
DEVELOP 100 TIMES FASTER.
PC SOFT

WebDev 17.0
Documentation version 17.0 - 0312

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

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

This documentation is not contractually binding. 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 2012: This publication may not be reproduced in part or in whole in any form without the express permission of PC SOFT.

TABLE OF CONTENTS

Introduction

Preliminary talks **19**
Overview of the tutorial **19**
How do I access the online help? **20**
Legend of the symbols **20**
If you are familiar with WebDev 16... ..... **20**
What is WebDev used for? **21**

PART 1 - DISCOVERING WEBDEV

Lesson 1.1. Concepts

Overview **26**
Main concepts **26**
Terminology **27**
In the analysis 27
In the pages and reports 28
The different types of pages proposed by WebDev **29**
The different types of pages 29
The different types of sites 32
The different browsers **32**
The different Web servers **33**
What's happening on the browser side **33**
What's happening on the server side **34**

Lesson 1.2. Environment required by the Tutorial

Overview **36**
Implementation **36**

Lesson 1.3. My first pages

Overview **38**
My first page: entering information **38**
Overview 38
Example project 39
Displaying a formatted text 42
Entering information 44
Button for validating a page 45
Test of a page 49
Anchoring the controls 50
Checking the information entered 51

My second page: Displaying the information entered **52**
Creating the page 52
Creating a static control 52
Managing the parameters 53
Creating the Popup page 54
Displaying the Popup page 56
Test of the site 57

Lesson 1.4. Programming concepts

Introduction **59**
Server code and browser code **59**
Server code 60
Browser code 60
Declaring the different types of variables **61**
Declaring the variables and their scope 61
Main statements of WLanguage **62**
Conditional statements (server code and browser code) 63
Loop statements (server code and browser code) 63
Comments 64
The MyPage keyword (server code) 64
Procedures and functions **65**
Definition 65
Local procedure 65
Global procedure and set of procedures 65
How do I decide whether a procedure must be global or local? 65
About passing parameters 66
Calling a procedure 66
Creating a procedure 66
A practical example on the procedures and the functions 67
Processing the strings (server code and browser code) **67**
Practical example 67
Details (server code and browser code) 68
Processing the numeric values (server code and browser code) **70**
Practical example 70
Details 70
Processing the currencies (server code only) **71**
Practical example 71
Details 71
Processing the dates and times (server code and browser code) **73**
Overview 73
Practical example 73
Input mask and returned value 73
The dates (server code and browser code) **74**
Practical example 74
What is today's date? 74
What is today's date, with the day and the month in letters? 75
How many days between two dates? 75
What is the day of the week corresponding to a given date? 75
Practical exercise 75

The times (server code and browser code) **76**
Practical example 76
What time is it? 76
How much time has passed between two given times? 76
Date, Time, DateTime and Duration variables (server code) 77
Calculations with dates and times 78

PART 2 - SITE WITH DATA

Lesson 2.1. Overview

Overview of the site created in this part **82**

Lesson 2.2. Project and analysis

Overview **84**
Creating the project **84**
Creating the analysis **85**
Creating the description of the data files **87**
Creating a data file and its items in the editor 87
Importing a CSV file 91
Importing files from the dictionary 94
Direct import of existing data files 95
Creating the links **95**
Generating the analysis **97**
Configuring the analysis for RAD **98**
Configuring the RAD in the description of the data files 98
Configuring the RAD in the description of items 99

Lesson 2.3. The full RAD

What is RAD? **102**
Generating RAD **102**
Test of the site **104**
Customizing the generated site **105**

Lesson 2.4. Edit page

Overview **107**
Creating an edit page **107**
Managing the addition of a record **110**
Creating the add button 110
Adding an input mask 111
Erasing the data after the addition 112
Viewing the records 112
Managing the upload of the book cover 113
Closing button 114

Managing the search and the modification **115**
Exact-match search 115
Generic search 116
Modifying the form displayed 117
Browsing the forms 117

Lesson 2.5. Multi-criteria search

Overview **120**
Creating a query with parameters **120**
Creating a page with a table based on the query **122**

Lesson 2.6. Printing the data

Overview **126**
Creating a report **126**
Starting the report print **127**

Lesson 2.7. The user groupware

Overview **130**
Including the user groupware **130**
Configuring the user groupware **131**

Lesson 2.8. The dashboard

Overview **134**
Automatic tests **134**

Lesson 2.9. Reference a site

Overview **138**
Adding a home page **138**

PART 3 - PAGES AND CONTROLS

Lesson 3.1. The editors

The environment **144**
The panes **144**
The different panes 144
Handling the panes 145
The environment configurations 145
The custom menu **145**
Customizing the wizards **147**

Lesson 3.2. Pages

How do I create a page? **149**
Description of a page: 7 tabs are available **149**
Practical example 150
"General" tab 150
"GUI" tab 151
"Details" tab 151
"Frame" tab 152
"Language" tab 152
"Note" tab 152
"Advanced" tab 152
"Style" tab 152
Processes associated with a page **153**
Global declarations (dynamic pages and PHP pages only) 153
Initialization (dynamic page and PHP page only) 153
Load (dynamic and PHP page, semi-dynamic and static) 153
Unload (dynamic and PHP page, semi-dynamic and static) 153
Synchronization (dynamic page only) 153
Closing (dynamic page only) 153

Lesson 3.3. The controls

Introduction **155**
Practical example 155
The standard controls **155**
Type of control: Static 156
Type of control: Edit control 157
Type of control: Button 160
Type of control: Link 162
Type of control: Image 163
Type of control: Clickable image 167
Type of control: Radio button 169
Type of control: Check box 170
Type of control: List box 171
Type of control: Combo box 176
Type of control: Table 177
Type of control: Looper 180
Type of control: TreeView 181
Type of control: Cell 182
Type of control: Chart 183
Type of control: Rating 184
Specific controls **184**
Type of control: Calendar 185
Type of control: Scheduler 185
Type of control: Captcha 185
Type of control: Bar Code 186
Type of control: HTML table 186
Type of control: Horizontal rule 186
Type of control: HTML 187
Type of control: Web Camera 187

Type of control: Java Applet 187
Type of control: Flash and Flex 188
Type of control: Silverlight 190
Type of control: IFrame 190
Type of control: Site map path 190
Type of control: Site map 191
Type of control: Pager 191
Type of control: Thumbnail 192

Lesson 3.4. Ergonomics of the site

Overview 194
Handling the styles of the controls 194
Defining the tab order of the controls 195
Displaying a help in the pages 195
The tooltip 196
Help message 196
Indication text 196
Create your pages by using the zoning mode 196
Anchor the controls 198
Use the rulers and the alignment options 198
Use the tabs to group your controls 199
Improve the navigation in your site 200
Use modern dialog boxes 200
Interface of the site: Use the page templates 201
What is a page template? 201
How do I create a page template? 202
How do I apply a page template? 202
Updating the template 202
How do I dissociate a template imported into a page? 203

Lesson 3.5. Reusability

What is the reusability? 205
Re-using a set of controls 205
The supercontrol 205
Internal page 206
Control template 207

Lesson 3.6. Questions/Answers

How do I display a page in a new window of the browser? 209
By programming 209
By using the page editor 209
How do I modify the color of a static by programming? 210
How do I make a button invisible? 211
How do I create a vertical menu in a WebDev page? 211
How do I standardize the appearance of my pages? 211
How do I add a popup menu to a table or to a looper? 212
How do I pass parameters to a page? 212
How do I group the controls in order to modify their properties by programming? 212
How do I transform a check box into a radio button? 213

How do I align the controls? 213
How do I give the same size to the buttons? 214
How do I add an image to the page background? 214
How do I display the date of site update? 214

PART 4 - DATABASES AND ANALYSES

Lesson 4.1. Introduction

Overview 218
The different modes for accessing the databases 218
Native Access 218
Direct ODBC access 219
OLE DB access 219
ODBC access via OLE DB 219
Which access should I use? 219

Lesson 4.2. Browsing the data files and the queries

HReadFirst browse 221
HReadSeek browse 221
FOR EACH browse 222
FOR EACH WITH browse 222
What type of browsing command should I choose? 223

Lesson 4.3. Managing the errors

Overview 225
Managing the errors in automatic mode 225
Managing the errors in advanced programmed mode 227

Lesson 4.4. The log process

What is the log process? 230
What is the log process used for? 230
How do I implement the log process? 230
How does the log process physically operate? 231
How do I successfully implement the log process? 232

Lesson 4.5. The transactions

What is a transaction? 234
A simple example of the need for transactions in some types of processes 234
The transactions on HyperFileSQL 235
Using the transactions 235
Implementing the management of transactions 235
Handling the records during a transaction: the rules to follow 236
What should I do if a transaction is interrupted? 236

Lesson 4.6. Data encryption

What is data encryption used for? 238
How do I encrypt the data files? 238
Implementing the encryption of the data files 238
Managing the encryption in WLanguage 240

Lesson 4.7. The replication

What is the data replication used for? 242
Implementing the data replication 242
Implementation 243
1. Activation 243
2. Declaring the master database 243
3. Declaring the subscriber databases 243
4. Replication 243

Lesson 4.8. The triggers

What is a trigger used for? 246
Definition of a trigger 246
Implementation 246
Step 1: describing the trigger (in the initialization code of the project) 246
Step 2: creating a global procedure linked to the trigger 247

Lesson 4.9. External databases

Overview 249
Connecting to a database in the data model editor 249
Connecting to a database by programming 251
Creating the connection: HOpenConnection 252
Associating the connection with the data files: HChangeConnection 252

Part 5 - Managing a HyperFileSQL Client/Server database

Lesson 5.1. Introduction

Overview 256
Why switch a site to HyperFileSQL Client/Server? 257

Lesson 5.2. Implementing a HyperFileSQL Client/Server site

Overview 259
Installing a local HyperFileSQL server 259
Creating a site in HyperFileSQL Client/Server mode 259
Migrating a site from HyperFileSQL Classic to HyperFileSQL Client/Server 260
Overview 260
Migrating the example 260
Features available in HyperFileSQL Client/Server mode 262

Lesson 5.3. Managing a Client/Server database

Overview 264
Configuring the computers 264
The HyperFileSQL Control Center 264
Creating a user account in the HyperFileSQL Control Center 266
Saving the database 269
Conclusion 269

Part 6 - Queries

Lesson 6.1. Create your first query

Introduction 274
Your first query 274

Lesson 6.2. Queries with count

Creating a query with count 281

Lesson 6.3. Sum queries

Creating a "Sum" query 284

Lesson 6.4. The SQL language

Source code of a query 287
The SQL commands 287
The SELECT statement 287
The FROM statement 288
The WHERE statement 288
The GROUP BY statement 288
The ORDER BY statement 288

Lesson 6.5. Using queries

Using a query from a report 290
Using a query from a table or from a looper 290
Using a query from WLanguage 290

Part 7 - Reports

Lesson 7.1. Introduction

Principles for printing on Internet 294
Direct print 294
Generation of documents (HTML, PDF, XML), ... 294
Basic vocabulary 294

Lesson 7.2. Your first report

Overview **298**
The reports to create 298
Creating your first report **299**
Creating the report: List of customers sorted by city 299

Lesson 7.3. Statistical reports

Overview **307**
Step by step **308**

Lesson 7.4. Crosstab reports

Overview **314**
Step by step **314**

Lesson 7.5. Running reports

Running reports in WLanguage **319**
iDestination function 320
iPrintReport function 320

PART 8 - ADVANCED PROGRAMMING

Lesson 8.1. Web features

Overview **324**
The cookies **324**
What is a cookie made of? 324
Managing the cookies with WebDev 325
Practical example 325
Uploading files **326**
Managing the upload with a single-file upload control 326
Practical example 326
The file download **327**
Managing the download 327
Practical example 328
Securing the pages **328**
Protecting the access to your pages by a password 328
Securing the data and the pages via TSL/SSL **331**
Implementing secured transactions via the TLS/SSL protocol 331
Transactions secured by TLS\SSL in a WebDev site 331
Secured payment **332**
System for secured payment in a WebDev site 333

Lesson 8.2. The timers

Introduction **335**
Programming the timers **335**
Practical example 335
Creating a timer 335
Stopping a timer 335

Lesson 8.3. Automatic management of errors

Overview **337**
Operating mode 337
Implementation 337
Types of errors 337
Using the automatic management of errors **338**

Lesson 8.4. The indirections

Overview **340**
Example **340**

Lesson 8.5. The management of emails

Overview **342**
The POP3 and SMTP protocols **342**
Starting an email session **343**
Sending an email **343**
Reading an email **343**
Disconnection **344**
Other possibilities **344**

Lesson 8.6. OOP

Concepts **346**
The classes 346
The objects 346
The members 346
The methods 346
Concept of inheritance 346
Constructor and Destructor 346
Data encapsulation 347
Creating an object-oriented program 347
Declaring a class 347
Describing the methods 347
Declaring and handling objects 348

PART 9 - ADVANCED PROJECT MANAGEMENT

Lesson 9.1. Required configuration

Overview **352**
Implementation **352**

Lesson 9.2. Dashboard

Overview **354**
Example 354
The different elements of the dashboard **354**
The lists 355
The buttons 356
The counters in relation with the Control Centers 357
Configuring the dashboard **358**

Lesson 9.3. SCM

Introduction **361**
SCM (Source Code Manager) **361**
Principle of SCM 361
Creating the SCM database 362
Including a project in the SCM 362
Opening a project from the SCM 365
Configuring the SCM 365
Checking out an element 367
Modifying the checked-out element 368
Checking in the checked-out element 368
Synchronizing the project 369
Off-line mode (or mobile mode) 369
SCM administrator 370

