy the site remotely".



- ▶ Display the next screen.

Before performing the setup, all the elements found in your site must be included in a library. A library is a file that groups all the elements created during the development steps (description of the database and pages, compiled source codes, ...). The HTML pages and the images are not included in the library.

Go to the next screen.
- ▶ You have the ability to include several languages in the library. In our example, we will keep the default options. Go to the next screen.

- ▶ 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 screen.
- ▶ Don't save the project and validate the library creation.

The setup wizard will now ask you some questions to define how your site will be deployed.



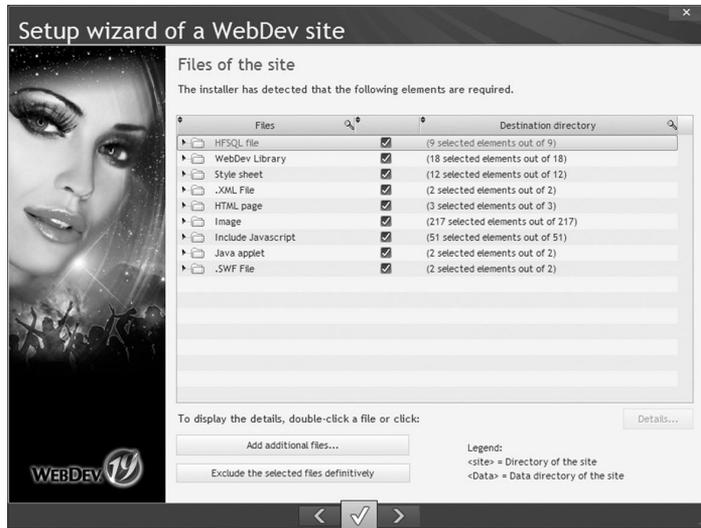
- ▶ In our case, we are going to perform a remote setup by FTP:
  1. Select the first option: "Deploy the WebDev site on a remote WebDev Application Server". Go to the next plane.
  2. To define the server parameters, the following information must be supplied by your hosting company. We are going to enter the information corresponding to the setup that was performed beforehand:
    - Address of the server (in our example, name of the computer where WebDev Application Server - 10 connections was installed).  
The name can be:
      - the name of a computer accessible via network ("TESTServ" for example),
      - an IP address (192.168.15.99 for example),
      - an Internet access (www.myserver.eu for example).
    - Characteristics of the user account entered by the hosting company in the WebDev account manager.  
In our example, the user name is ADMIN and the password (and its confirmation) is ADMIN.
    - Characteristics of the FTP account entered by the hosting company when creating an FTP account. In our example, the user name is "test" and the password (and its confirmation) is "test".



Notes

Caution: the user name can be preceded by the name of the domain to avoid confusions.  
For example: "mycomputer\test" or "mydomain\test"

3. Once the information regarding your WebDev account and your FTP account was successfully entered, go to the next screen.
4. Enter the parameters for deploying the site. We will keep the default options. Go to the next screen.  
WebDev establishes the connection and displays the list of files that will be transmitted to the server.



Go to the next screen.

5. 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 screen.

## 6. Specify the parameters of the site:



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 idle duration before disconnecting a user: this option is used to free all the resources occupied by the session of the Web user if no action was performed by this one since the specified duration.

By default, your site is enabled immediately after the setup. The users will have no access to your site if "Enable the site at the end of setup" is unchecked.

The wizard proposes to perform a setup:

- **immediate setup:** the files will be immediately transferred to the server and your site will be immediately installed.
- **delayed setup:** the files will be immediately transferred to the server but your site will be installed at the specified date and time ("Program the deployment for a later date").

7. Go to the next plane.

8. The wizard proposes to automatically generate statistical files for the site. These statistics describe the actions performed on the site, the origin of the Web users, ... Keep the default options and go to the next screen.

9. Display the next screen and validate the setup. The setup wizard transfers the files.

During the file transfer, the wizard compresses and encrypts the transferred data. Your data is transferred with a high-security level.

At the end of setup, a link allows you to immediately start the site.

Once the files have been installed, the wizard programs the activation of your site. If you have chosen an immediate setup, your site will be available in less than two minutes.

## LESSON 10.2. REMOTE MANAGEMENT

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

---

- Managing a dynamic WebDev site remotely.
- Using WAdminWeb190.



Estimated time: 10 min

## Overview

The WebDev application server proposes several tools for managing the dynamic sites, among which **WAdminWeb190**. This tool is a WebDev site included in the WebDev engine. It is used to manage a dynamic WebDev site remotely.

Caution: 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 will be used to update the parameters of your site.

**Notes**

To deploy a dynamic WebDev site, the WebDev application server must be installed on the servers.