Lesson 9.4. Control Centers

Overview **372**
The Project Monitoring Center **372**
Managing your time **374**
The other Control Centers **375**

Lesson 9.5. Business rules

Overview **377**
Example 377
Application on a real example **377**
Creating a business rule 377

Lesson 9.6. The external components

Overview **381**
Teamwork 381
Large projects 381
The databases accessed by several projects 381
The processes used in several projects 382
The ability to distribute a feature or set of features 382
Multi-product component 382
Step by step **382**
Step 1: Creating an external component 382
Step 2: Using the component 386
Distributing a component **387**
Standard distribution 387
Professional distribution 388

Lesson 9.7. Multi-configuration

Overview **390**
Creating a project configuration **390**
Managing the project configurations **391**
Multiple generation **391**

Lesson 9.8. The debugger

Overview **393**
Debugging a page **393**
Starting the debugger 393
Viewing the information in the debugger 394
Trace window of the debugger **395**
Debugging a project **395**

Lesson 9.9. The performance profiler

Overview **397**
Using the performance profiler **397**
Optimizing a process with the performance profiler **398**
Reading the result of the performance profiler 398
Choosing a process to optimize 398
Optimizing a process 398

Lesson 9.10. Import/Export

Importing elements **401**
Exporting elements **402**
Specific import operations **403**
Importing a WinDev project 403
Importing an HTML page 403

Lesson 9.11. UML and 3-tier

What is the UML? 406
Diagrams managed by WebDev 406
Class diagram 406
Use case diagram 406
Object diagram 406
Component diagram 406
Activity diagram 406
Sequence diagram 406
Collaboration diagram 406
State-transition diagram 407
Deployment diagram 407
3-tier 407
What the 3-tier is? 407
How do I implement the 3-tier? 407
Handling the different layers 408

Lesson 9.12. Flexible modeling

Principle 410
Operations 410

Lesson 9.13. Multilingual

What is a multilingual site? 414
Choosing the project languages 414
Localizing the analysis 415
Localizing the project elements 417
How do I translate this information? 418
Direct input of the translations 418
Special cases 419
Translation with WDMSG and WDTRAD 420
Programming the change of language 420
Managing the Unicode and the specific character sets 421

PART 10 - DEPLOYING SITES

Lesson 10.1. Deploying a dynamic site

Overview 426
Required configuration 426
Installing the "WebDev Application Server - 10 connections" 426
Configuring the WebDev account for deployment 427
Creating a Windows user account 428
Deploying your dynamic site 428
Preparing the setup 428

Lesson 10.2. Management remote

Overview 434
Using WDAdminWeb170 435

Lesson 10.3. Site statistics

Introduction 437
Configuring the server 437
Using WDStatistic 438
Starting WDStatistic 438
Setting 438

Lesson 10.4. WinDev Back Office

Overview 440

Conclusion

INTRODUCTION

Preliminary talks

Caution: this manual is a tutorial. You should have the programming Guide next to you when you are using WebDev. You can also consult the online help of 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 course and to learn WebDev: you'll find it well worth it!

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

This course 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 course 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.

The language aspect is only one of the many aspects of development. Programming is a lot easier if all the aspects of development are taken into account.

Overview of the tutorial

The tutorial has been designed to progressively teach you how to use WebDev. By following this course:

- you will discover the main concepts explained here informally; these are the concepts you need to learn and understand.
- you will also be asked to perform tasks that demonstrate 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 (directly 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 forget to consult the online version of the Tutorial (PDF file accessible from "? .. Tutorial .. Tutorial (PDF)".

How do I access the online help?

1. In the editors, press the [F1] key.
2. In the editors, the help menu (symbolized by "?") enables you to display the help summary or to search for specific information.
3. Furthermore, from the code editor, a specific help can be directly accessed for each selected function via the [F1] key.

Legend of the symbols


This symbol indicates the duration of the lesson. Please note that the actual time may vary according to your level of experience.


An example is available to complement the lesson.


This symbol introduces a "Tip", we 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 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 16...

If you are familiar with WebDev, following this course 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 enables you to develop Internet and Intranet sites in many fields:

- E-commerce (sales, rentals, bookings, ...)
- Multimedia (description of companies, "showroom" sites, ...)
- Intranet (logins, secured 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, you don't need to find and add modules to be able to design, test and online (or "deploy") a site.

The WebDev 5GL (5th Generation Language), the 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!

The WLanguage, available in English and in French, allows you so save a lot of time!


PART 1
Discovering
WebDev
DEVELOP 10 TIMES FASTER
PC SOFT

LESSON 1.1. CONCEPTS

This lesson will teach you the following concepts ...

- The main concepts of WebDev
- The vocabulary used in WebDev



Estimated time: 10 min

Overview

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

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

WebDev enables you to create:

- static Internet/Intranet sites. These sites manage data that does not change (corporate sites, sites available on a CD-ROM, ...).
- 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.

WebDev includes a powerful HyperFileSQL database engine. The HyperFileSQL engine is available in network version and in Client/Server version.

Several other databases can be used but we recommend that you use HyperFileSQL for your sites in order to optimize the processes of the data files.

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.

Before we start, let's take a look at the basic information: the concepts and the vocabulary used by WebDev.

Main concepts

WebDev enables 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.

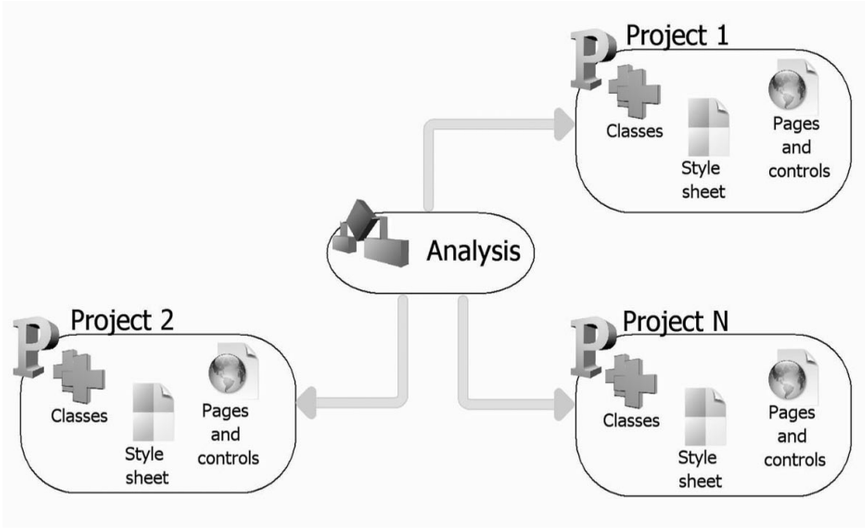
If your site handles data, WebDev allows you to define the structure of the database via **the analysis**. The WebDev analysis contains the description of the data files (also called "Tables" in several databases). These files will contain the data of the application.



Notes

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

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 vocabulary used in WebDev. Indeed, several terms are specific to WebDev and they may differ from the ones used in other tools.

In the analysis

File: The analysis is used to describe the structure of the files found in the database. 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. The record of a data file corresponds to all the items defined for the file.

Item: In the analysis, an item corresponds to an area 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 HyperFileSQL 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 browses performed on the data files. In WebDev, if a HyperFileSQL data file contains several key items, a single index file will be created at run time.

In the pages and reports

Page: The pages are used to display or enter information. The Web user can 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, text files, 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, and so on).

Style: The style groups the graphic characteristics of an element: background image, border, font, ... Styles of the different elements found in the interface of a WebDev site are grouped in a style sheet.



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

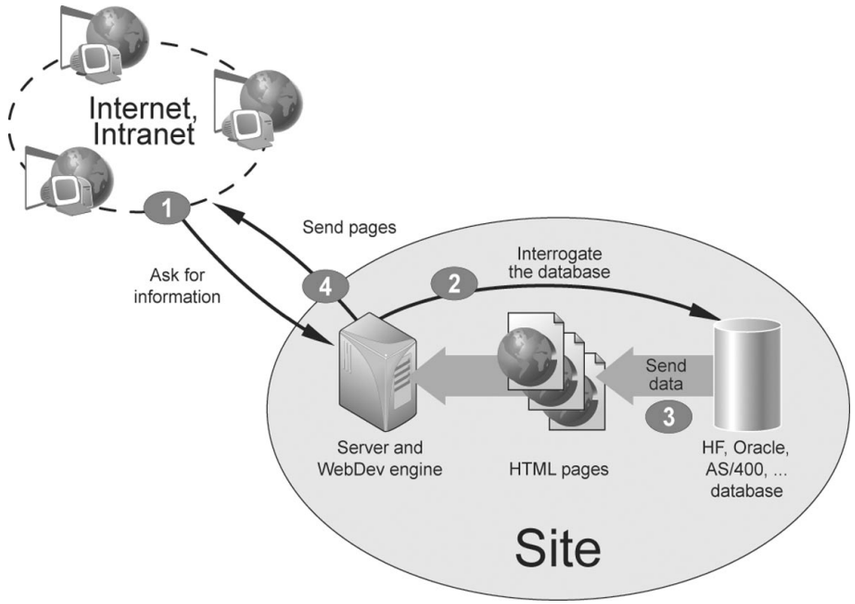
The different types of pages proposed by WebDev

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 the Internet (online order, ...), ... choose dynamic pages.



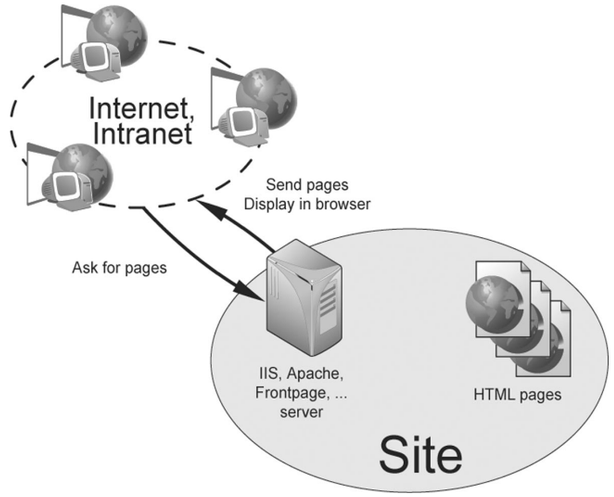
A dynamic page is used to display data coming from a database, as well as images and interactive text.

For example, to display some 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.

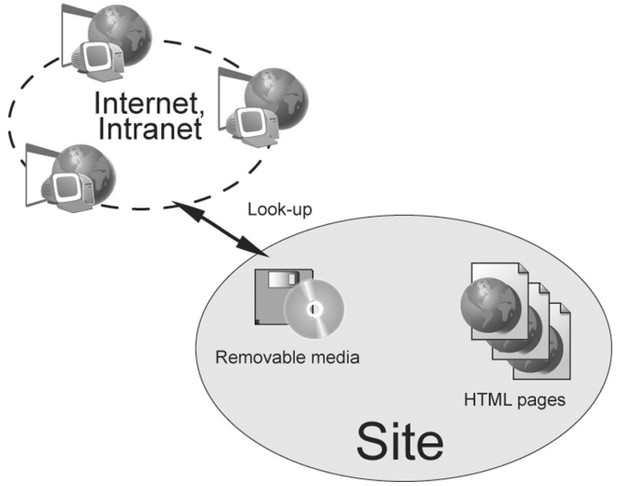
Static pages

When your site only needs to display fixed text and images, choose static pages. These static pages can be:

1. Hosted on a server



2. Saved on a removable media



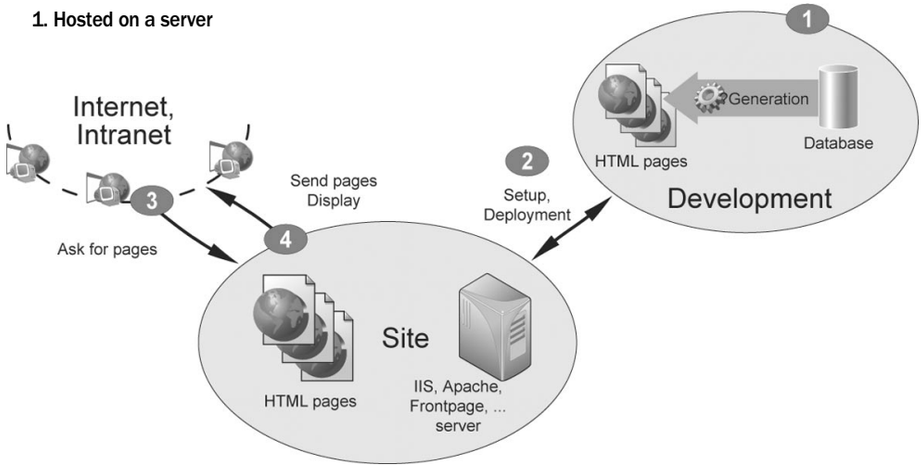
A static page is used to display text and images but the Web user cannot "interact" with them. To display information about several products, you will have to create as many static pages as the number of products.

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

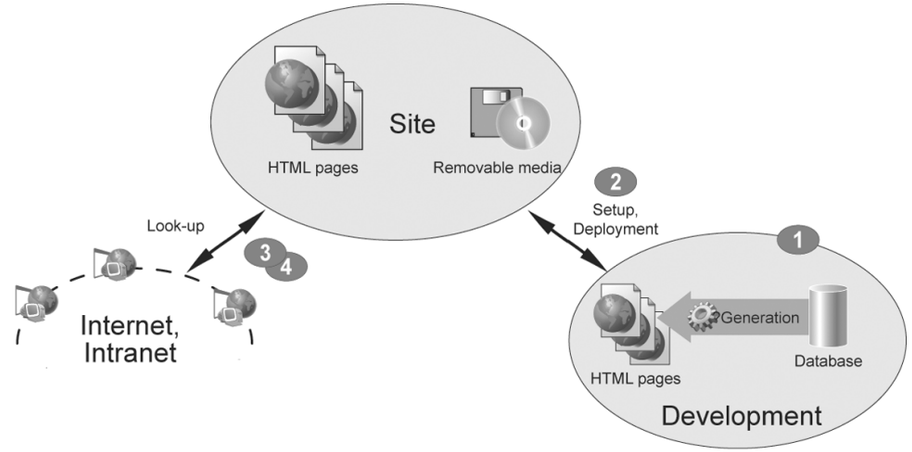
Semi-dynamic pages

Many companies propose their product catalog on CD-ROM. It may be interesting to choose an Internet site. Indeed, an Internet site on CD-ROM 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 (generate online orders, create 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-ROMs.

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 the site was developed. 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-ROMs.

The different browsers

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

There are a lot of different browsers allowing you to access the Internet depending on your operating system. The most common are:

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

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

WebDev enables 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 use "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),
- Omni HTTPD,
- FrontPage 97 server (version 2.0),
- FrontPage 98 server (version 3.0),
- Apache (versions 1.x) for Windows,
- Apache2 (versions 2.x) for Windows,
- Netscape Enterprise Server.

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 from 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" are programmed in WLanguage when developing your site (orange bars). The server actions require a WebDev runtime engine to be installed 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 1.2. ENVIRONMENT REQUIRED BY THE TUTORIAL

This lesson will teach you the following concepts ...

- **Starting WebDev**
- **Configuring WebDev to follow the Tutorial**



Estimated time: 5 min

Overview

WebDev allows you to configure your environment. Several modes are available:

- **Simplified environment**: This mode enables you to discover the main features of WebDev.
- **Full environment**: This mode proposes all the features of WebDev.
- **Retrieve the configuration of your environment XX**: This mode uses the features available in version XX installed on your computer (where XX corresponds to the WebDev version: 12, 14, 15, 16, ...).

At any time, regardless of the type of environment used, you have the ability to add or delete the access to some unused features.

To follow this Tutorial, we recommend that you work with a simplified environment. The advanced features will be added as this Tutorial goes along.

Implementation

- To use the simplified environment of WebDev:
  1. Start WebDev 17.
  2. A wizard starts if WebDev 17 was never started before. This wizard enables you to choose your work environment.
  3. Select "Simplified environment" and validate.
- This wizard is not displayed if WebDev was already started on your computer. To check and modify (if necessary) the configuration of your environment, perform the following operations:
  1. Select "Tools .. Options .. Options of the environment".
  2. Click "Restart the wizard for configuring the environment".
  3. Select "Simplified environment".
  4. Validate your choice.
  5. Validate the options of the environment.

That's it, WebDev is configured to follow the Tutorial.

LESSON 1.3. MY FIRST PAGES

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. These examples will enable you to understand how Internet sites operate and to get familiar with the programming concepts used in WebDev. 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 the dates and times.

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

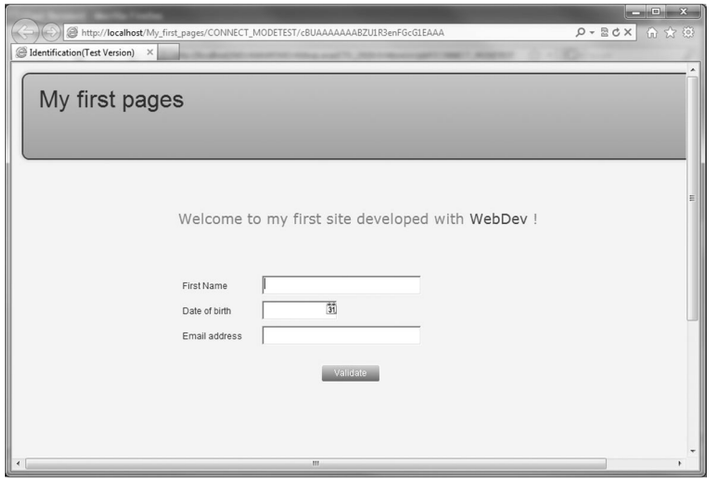
- Identification page.
- Home page.

For practical reasons, these pages will be grouped in a project.

My first page: entering information

Overview

You are going to create the following page:



This page allows the Web user to identify himself and 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.

Example project

- Start WebDev 17 (if not already done).
- Close the current project if necessary. The home window is displayed.
- In the home window, click "Tutorial" and select the project named "My first pages (Exercise)". The project is loaded.
  Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. My first pages (Exercise)".



Important

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

The "My_First_Pages" project is a **dynamic** project. This project is used to create a dynamic site: some operations will be performed on the browser, some operations will be performed on the server.

- To view the characteristics of the project:
  1. Select "Project .. Project description".
  2. In the window that opens, in the "Project" tab, you have the ability to check the type of site for this project: "Dynamic WebDev".
- In the project description, let's take a look at the style options of our site.
  1. Click the "Skin/Styles" tab.
  2. This screen allows you to see that this project uses the iStyle skin associated with the "iPhos & Light" palette.



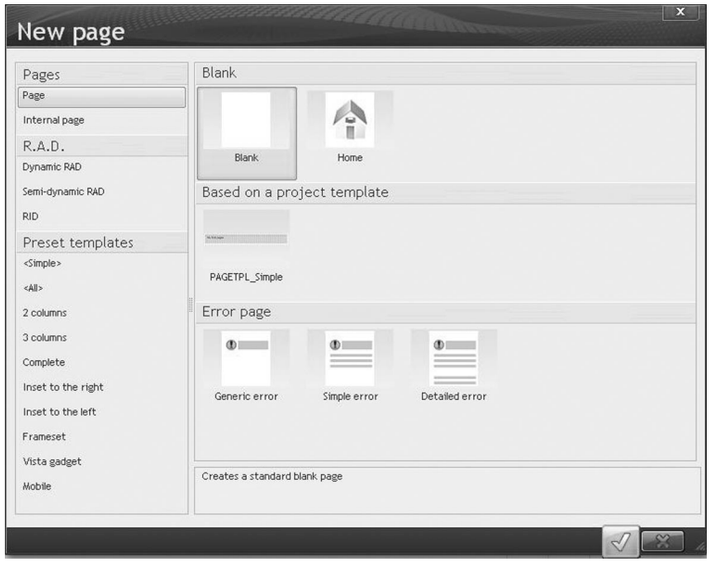
Note

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

3. You have the ability to change the skin or the palette of the site to give it a completely new style.
4. Close the description window of the project.

Creating the first page

- To create your first page:
  1. Select "File .. New" from the main menu of WebDev.
  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 (green button at the bottom of the screen).

We are now going to enter information about the page (name, title and description).

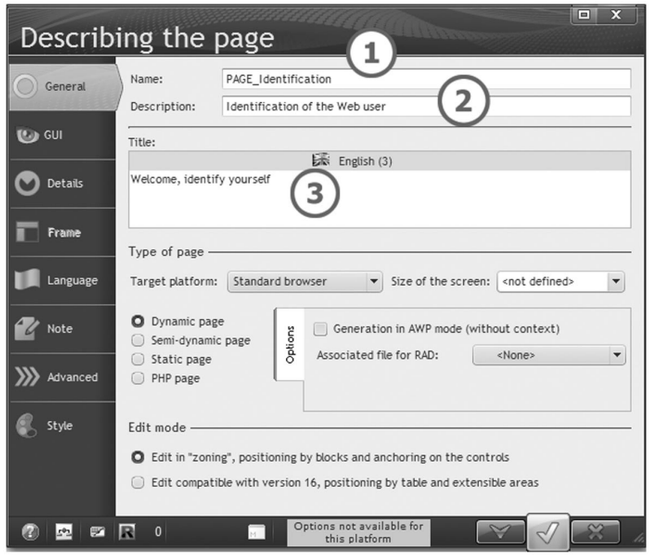
- Display the description of the page ("Page .. Description"). This window contains the name of the page: "PAGE_NoName1".



Note

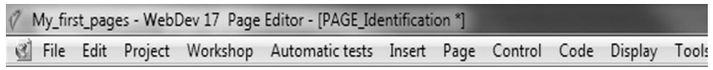
Study the name of the page proposed by WinDev: this name starts with the letters "PAGE_". This prefix is automatically implemented because the project uses a programming charter. The programming charter is used to define a prefix for each type of object. This programming charter will be presented later in this tutorial.

- 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 description of the page: "Identification of the Web user". This description is intended for the developer only. This description will be displayed in the documentation, when viewing the project elements, ...
3. The title of the page: "Welcome, 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  . Validate the window that is displayed.

Notes

For each page created in the editor, WebDev generates several types of files:
- A "WWH" file that contains the description of the page for the page editor. This file is saved in the project directory.
- 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, ...).

Displaying a formatted text

Welcoming a Web user is simple politeness, so you surely won't mind taking trouble with the layout and the appearance of this page. You are now going to learn how to insert text into your page and to format it ... Put your designer hat on!

▶ The HTML static is one of the ways proposed by WebDev to insert a text. To create an HTML static:

1. Click  in the toolbar.

2. Position the mouse cursor in your page, the outline of the HTML static is displayed.

3. Click the position where the HTML static must be created.

▶ To enter the text of the HTML static:

1. Click the static. The edit cursor is displayed as well as the toolbar of the HTML static.

2. Type the following text: "Welcome to my first site powered by WebDev!".

3. Click outside the HTML static to validate the input.

4. Click the control and resize it with the sizing handles in order to display the entire message. The sizing handles are the little squares that are displayed around the selection border of the HTML static.

Our welcome message is a little dull ... Now, all we have to do is format it!

▶ To format the text of the HTML static:

1. Select the HTML control and click in the control. The edit cursor is displayed. A formatting toolbar is displayed above the control.

2. Select the entire text with the mouse.

3. In the list of fonts (in the formatting options), choose "Verdana, Arial, Helvetica, sans-serif" (if not already done).

Internet

On Internet, the display fonts used are the ones found on the computer of the Web user. In our case, the default font will be "Verdana". If this font is not found on the computer of the Web user, the Internet browser will choose the "Arial" font. If this one is not found either, the browser will choose the "Helvetica" font, and so on.

In most cases, the specified fonts are grouped according to their similarity.

4. In the list of font sizes (in the formatting options), select "medium".

Internet

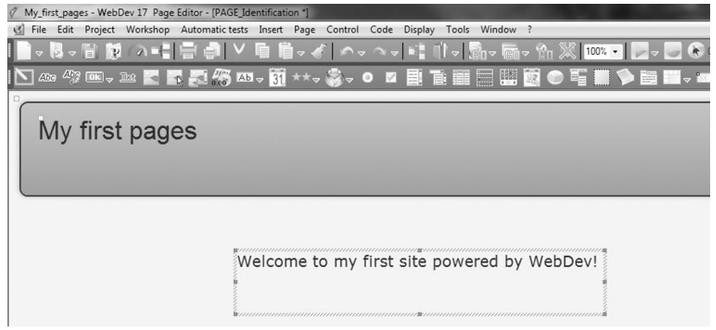
The size of a font can be expressed according to several methods:
- in relative size (5 for example): this size depends on the default size defined in the parameters of the browser,
- in standard size (medium for example),
- in points (8pt for example),
- in pixels (10px for example).

5. Enlarge the control if necessary.

Notes

Tip: To quickly resize a control, select "Adapt the size" from the popup menu of the control.

You also have the ability to center it in the page via "Control .. Alignment .. Center horizontally". Your static should look like this:

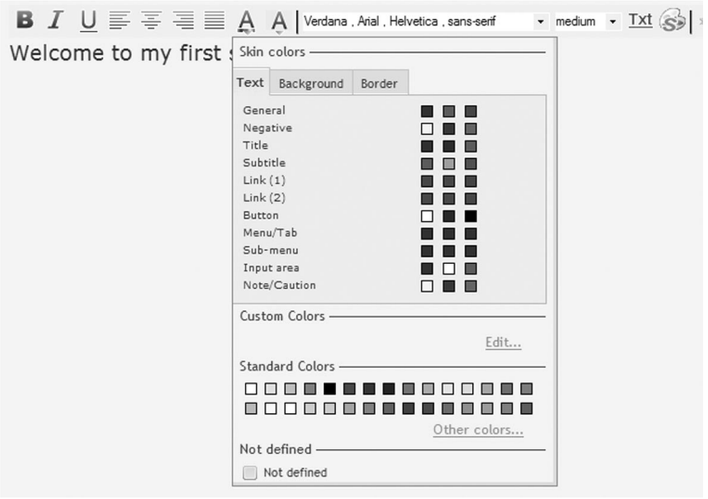


▶ Let's add a bit of color!

1. Select the part of text that must be colored ("WebDev" for example).

2. In the formatting options, click  . A window proposing a choice of colors is displayed.

3. Select the requested color (blue for example) and validate.



4. Perform the same operations for the entire text if necessary.
Your welcome message is a bit more welcoming now!

Entering information

We are now going to create the different controls used to identify the Web user.