► To start WAdminWeb190:

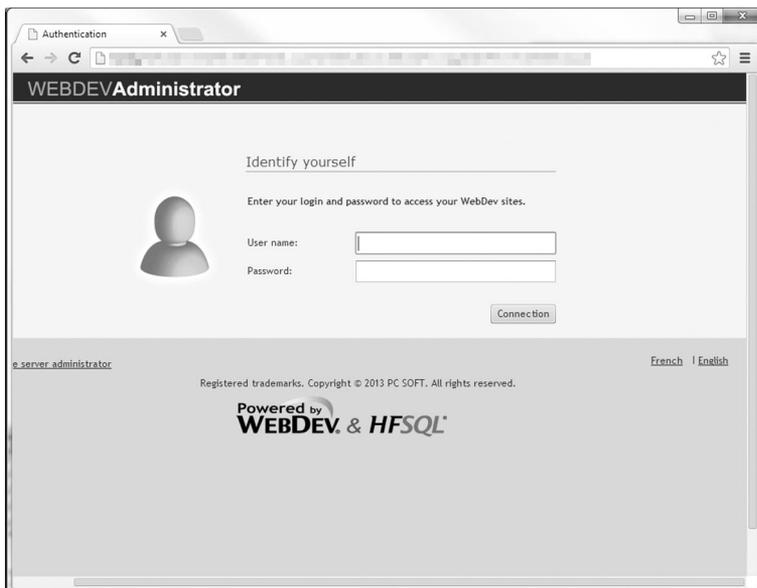
1. Start the WebDev administrator (WD190ADMIN) on the deployment computer.
2. Open your favorite browser (Internet Explorer for example) on your computer.
3. Enter the following URL in the address bar of the browser (while respecting the case):

```
http://computer/WD190AWP/WD190Awp.exe/CONNECT/WAdminWeb190
```

or

```
http://computer/WAdminWeb190
```

where "computer" is the name of the test deployment server (your computer for example).



4. Enter the login and password supplied by the site manager.
5. Click "Connection". The remote administrator is available.

## Using WAdminWeb190

WAdminWeb190 includes most of the options available in the WebDev administrator.

The screenshot shows the WEBDEV Administrator Configuration window. The window title is "Configuration" and the browser address bar shows "x". The main content area is titled "WEBDEV Administrator" and has a navigation menu with "Connections", "Sites", "Webservices", "Configuration", "Accounts", "Help", and "ADMIN".

**Maximum number of connections**

|   |                                  |
|---|----------------------------------|
| Maximum Number of Connections to Server:            | <input type="text" value="250"/> |
| Maximum number of connections to a site/webservice: | <input type="text" value="250"/> |
| Maximum number of connections per Web user:         | <input type="text" value="0"/>   |

**Managing the sessions/requests**

|   |                                       |
|---|---------------------------------------|
| Disconnect the inactive users since (HH:MM:SS): | <input type="text" value="00:50:00"/> |
| Maximum duration of a request (HH:MM:SS):       | <input type="text" value="00:50:00"/> |
| Duration of AWP contexts (HH:MM:SS):            | <input type="text" value="00:50:00"/> |
| Number of cached sessions:                      | <input type="text" value="0"/>        |

**Log file**

Notes:

- The directory of the log file will not be automatically created.
- The users who run the Web sites or the webservices must have sufficient permissions to write into the directory of the log file.

Generate a log file to manage traffic statistics (.log)

Directory:

Save in the global file and in the file of each application

**Remote debugging**

Allow the remote debugging

Main port:

Range of ports for sessions from:  at

See the online help for more details.

## LESSON 10.3. SITE STATISTICS

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

---

- Retrieving the statistics about a dynamic WebDev site.
- Using WStatistic.



Estimated time: 10 min

## Introduction

The traffic statistics are very important for a business site. They can be used to analyze the behavior of the Web users and to improve the navigation in the site.

WebDev includes an advanced tool for performing statistical calculations about the dynamic sites.

## Server configuration

On your deployment server (if your hosting company proposes to manage the statistics), WDStatistic is used to analyze the different connection logs created by the WebDev administrator.

In order for WDStatistic to operate properly, the deployment administrator must be configured as follows:

- to generate a log file for all the sites found on the server, check "Generate a log file...". This option is found in the "Configuration" tab of the WebDev administrator:

The screenshot shows a configuration dialog with a checked checkbox labeled "Generate a log file to manage traffic statistics (.log)". Below the checkbox is a text field labeled "Directory:" containing the path "D:\WebDev\Programs\Engine\Windows\Log". To the right of the text field is a small square button with three dots.

- to generate a log file for a specific WebDev site, check "Generate a log file...". This option is found in the "Sites" tab, "Custom parameters":

The screenshot shows a configuration dialog with a checked checkbox labeled "Generate a log file specific to this site to manage traffic statistics (.log)". Below the checkbox is a text field labeled "Directory:" containing the path "F:\WebDep\Data\My\_Library\Log". To the right of the text field is a small square button with three dots.