Entering information in a page is performed via the edit controls. We are going to create the following edit controls:

- First name
- Date of birth
- Email address

▶ To create the "First name" edit control:

1. Click Ab . Click the position where the control must be created.
2. Right-click the control and select "Description".
3. Specify the name of the edit control: "EDT_FirstName".
4. Specify the caption: "First name".
5. Select the "Text" type. Select the following input mask: "1st letter in uppercase characters".
Validate the description window.



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 may be very annoying when performing a search for example. WebDev enables you to manage an input mask for a control. This is used to automatically format the value entered, without any code line.

▶ To create the "Date of birth" edit control:

1. Click the arrow found on the right of the icon Ab. In the list of preset controls, select a "Date" control (with the small calendar) and click the position where the control must be created.
2. Right-click the control and select "Description".
3. Specify if necessary:
- the name of the control: "EDT_Date_of_birth".
- the caption of the control: "Date of birth".
- the type of the control: Date.

4. Validate the description window. In the editor, resize the control if necessary (via the handles) to view its caption properly.



Notes

To align these two controls properly, use the alignment options:
1. Select the date control then the control that displays the first name.
2. In the toolbar, click the arrow found on the right of the alignment icon ( ).
3. The list of icons corresponding to the alignment features is displayed.
4. Hover each icon: the result is automatically proposed.
5. Select "Inside and outside left border".

▶ To create the "Email address" edit control:

1. Click the arrow found on the right of the icon Ab. In the list of preset controls, select an "Email" control and click the position where the control must be created.
2. Right-click the control and select "Description".
3. Specify if necessary:
- the name of the edit control: "EDT_Email_Address".
- the caption of the control: "Email address".
- the input mask: "Email address".

4. Validate the description window. In the editor, resize the control if necessary to view its caption properly.

Button for validating a page

We are now going to create a validation button. This button will be used to:

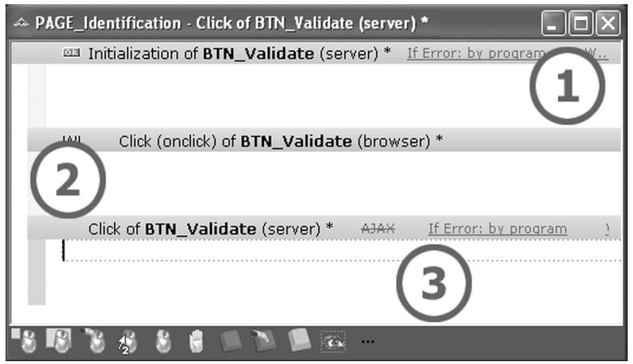
- Inform the Web user of the data entered,
- Calculate his age,
- Display the data in a new page.

▶ Add a button to the "PAGE_Identification" page:

1. Click Ok in the toolbar.
2. Position the button at the requested location in the page.
3. Right-click the button and select "Description" from the popup menu.
4. Specify the name and the caption: "BTN_Validate" and "Validate".
5. Validate the description window.
6. Display the code associated with the button ("Code" from the popup menu of the button).

The code edit window (called "Code editor") is displayed for the button that was created.

In the code window that is displayed, the code is divided into several sections of different colors:
- Initialization of BTN_Validate (server), orange code (1)
- Click (onclick) of BTN_Validate (browser), green code (2)
- Click of BTN_Validate (server), orange code (3)



The orange code is the code that runs on the server and the green code is the code that runs 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 are made of 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 are made of 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 is converted into 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 will redisplay the page.

▶ In this example, we will be using a server code and a browser code:
- in browser code, we will display the information entered.
- in server code, we will calculate and return the age of the Web user.

▶ Enter the following code in the browser click code:

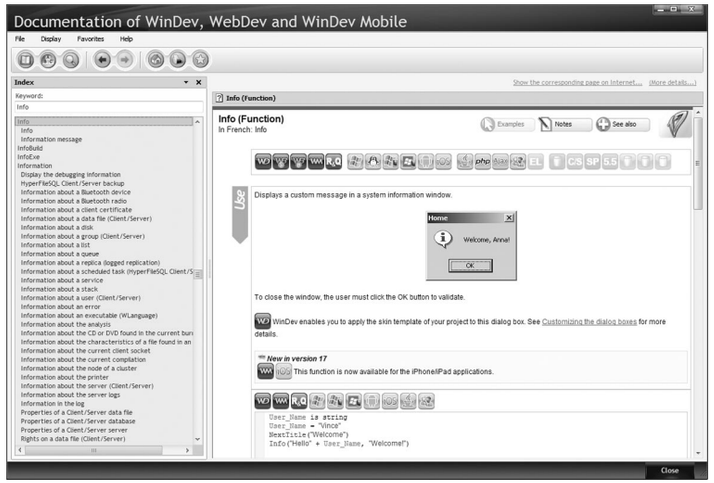
```
Info("You've entered: " , EDT_FirstName , ...
                          EDT_Date_of_Birth, EDT_Email_Address)
```

Notes

By entering this code in the code editor, you have noticed that different colors are used by the elements entered. The code editor enables 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.

In this code, the WLanguage function named **Info** is used to display an information box to the user. This information box displays a text and the content of the controls filled by the user.

▶ The online help enables you to get more details about this function: all you have to do is position on the name of the function and press [F1].
The help of the function is displayed in a specific "help browser".



If your computer is equipped with an Internet access, the help page currently found in the "help browser" can be displayed in its Internet version ("Show the corresponding page on Internet" link).



Notes

The online help for WinDev, WebDev and WinDev Mobile is available on Internet. Therefore, the online help can be accessed from any computer equipped with Internet access, without having to install the product. This help is updated on a regular basis.

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

The online help of WebDev enables you to get detailed information about the 2000 WLanguage functions. The online help also contains the help about the editors, controls, tips, ...

The online help is common to WinDev, WebDev and WinDev Mobile. The pages displayed correspond to the product currently used.

To start the Internet online help from the product:

1. Select "Tools .. Options .. General options of WebDev".
2. In the "Help" tab, select the access mode to the help database.

- Enter the following code in the server click code:

```
My_age is string
My_age = Age(EDT_Date_of_birth)
My_age = Middle(My_age, 3, 2)+" years "+ ...
     Middle(My_age, 5, 2)+ " months "+Middle(My_age, 7, 2)+ ...
    " days"
Info("You are "+ My_age)
```


This code calculates the age of the Web user from the date of birth and displays the result in a readable format. **Age** returns the age in years, months and days in YYYYMMDD format. To get:

- the number of years, we extract the last 2 digits corresponding to the year.
- the number of months, we extract the last 2 digits corresponding to the month.
- the number of days, we extract the last 2 digits corresponding to the day.

Notes

Why should I use a server code?

The WLanguage function named **Age** is an advanced function available on the server only.

WebDev indicates the functions that are available or not via the compilation errors. Copy the server code into the browser code: **Age** is displayed in red and a compilation error indicates that this function is not available.

Test of a page

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

- To save the modifications made to your page, click in the toolbar.
- Click . This icon is used to run the test of the page in the default browser installed on your computer.

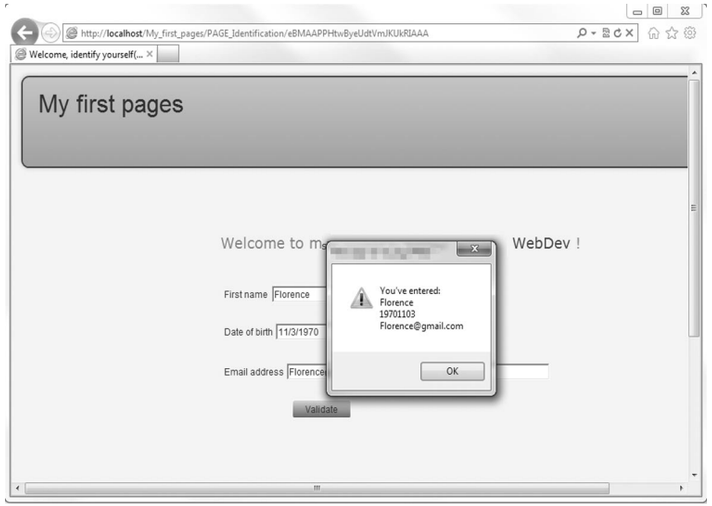
Notes

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 the dynamic WebDev sites. The "Connections" tab enables you to view your current connection to the site named "My_first_pages".

In this case, we are in test mode of the development computer, so 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. Enter a date of birth.
3. Enter an email address.
4. Click the "Validate" button.



5. Validate the two dialog boxes.

We are going to run another simple test. Enlarge your browser: the page controls remain on the right.

We are now going to improve this page on two levels:

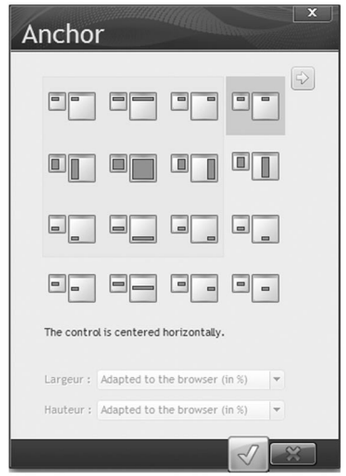
- We are going to anchor the controls in order for the controls to be centered when the browser is enlarged.
- We are going to check whether the controls have been filled. Indeed, the message is always displayed, no matter whether the controls are filled or not.

Anchoring the controls

WebDev allows you to easily anchor the controls. To do so, select "Anchoring" from the popup menu of the controls. In our example, we are going to anchor the controls so that they remain centered in the page.

▶ To anchor the "Welcome..." caption:
1. Select the control.
2. Display the popup menu of the control (right mouse click) and select "Anchoring".

The following window is displayed:



3. Select "Centered horizontally". Validate.
4. Repeat these operations for the edit controls and for the button of the page.
5. Save the page and run its test.

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.

The method for checking the three edit controls is the same one. The browser code of the "Validate" button becomes:

```
IF EDT_FirstName="" THEN
    Info("Enter a first name")
    ReturnToCapture(EDT_FirstName)
END
IF EDT_Date_of_birth="" THEN
    Info("Enter a date of birth")
    ReturnToCapture(EDT_Date_of_birth)
END
IF EDT_Email_Address="" THEN
    Info("Enter an email address")
    ReturnToCapture(EDT_Email_Address)
END
Info("You've entered: " , EDT_FirstName , ...
                    EDT_Date_of_Birth, EDT_Email_Address)
```

▶ To save the modifications made to your page, click in the toolbar and run the test of the page. A message is displayed if no information is entered.

▶ Go back to WebDev.

My second page: Displaying the information entered

To improve our communication with the Web user, we will:
- Ask the Web user to confirm the information entered.
- Display a second custom home page. This page will display a custom welcome message.

In a first time, let's create the page.

Creating the page

▶ To create the page:
1. Select "File .. New".
2. Click "Page". The wizard for page creation is displayed.
3. Select the "Page" option to the left of the window then select the "PAGETPL_Simple" preset template for example.
4. Validate.

We are now going to enter information about the page (name, title, description).

▶ Select "Page .. Description". The description window is displayed. Enter the following information:
1. The name of the page: "PAGE_Welcome".
2. The description of the page: "Welcoming the Web user".
3. The title of the page: "Welcome".
4. Validate the description window.
5. Save the page.

This page must display to the Web user a message containing the information entered in the previous page as well as the age of the Web user.

The information will be displayed in a static.

How do I get the information entered in the previous page? This information is not saved in a database.

The solution consists in passing parameters to our second page: when the identification page is validated, the data entered will be passed in parameter to the welcome page.

Passing the parameters must be managed by the two pages: the page used to enter the data and the page used to display the data.

Creating a static control

We'll start by creating a static control in our page. This static control will allow us to display the welcome text. As this text is based on the information entered in the previous page, we'll leave static control empty: its content will be defined by programming.

▶ To create a Static control:
1. Click Abc then click the location in the page where the control must be created.
2. In the control description, enter the following information:
   - The control is named STC_INFO.
   - The control has no caption.
3. Validate the description window.
4. We will now associate a style with the text that will be displayed in this static.

Select "Choose a WebDev style" from the popup menu of the control. Select the "STC_Simple_Bold" style. Validate.
5. In the editor, change the size of the caption: use the handle to increase the size of the caption (in height and width).
6. Save the page.

Managing the parameters

Now that the Static control is created, let's see how to manage the parameters sent to the page. Indeed, the "PAGE_Identification" page will send the personal details of the Web user (first name, date of birth and email address) to the "PAGE_Welcome" page.

▶ In order for a page to receive parameters, these parameters must be declared in the "Declaration code of the global variables" of the page.
1. Display (if necessary) the "Page_Welcome" page in the editor.
2. Display the description of the page ("Page .. Description") and click the "Code" button (bottom left). The code editor is displayed with the different Server and Browser processes associated with the page.
3. In the "Declaration code of the global variables of PAGE_Welcome", enter the following code:

```
PROCEDURE PAGE_Welcome(FirstName, DateOfBirth, EmailAddress)
```

This line is used to declare that the "PAGE_Welcome" page will accept three parameters when it is opened. These parameters can be handled in the code of the page via the names of the variables used (FirstName, DateOfBirth, EmailAddress).

▶ We now want to use the parameters to build the content of the static displayed in the page. To do so, we will be using the initialization code of the page.

▶ Enter the following code in the initialization code of the page:

```
// Calculate the age
My_Age is string
My_Age = Age(DateOfBirth)
My_Age = Middle(My_Age, 3, 2)+ ...
          " years "+Middle(My_Age, 5, 2)+ " months "+...
          Middle(My_Age, 7, 2)+ " days old"
// Build the STC_INFO static
STC_INFO = [
Welcome %1 to my first site.

Your email address is %2.

You are %3.
]
STC_INFO = StringBuild(STC_INFO,FirstName, ...
           EmailAddress, My_Age)
```

- Let's try to understand this code.
  - First of all, the first part of the code is used to calculate the age and to build the result of the calculation in letters. We already know this code.
  - The second part of the code is used to build the content of the static.
    The STC_INFO static is initialized with a multiline string containing the text to display and several %1, %2, %3. These %n represent the parameters that can change. They will be automatically replaced by the requested value when ***StringBuild*** is called.



Notes

Why not directly build a string with the name of the variables?

The content of the static could have been built as follows:

```
STC_INFO = "Welcome "+ FirstName + "to my first site."+ CR + CR+ ...
           "Your email address is "+ EmailAddress +"."+ CR + CR + ...
           "You are " + MyAge+"."
```

This notation presents several drawbacks:

- the readability of the code: a multiline string is easier to read.
- the translation: a multiline string can be entirely translated. The words and the parameters can be reversed. The translator has a single obligation: include %1, %2, %3.

Creating the Popup page

- Before running the test of our pages, we must modify the PAGE_Identification page. This page will display a Popup dialog box. If the user confirms the information displayed, the PAGE_Home page will be opened by passing parameters.

1. Display the PAGE_Identification page in the editor. If this page is opened, all you have to do is click the corresponding tab.



2. Insert a Popup page via "Insert .. Popup". A new page is displayed in the page editor.

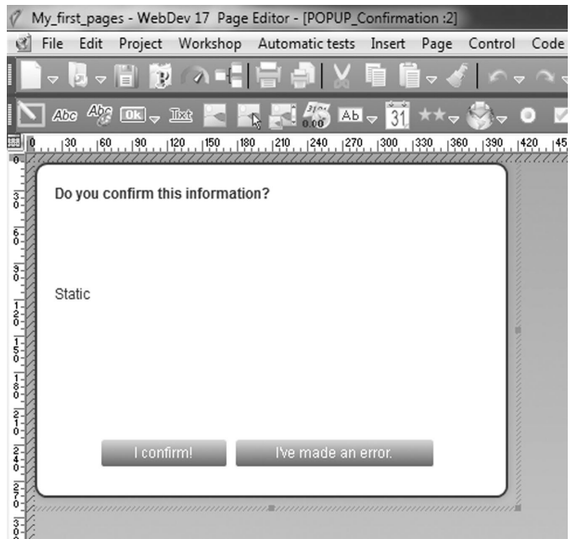
3. Display the description window of the Popup page. In the "General" tab, specify the name of the page: "POPUP_Confirmation".

4. In the "Style" tab, we are going to define the style of our Popup page. Select the "Border/Background" element and define:

- a border color (dark blue for example),
- a background color (light gray for example),
- a preset type of border and background.

5. Save.

- We are now going to create the different controls of this popup. The result will be as follows:



This popup page contains the following controls:

- A static control to ask the question and a static control to display the information,
- Two buttons to validate the dialog.

- We are going to create the two Static controls:

1. Click Abc in the toolbar, then inside the Popup page. Position the control at the top of the Popup. Edit the caption directly ("Edit the caption" from the popup menu) and enter "Do you confirm this information?". Validate.

2. Click Abc in the toolbar, then inside the Popup page. Position the control below the previous static control. Modify the name of this control ("STC_Info"). Enlarge this control. This control will be used to display the data to validate.

- We are now going to create the two Button controls. These two controls will be positioned below the last created static control.

1. Click Ok in the toolbar. Create the two buttons.

2. The caption of the first button is "I confirm!". This button allows the Web user to confirm the information displayed and to display the "PAGE_Welcome" page with the information entered. The server code of this button is as follows:

```
PageDisplay(PAGE_Welcome,EDT_FirstName,...
                         EDT_Date_of_birth,EDT_Email_Address)
```

This code is used to display the "PAGE_Welcome" page by passing the parameters expected by this page (parameters defined in the "Declaration code of global variables" of "Page_Welcome" page).

3. The caption of the second button is "I've made an error". This button allows the Web user to go back to the page in order to modify the information. This button closes the Popup displayed: no server action is required.

- In the description window of the control, specify the following options and validate:
  - Operation on the controls: None
  - Action: None
- Display the code of the button ("Code" from the popup menu).
  The browser code of this button is as follows:

```
PopupClose(POPUP_Confirmation)
```

This code is used to close the Popup.

Displaying the Popup page

All we have to do now is display the Popup page from the validation button of the "PAGE_Identification" page.

1. Display the code of the "BTN_Validate" button found in the "PAGE_Identification" page.

2. Write into comments the browser code line containing the function named "Info". To write a code line into comments, all you have to do is type the "// " characters in front of the beginning of the line.

3. Enter the following code:

```
POPUP_Confirmation.STC_Info = ...
    "First name: " + EDT_FirstName +CR+ ...
    "Date of birth: " + ...
    DateToString(EDT_Date_of_birth, "MM/DD/YYYY")+ ...
    CR+ "Email: " + EDT_Email_Address
PopupDisplay(POPUP_Confirmation)
```

This code is used to:

- build the caption that is assigned to the STC_INFO control found in the Popup page.
- Display the Popup page via ***PopupDisplay***.

4. Write into comments the server code lines. To write several code lines into comments, select the lines and select "Code .. Comment out".

5. Close the code editor.

6. In the description window of the BTN_Validate button, specify the following options and validate:

- Operation on the controls: None
- Action: None

7. Save your modifications.

Test of the site

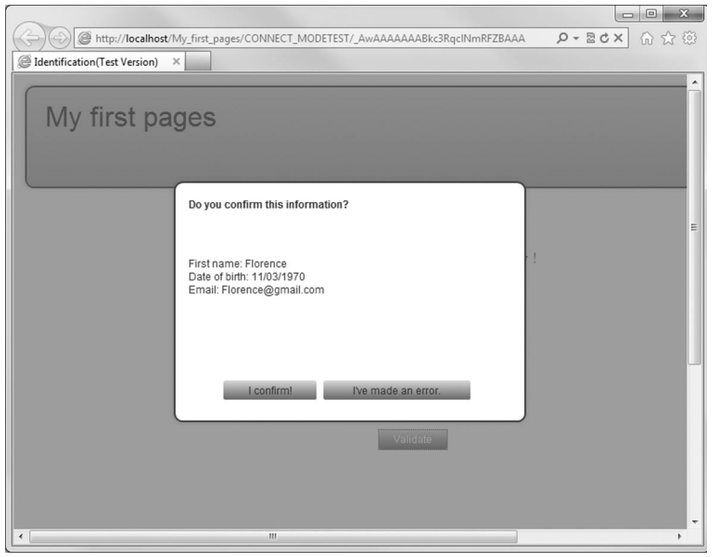
We are now going to run the test of our first site:

1. Click  . WebDev asks for the page that corresponds to the first page of the project.



Select the "PAGE_Identification" page and validate.

2. The home page is displayed in the browser. Enter the requested information and validate. The confirmation popup is displayed.



LESSON 1.4. 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 for WebDev and its language (WLanguage) have been used in the previous lesson. This lesson enables 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.

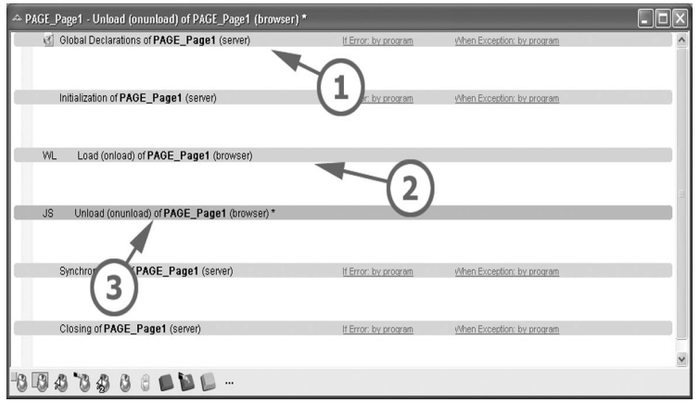
The pages found in the dynamic sites have a server code and a browser code. The actions can be performed locally (on the computer of the Web user) and on the server (displaying a page that contains information read in a database for example)

The pages found in the static sites have 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 then the page is sent to the server. The server code is run before a response page is sent toward 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, ...

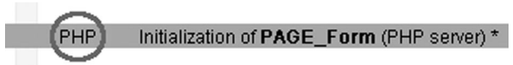
The WebDev engine is required to run this code (PHP engine for a PHP site). This code is available in the dynamic sites only.

For the dynamic PHP sites, some PHP code can be directly entered in the code editor.

- To switch from WLanguage code to PHP code, click the "WL" symbol found in front of the code.



- To switch from PHP code to WLanguage code, click the "PHP" symbol found in front of the code.



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, click the "WL" symbol found in front of the code.



- To switch from JavaScript code to WLanguage code, click the "JS" symbol found in front of the code.



Declaring the different types of variables

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

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

WLanguage enables you to handle two types of variables:

- 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 simple 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 code 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 of one of the previous types.



Notes

If your project uses a programming charter, the name of all the variables will be automatically prefixed. This prefix enables you to easily identify the type and the scope of your variables:

- "g" prefix for the global variables
- "n" prefix for the integer type
- "s" prefix for the string type
- "r" prefix for the real type, ...

Caution: the display of the prefixes must be enabled ("Advanced" tab of the project description, available for a standard environment only).



Caution !

Most developers are tempted to declare all their variables as "global" to 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 side-effects problems.

Therefore, we advise you to declare a limited number of global variables then to declare the local variables.

If you want to share values between 2 pages, we highly recommend that you use the method for "passing parameters" that was previously explained in this tutorial.

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)

The WLanguage enables 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
   OTHER CASE
   // it's another day
END
```

Loop statements (server code and browser code)

The WLanguage proposes several methods for managing the loops:

- **FOR**, **END** for a specified number of iterations.
- **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

The WLanguage also proposes loops (FOR EACH, END) 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 running.

Comments

To write comments in your 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, then press [CTRL] + [/] (on the numeric keypad).

To reverse this operation (to remove the comments), select the code lines with the mouse or with the keyboard, then 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 local code (process of a control, ...) or global code (global procedure, class, ...) independent of the current control.

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

The MyPage keyword (server code)

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

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


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. For more details, see "Questions and Answers" at the end of this tutorial.

Local procedure

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

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

Global procedure and set of procedures

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

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

- A set of server procedures; file with ".WDG" extension.
- A set of browser procedures; file with ".WWN" extension.

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

A set of procedures is used to:

- share the global procedures among several developers, for the same project.
- share the global procedures among several projects. Indeed, the same set of procedures can be used in several projects.

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

To decide 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 "My_First_Pages" project, you have noticed that a procedure managed some parameters. The parameters can be mandatory or optional.

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] simultaneously to go back to the name of the procedure in the previous code.

Creating a procedure

A procedure can be created directly from the project explorer ("Project Explorer" pane) available in the main editor of WebDev.

- 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 comes up in the project explorer.
- To create a global procedure:
  1. Select the set of procedures where 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 of 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 ("Create a procedure .. Create a procedure containing the selected code" from the popup menu).

A practical example on the procedures and the functions

Enough theory, let's go down to work!

- Open the project named "Programming_Concepts.WWP".
  1. Close the current project if necessary. The home window is displayed.
  2. In the home window, click "Tutorial" and select the "Programming concepts" project. The project is loaded.

  Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. Programming concepts".
- Open the "PAGE_Procedures.WWH" page (double-click the name of the page in the project explorer).
- Run the test of the page ("Go" icon).
  1. Click the "Change" and "Change the image" buttons to run a procedure.
  2. Click the "What is today's date?" button to run a function.

  The code run is displayed in both cases.

Processing the strings (server code and browser code)

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

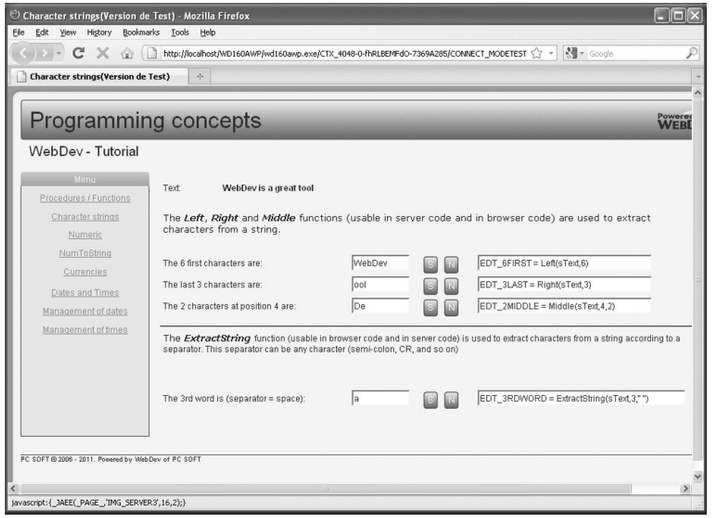
The WLanguage offers several features for handling the character strings: WLanguage functions, extraction operators, concatenation operators, ...

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

Practical example

- Open (if necessary) the "Programming_Concepts.WWP" project ("? .. Tutorial .. Programming concepts").
- Open the "PAGE_Handling_strings.WWH" page (double-click the name of the page in the project explorer).
- To run the test of the page, click  .

The "PAGE_Handling_strings" page presents the different operations that will be explained in this lesson:



Details (server code 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. This is called **string concatenation**.
The "+" operator is used to concatenate two strings:

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

A section of a string can be extracted by:

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

```
Info(STC_Text[[1 to 6]])   // displays "WebDev"
```

- the function named ***ExtractString***:

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

- the function named ***Middle*** that extracts a section of a string from a string:

```
Info(Middle(STC_Text2,27,6))  // displays "WebDev"
```


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

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

- the function named ***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 search for 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(TEXT2, SoughtString, 1, IgnoreCase)
```

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

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

Processing the numeric values (server code and browser code)

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

Practical example

- Open (if necessary) the "Programming_Concepts.WWP" project ("? .. Tutorial .. Programming concepts").
- Open the "PAGE_NumericCalculations.WWH" page (double-click the name of the page in the explorer).
- To run the test of the page, click GO. This page gives an overview of the operations that can be performed on the "numeric values".