Note: you also have the ability to configure a site to create a specific log file from the remote administrator.

- ▶ We are going to create a log file for one of the dynamic sites found in this tutorial (the "WW\_Site\_With\_Data" project for example).
  1. Start the WebDev administrator if necessary: on the "Tools" pane, in the "Web utilities" group, click "WAdmin".
  2. In the "Sites" tab, select the project for which the log file must be generated ("WW\_Site\_With\_Data" for example).
  3. Select "Custom parameters".
  4. Select "Generate a log file specific to this site..." and specify the directory for creating the log file (you can keep the default directory).
  5. Click the "Apply" button and validate the directory creation if necessary.
- ▶ To run the test of the "WW\_Site\_With\_Data" site in WebDev and to create the log file:
  1. Open (if necessary) the "WW\_Site\_With\_Data" project in the editor: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Site with data (Answer)".
  2. Run the test of the project.
  3. Handle the different pages of the site.

## Using WStatistic

### Starting WStatistic

On your computer (development computer for example), WStatistic is available in the "\Programs" directory of the setup directory of WebDev ("WStatistic.exe" file). To start WStatistic from WebDev, on the "Tools" pane, in the "Web utilities" group, click "WStatistic".

- ▶ Start WStatistic.

### Setting

We are going to perform a first setting in the "Site management" section.

1. Click "Site management". Validate the warning message if necessary.
2. In the "Site management" window, click the "Logs" tab.

First of all, the location of the log files must be specified to WStatistic.



Notes

In our example, a manual import will be performed (the log files are found on the development computer).

To systematically use WStatistic, we advise you to perform an automatic import. Therefore, the next logs will be automatically retrieved by WStatistic.

See the online help about WStatistic for more details.

3. Select the type of connection ("Directory" in our case), the start date for importing the logs. In the "Directory" control, specify the path of the logs (specified in the WebDev administrator beforehand).
4. Validate.

- ▶ To import the log files:

1. Click the "Import" button found in the main screen of WStatistic. A selection window is displayed. Click "Start an import now".



Notes

Caution: you cannot import the statistics about the operations performed today.

2. At the end of import, close the import window ("Close" button).
3. Validate the window for importing the logs.
4. The statistical dashboard is displayed in the main window of WStatistic. You have the ability to select the type of statistics to display.

Note: in our test, the statistics are calculated for a single connexion, at a given time. At run time, all the connections to the site are taken into account.

# LESSON 10.4. WINDEV BACK OFFICE

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

- Using a WinDev application in Back Office.



Estimated time: 15 min

## Overview

---

A business site can allow the Web users to place orders, make bookings, buy products, request information about the company, ...

All the details entered by the Web user are saved in a HFSQL data file in order for these details to be retrieved by a Back Office application. This application will operate on the data files without disturbing the execution of the site.

This Back Office application can be developed with WinDev. Therefore, the data will be compatible and there will be no need for conversion. Furthermore, no matter whether the server is found in your premises or at the hosting company, WinDev allows you to easily develop a Back Office application for your WebDev site! Both products have the same operating mode, so they get on very well together ...

Then, WLanguage (that is fully compatible between WinDev and WebDev) allows you to transfer the long processes previously performed by a WebDev site to a WinDev application, regardless of the Internet glitches!

Some examples of Back Office applications developed with WinDev and used with a WebDev "Front Office" site:

- automated mailshot (WinDev), sales and bargains (WebDev),
- transfer the orders to a company HQ (WinDev), online purchase (WebDev),
- monitoring of industrial equipment (WinDev), sales report via Extranet or Intranet (WebDev),
- banking application (WinDev), viewing accounts and investments (WebDev),
- ...



Notes

If you do not own WinDev, contact PC SOFT sales department to get a free brochure presenting WinDev 19 or visit the Internet site of PC SOFT: <http://www.windev.com!>



# CONCLUSION

**The tutorial is over now!**

This tutorial has discussed a variety of subjects, but not all the features of WebDev, far from it! You are now familiar with the main concepts.

We recommend that you spend another day exploring the different menu choices of WebDev, for all the modules.

You can also explore the examples supplied with WinDev: some of them are simple and address a single topic while others are more complex. These examples illustrate the different aspects of WebDev. Reading the source code is also a good way to learn.

It would take too much room to discuss all the available topics (there are hundreds, even thousands!). WebDev proposes several other features not presented in this lesson:

- sockets, HTTP and telephony functions
- creation of skin templates ...
- nested reports, queries with parameters ...
- dynamic compilation, calls to DLL, external languages ...
- Vista gadgets, JSON, ...

See the online help for more details.

We wish you great development experiences with **WebDev!**