Details

A numeric 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, 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***:

```
// Display 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***:

```
// Display 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***:

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

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

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

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

```
// Calculate 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 (if necessary) the "Programming_Concepts.WWP" project ("? .. Tutorial .. Programming concepts").
- Open the "PAGE_Currency.WWH" page. This page gives an overview of the operations that can be performed on the "Currencies".
- To run the test of the page, click GO.
  Test the different actions presented in the page. The code run is displayed in the page.

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 other hand, with the following code, the result is correct:

```
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 gives you the ability to configure the number of digits for the integer part and the number of digits for the decimal part.

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

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 into a numeric variable (and conversely).

For example:

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

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

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


- Open the "PAGE_NumToString.WWH" page and run its test.

Notes

Server code 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 code and browser code)

Overview

To easily mange the dates and the times in your applications, WinDev includes:

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

Practical example

- Open (if necessary) the "Programming_Concepts.WWP" project ("? .. Tutorial .. Programming concepts").
- Open the "PAGE_DateTime.WWH" page. This page illustrates the explanations given in the rest of this lesson.
- Run the test of this page.

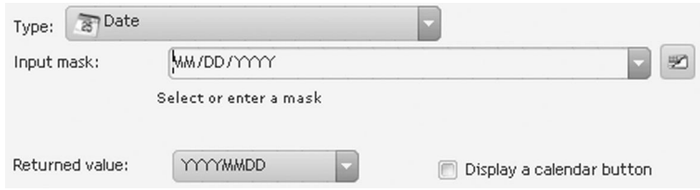
Input mask and returned value

The dates and the times are strings with a preset format.

In a "Date" or "Time" control, we must distinguish between:

- The **input mask**: the date and the time will be displayed in this format.
- 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:



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" (for example: the value entered in "23/04/2011" format will return "20110423" to the program).

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 the following format
"25/12/98"
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 the following format
"12/25/2098"
TDATE="19981225"      //displays the date in an incorrect format
```

You should pay great attention to the format of the value returned 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 milliseconds).



Notes

Reminder: By default, the format (input mask) of the control corresponds to the date mask defined by the project ("Project .. Project description", "Languages" tab). 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.

The dates (server code and browser code)

Practical example

- Open (if necessary) the "Programming_Concepts.WWP" project ("? .. Tutorial .. Programming concepts").
- Open the "PAGE_HandlingDates.WWH" page. This page illustrates the explanations given in the rest of this lesson.
- Run the test of this page ().

What is today's date?

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

```
Info("Today's date is " + Today())
```

To check this feature in our example, click the "What is today's date?" 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's date is " + DateToString(Today(), ...
                                  maskSystemDate))
```


The function named ***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 project characteristics ("Project .. Project description .. Languages") .
To check this feature in our example, click the "What is today's date (in clear)?" button.

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

- To spell out the date, use ***IntegerToDayInAlpha*** and ***IntegerToMonthInAlpha***:

```
varDate, varTDate are Strings
varTDate = Today()
varDate = IntegerToDayInAlpha(DateToInteger(varTDate))+ ...
    " " + varTDate[[7 to]] + " " + ...
    IntegerToMonthInAlpha(DateToInteger(varTDate))
Info("Today's date is "+varDate)
```

The functions named ***IntegerToDayInAlpha*** and ***IntegerToMonthInAlpha*** spell out the day and the month corresponding to the date passed in parameter.
IntegerToDayInAlpha and ***IntegerToMonthInAlpha*** expect an integer. Therefore, ***DateToInteger*** must be used to convert a "string" date into an integer.
To check this feature in our example, click the "What is today's date (in letters)?" button.

How many days between two dates?

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

```
NumberOfDays is int
NumberOfDays = DateDifference("20100101",Today())
Info("The number of days between", ...
     DateToString(DateSys(),maskSystemDate) +...
     " and " + DateToString("20100101",maskSystemDate) + ...
     " is " + NumberOfDays)
```

To check this feature in our example, click the "Number of days between two dates" button.

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

- To find out the day of the week corresponding to a given date, use ***IntegerToDayInAlpha***:

```
Day is string
Day = IntegerToDayInAlpha(DateToInteger("17890714"))
Info("The 14th of July 1789 was a " + Day)
```

Note: You also have the ability to use ***DateToDayInAlpha***.

```
Day = DateToDayInAlpha("17890714")
```

To test this feature in our example, click the "Day of the week" button.

Practical exercise

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

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

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

The times (server code and browser code)

Practical example

- Open (if necessary) the "Programming_Concepts.WWP" project ("? .. Tutorial .. Programming concepts").
- Open the "PAGE_HandlingTimes.WWH" page. This page illustrates the explanations given in the rest of this lesson.
- Run the test of this page ("GO" button).

What time is it?

- To find out the time, all you have to do is use the function named ***Now*** (or ***TimeSys***). The function named ***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 "What time is it?" button.

- You want to display the time in "HH:MM:SS:CC" format? The code becomes:

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

The function named ***TimeToString*** transforms a time in "HHMMSSCC" format into a string in "HH: MM: SS:CC" format.
To test this feature in our example, click the "What time is it (again, but in clear)?" button.

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

```
MyTime is string = ...
     Left(TimeToString(Now()),8)
Info("It is "+ MyTime)
```

The function named ***Left*** returns the first *n* characters of a string (8 in our example).
To test this feature in our example, click the "What time is it (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())
Sign_Time is string
Result_Time is string = IntegerToTime(Abs(Diff))
IF Diff < 0 THEN Sign_time = "-"
Info("Time passed: " + Sign_Time + Left(Result, 2) + ...
        "hours"+ Middle(Result_Time, 3, 2) + "minutes"+...
        Middle(Result_Time, 5, 2) + "seconds" +...
        Middle(Result_Time, 7, 2) + " hundredths of a second")
```


The function named ***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.
The function named ***TimeToInteger*** performs the reverse operation.



Caution !

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

To test this feature in our example, click the "Time (in seconds) passed since 12:15" button.

Date, Time, DateTime and Duration variables (server code)

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

These types of variables are available in server code only.

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

Type	Default format
Date	YYYYMMDD
Time	HHMMSSCC
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 can use the following syntaxes:

```
MyDate is Date = "20111021"
// Displays the formatted date
Info(DateToString(MyDate,maskSystemDate))
//Displays "10/21/2011"

// 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,maskSystemDate))
//Displays "11/22/2011"
```

In this code, **Year**, **Month** and **Day** are properties of the WLanguage. The following syntax enables you to use the WLanguage properties on WebDev elements:

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

See the online help for more details.

Calculations with dates and times

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

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



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

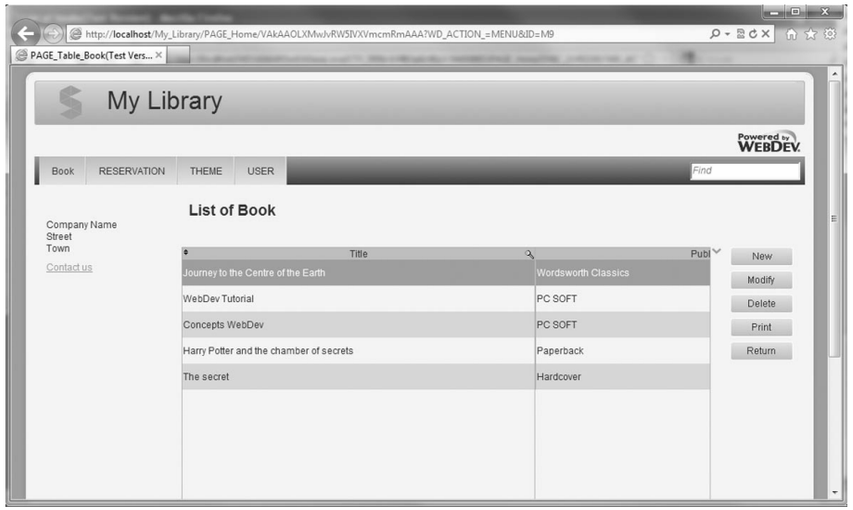
In this part, we will present the databases that can be handled by WebDev. For teaching purposes, we are going to develop an Intranet site from A to Z, from the creation of the analysis to the test of the site. The deployment of the site will be explained in a specific section.

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 HyperFileSQL Classic, the free database supplied with WebDev. The HyperFileSQL Client/Server database will be handled later in this tutorial.

Let's see what you are going to do first. You can directly open the corresponding project using "? .. Tutorial .. Site with data (Library)". Click the "GO" button to start the execution of the site.



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

We will then create the site via the RAD module (Rapid Application Development).

Creating the project

▶ To create the project:

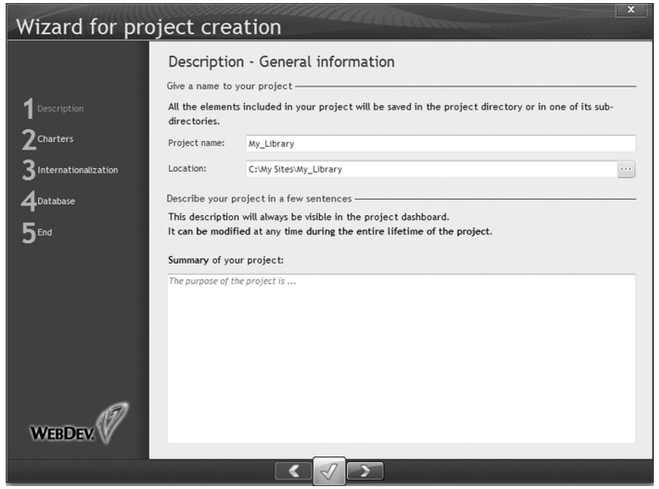
1. Start WebDev (if not already done). Close the current project if necessary.

2. In the home window, click "Create a project" then double-click "Internet site". The wizard for project creation starts. The different planes of the wizard help you create your project. The information specified in this wizard can be modified afterwards.

Notes

Tip: To create a project, you also have the ability to select "File .. New .. Project".

3. The first plane of the wizard is used to enter the name of the project, its location and its description. In our case, this project will be named "My_Library".



By default, WebDev proposes to create this project in the "\My Sites\My_Library" 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 summary of the project, type "The purpose of the project is to manage a library".

4. The different steps of the wizard are specified on the left side of the wizard. These steps can be clicked directly. The other screens of step 1 ("Description") are not fundamental, so click "2-Charters" directly.

Notes

The different options for project creation presented in this paragraph assume that you are using the simplified environment of WebDev (see “Environment required by the Tutorial”, page 35).

Additional options may appear if you are using another type of environment.

5. This step is used to define the style book. Choose the "iStyle" skin. Go to the next screen of the wizard (click the yellow arrow oriented to the right).

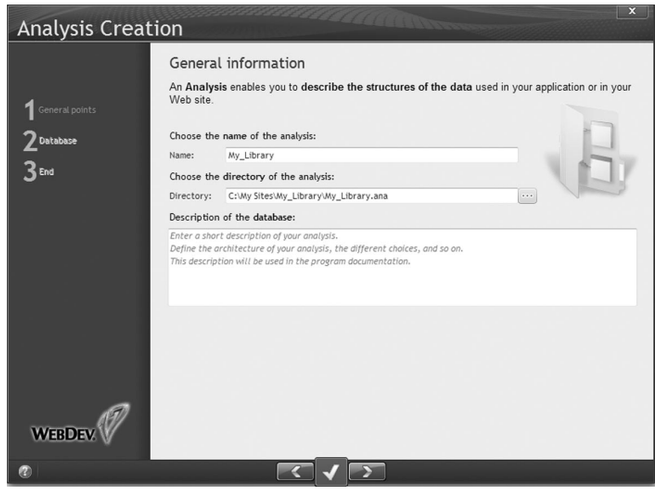
6. We will now specify the information regarding the database. Click "4 Database".

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

Creating the analysis

▶ The steps of the wizard for analysis creation are as follows:

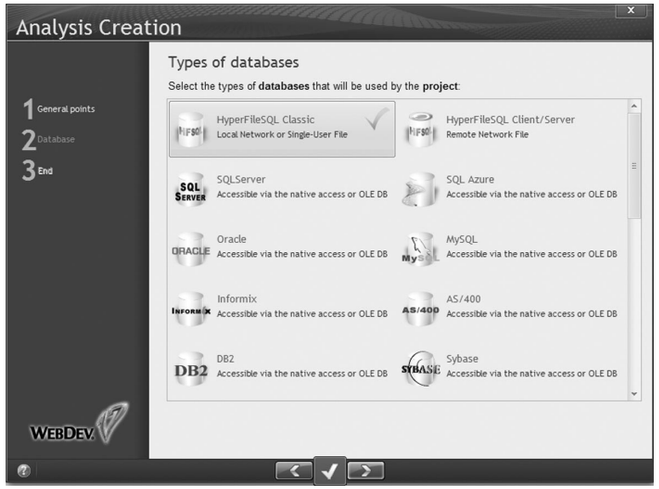
1. Specify the name and directory of the analysis.



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 parame-

ters. Go to the next wizard screen.

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

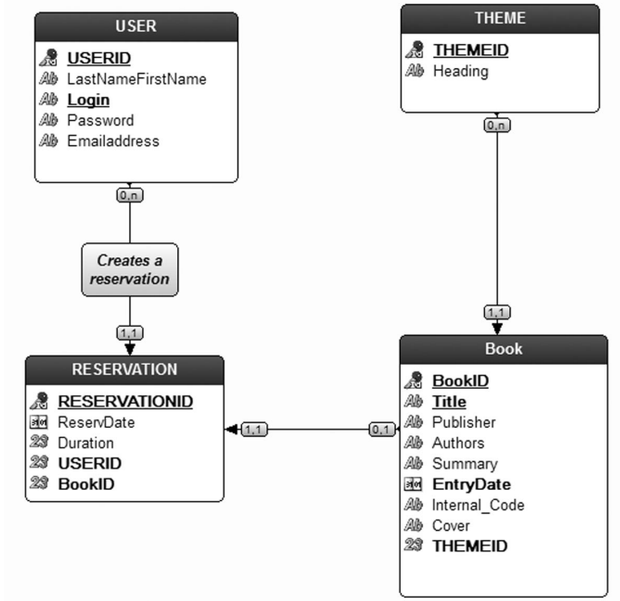


3. Validate. The wizard for creating a data file starts automatically.

Creating the description of the data files

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

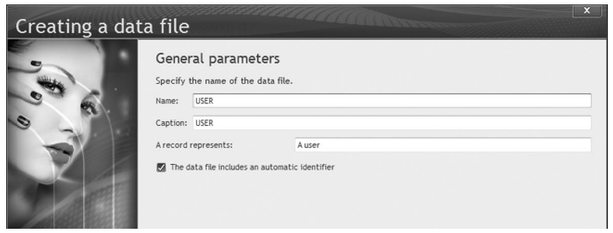
- BOOK
- RESERVATION
- THEME
- USER



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

"A record represents" 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 "The data file includes an automatic identifier".
This option indicates whether the data file must include a unique key, automatically managed by WebDev.

Notes

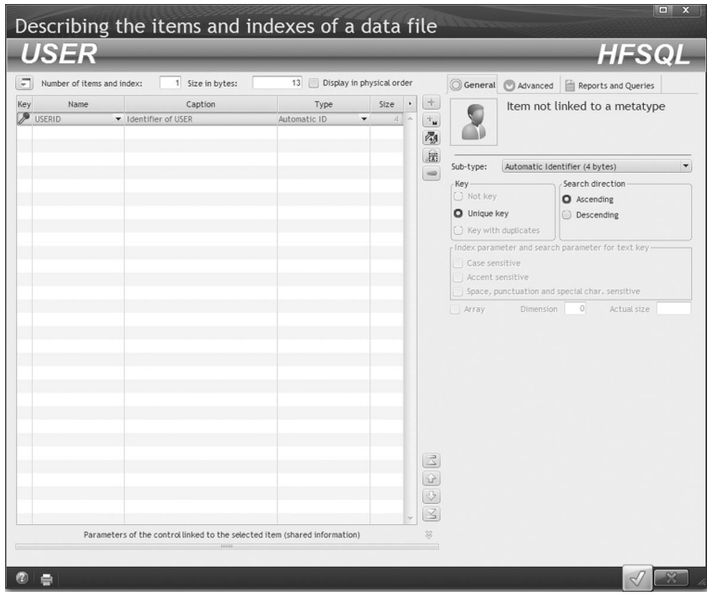
To create the identifier (an identifier is a unique key), you can create a numeric item whose type is "Automatic identifier".

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.

This choice can be unchecked 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 the database associated with the data file. We are going to work on HyperFileSQL Classic data files. Go to the next screen.
5. Keep the default options and go to the next screen.
6. Click the green button to validate. The data file is automatically created in the analysis.

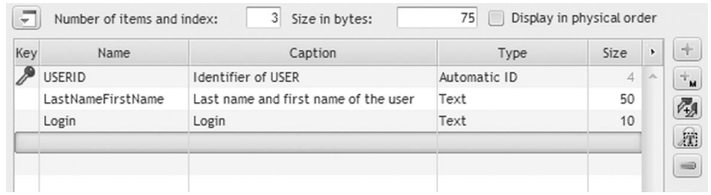
The description window of items is opened.



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 "LastNameFirstName" item. This item will contain the last name and the first name of the user.
  1. In the description window of items, click the "Name" column of the first empty line. This column automatically becomes editable. Enter "LastNameFirstName".
  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 and first name of the user". In the "Type" column, the "Text" type is automatically selected. Don't change anything.

- We are 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 line. This column automatically becomes editable. Enter "Login".
  2. We are going to modify the size of the item. Click "50" and replace "50" by "10". Click the next line. The values are automatically updated.



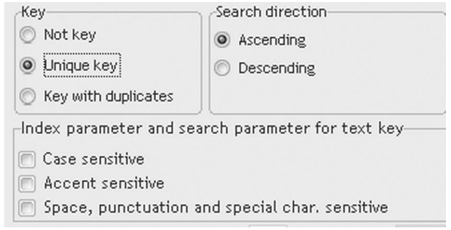
3. This item will be a key item in our data file: the keys are used to improve the speed for accessing the data and to simplify the browse operations performed on the data files. You will have the ability to perform searches or browses on this item.

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,
- unique key: the value of this key will be unique in the entire data file (which means in all the records found in the data file)
- key with duplicates: the value of this key can be found several times in the data file.

4. To define the key, you must: reselect the line 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 the key. In our case, the login is a unique key.



▶ We are now going to create the "Password" item.

1. Position 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 bottom 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 (right section of the screen) and indicate that this item must be generated by RAD for the controls in form mode only (uncheck "a column in table mode" and "a control for the reports").

We are going to create an email item. Easy! To do so, we are going to use the metatypes.

Notes

Metatype

A metatype is a combination of characteristics for an item and for its linked control. For example, a "Fax" metatype will contain the type and length of the item, the input mask, the alignment, ...

You can use the metatype items supplied with WebDev or create your own metatypes.

To use a metatype item, click the "Metatypes" button in the description of the analysis items.

▶ To add an item created from a metatype:

1. Click  .

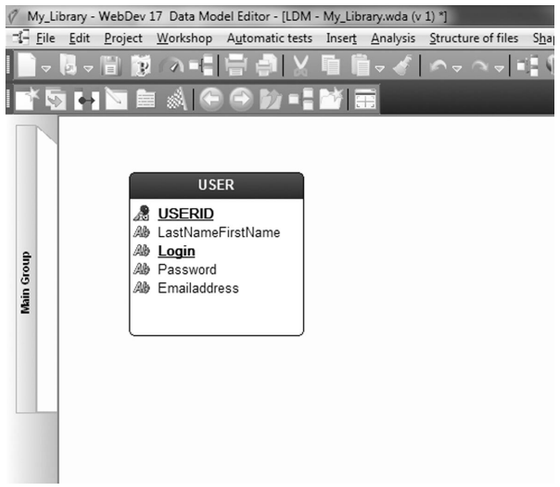
2. Select "Email address".

3. Validate.

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

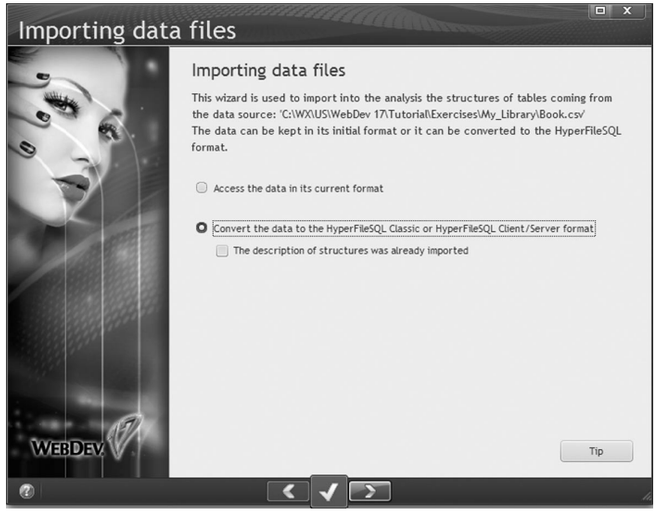
To create the BOOK data file (containing the characteristics of the books), we will be using another method: importing a CSV file. From the CSV file containing the data, WebDev will create the description of the data file in the analysis as well as the corresponding HyperFileSQL data file with the data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

To import a CSV file into the analysis:

1. In the file explorer of Windows, open the following sub-directory of WebDev: "\Tutorial\Exercises\My_Library".

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.

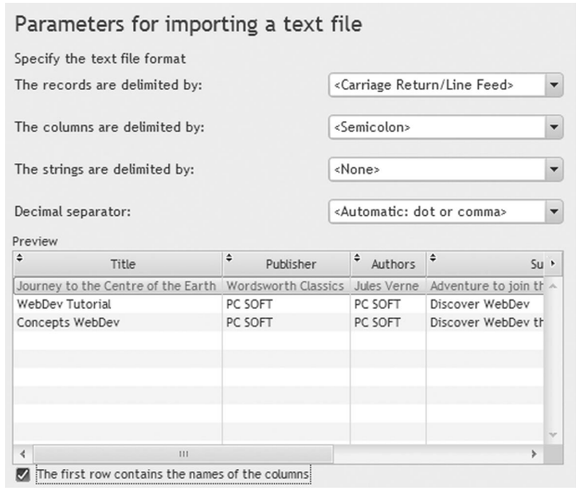


4. The content of the CSV file will now be converted to the HyperFileSQL format. Select "Convert the data to the HyperFileSQL Classic or HyperFileSQL Client/Server format". Go to the next wizard screen.

5. Select the format of the files to import. Select "Text file". Go to the next plane.

6. WebDev indicates the path of the file to import. Go to the next wizard screen.

7. Indicate the following import parameters:



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

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

9. Validate the wizard. The wizard for data conversion starts.

10. Validate the creation directory of the HyperFileSQL data files. Go to the next wizard screen.

11. Uncheck "Save the conversion parameters" and validate. 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.

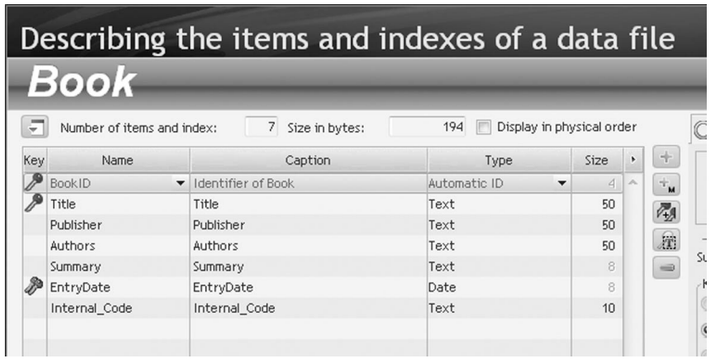
2. This data file contains no automatic identifier: check "Automatic identifier".

3. Click  to display the description of the items found in the data file.

4. We are going to modify some information:

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

We get the following data:



- 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.

WebDev proposes a specific pane, the "Dictionary" pane, allowing you to easily handle the dictionaries.

- To import elements from a dictionary:
  1. Display (if necessary) the "Dictionary" pane ("Display .. Toolbars .. Panes .. Other panes .. Dictionary"). A dialog box asks to synchronize the analysis. Click "No".
  2. Click the pane and select "Open" from the popup menu.
  3. Select the "Dictio_TUT" file found in the following WebDev sub-directory: "\Tutorial\Exercises\My_Library". The dictionary is displayed 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 HyperFileSQL data files. The last file was prepared in this format.

- To import the HyperFileSQL data files:
  1. In the file explorer of Windows, open the following sub-directory of WebDev: "\Tutorial\Exercises\My Library".
  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.

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

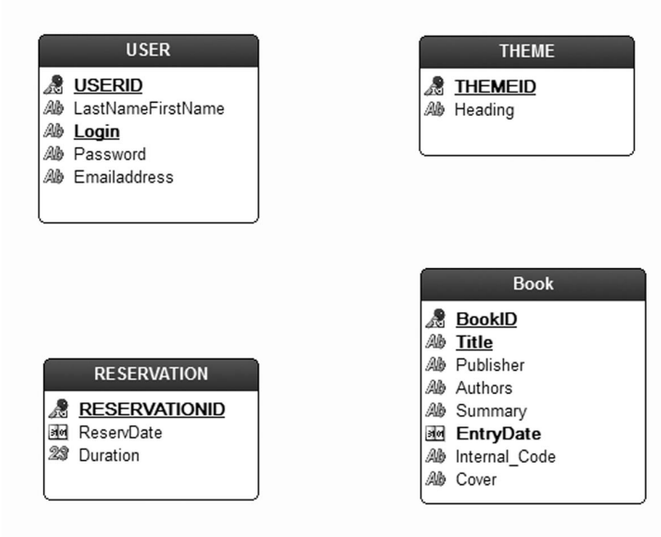
Note: To handle the data found in the THEME file, copy the THEME.fic file and the THEME.ndx file (found in the "\Tutorial\Exercises\My Library" directory) into the EXE sub-directory of your project.

Notes

To open the directory of the current project in WebDev, click  .

Creating the links

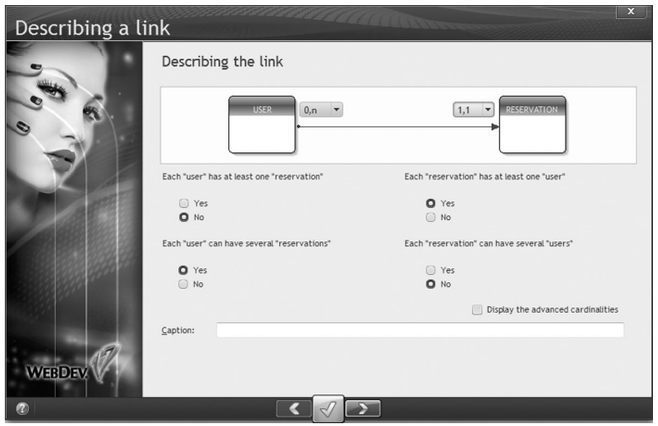
All the descriptions of data files required by the site for library management have been created.



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. Select "Insert .. 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:



- 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

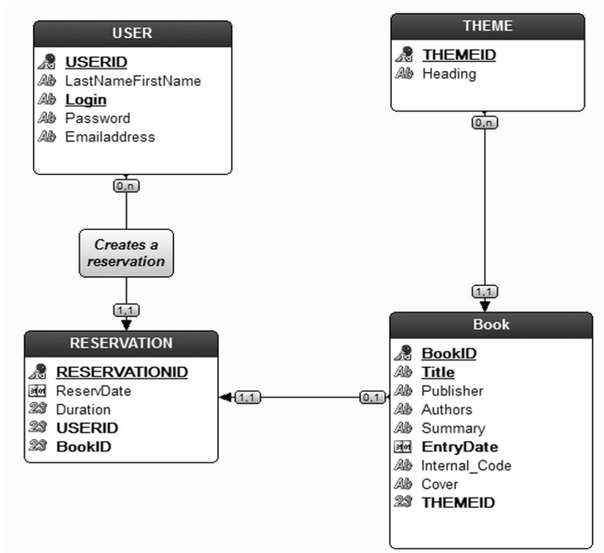
You also have the ability to enter the cardinalities of the link in the wizard.

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:



Generating 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 the 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 made to the analysis and to 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 made to 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 whenever it is modified. Generate the analysis once all the modifications have been made and go to the programming step.

- To generate the analysis:
  1. In the data model editor, select "Analysis .. Generation".
  2. The generation of the analysis is automatically performed.

3. The wizard for automatic modification starts.

Notes

What is the automatic modification?

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.

4. Keep the selected option and go to the next screen.
5. Validate the search location: the physical data files are sought in the EXE directory of your project.
6. Select all the proposed data files (including the THEME file). Go to the next screen to validate.
7. Validate the different planes until the automatic data modification is performed.

Configuring the analysis for RAD

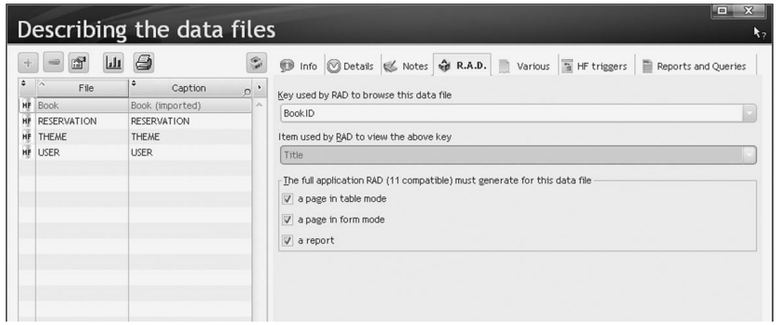
When generating the site, the pages and the reports, the RAD module (Rapid Application Development) uses the options specified in the analysis. We recommend that you take a little bit of time to configure these options in order to get the best possible result. The elements used by RAD are defined:

- 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. Select "Structures of files .. Description of data files".
2. We are going to modify the caption of the Book file first by removing "(Imported)".
   Select the "Book" file in the list on the left and modify its caption by deleting "(imported)".
3. Select the "Book" file in the table 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 viewing.
- The elements generated by RAD for the data file.

The table below summarizes the configuration to perform:

Data file	*Key used*	*Item to view*	*Element to generate*
Book	BookID	Title	Table, Form, Report
RESERVATION	RESERVATIONID		Table, Form, Report
THEME	THEMEID	Heading	Table, Form, Report
USER	USERID	LastNameFirstName	Table, Form, Report

▶ To perform these modifications:

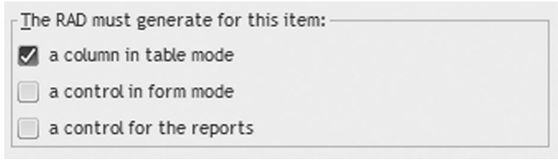
1. Select the file in the table.
2. Perform the modifications in the RAD tab.
3. Click the "Validate" button (this step is optional).
4. Go to the next file.

▶ 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:



   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 items found in the data file.

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). 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: 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. Select "Analysis .. Generation" ().
You can cancel the wizard for automatic data modification that starts 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 RAD?**
- **Generating RAD**
- **Test of the project**

 Estimated time: 20 min

What is 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.

We shall now 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 source code required for these pages to work must be written by the developer.

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

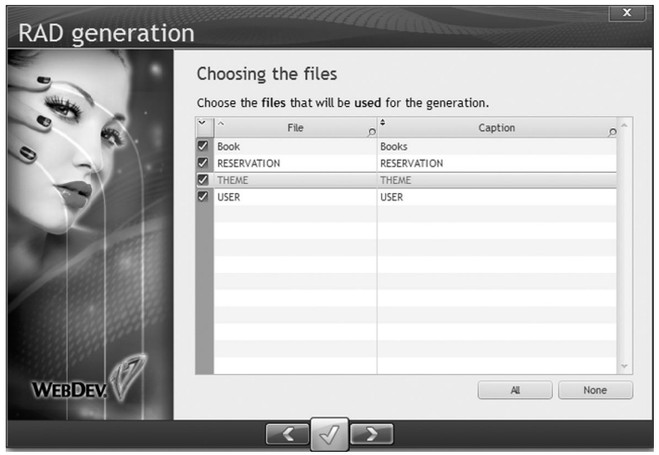
Generating RAD

- To start generating RAD:

  1. Select "Workshop .. Full Application RAD". The wizard for generating the RAD application starts.

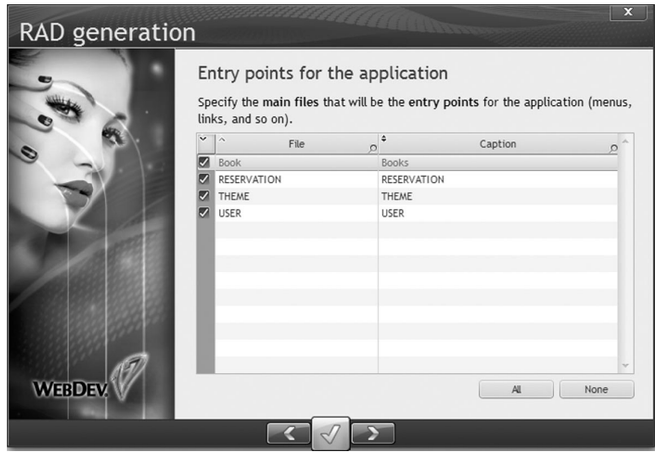
  2. Select the template that will be used for the RAD generation: "Classic RAD (Intranet/Extranet)" for example. Go to the next wizard screen.

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.

Test of the site

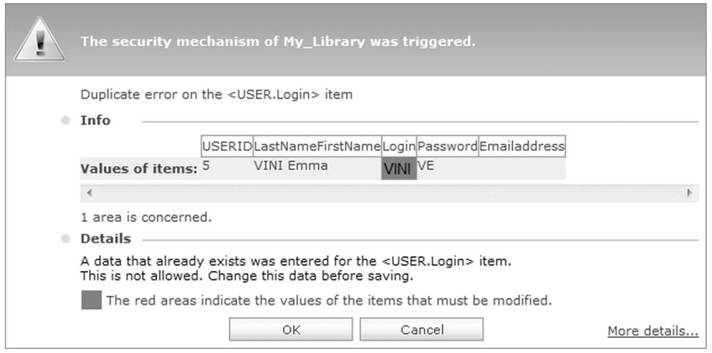
Let's now run the test of the generated site.

▶ 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 name of the user: MORGAN 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 France	TASTE	TF
VINI Hans	VINI	VH
VINI Emma	VINI	VE

9. A special window is displayed when the last record is validated:



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 the automatic management of the HyperFileSQL 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.

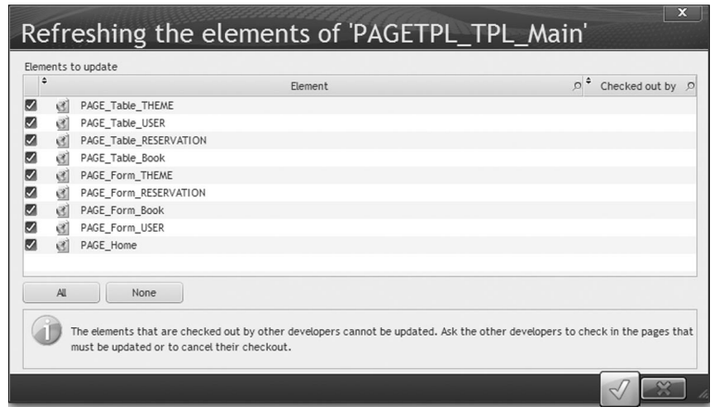
Customizing the generated site

When running 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, 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 description of the control ("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 ).
6. To update the project pages that use this template, select "Page .. 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 and run the test of the site.

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

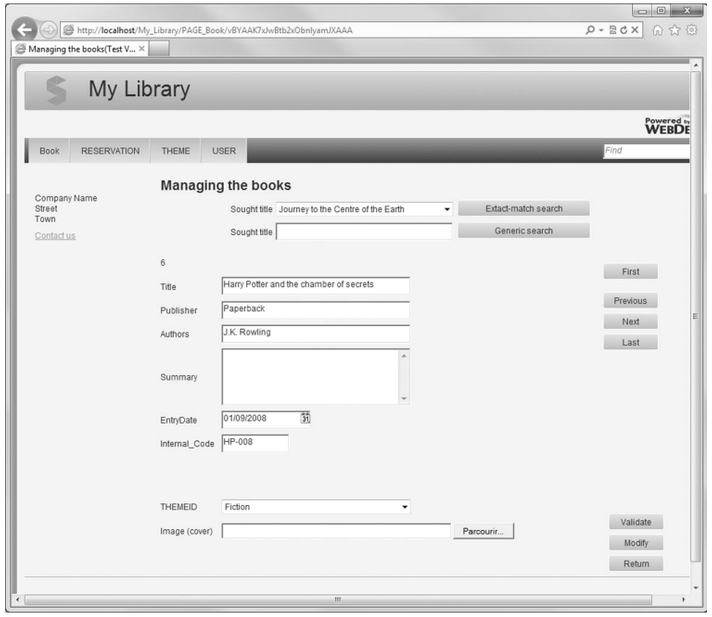


Estimated time: 50 min

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 the data files and 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 the 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:

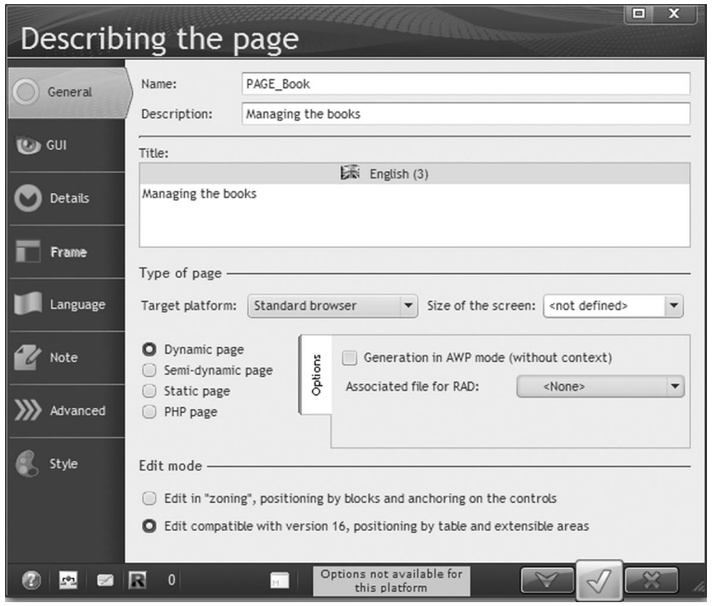
1. Create a new page ("File .. New .. Page").

2. In the wizard for page creation, select "Page" on the left of the window then select the "PAGETPL_TPL_Main" template.

3. Validate.

The new page is displayed in the editor.

4. Display the description window of the page. To do so, click the right section of the page ("Page..Description"), display the popup menu 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 the books".



5. Validate the description window and save the page.
6. The different data files described in the analysis are listed in the project explorer (last item named "My_Library").
7. 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 separator to display it (- - -).

8. 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.
9. "Drag and Drop" these items toward the page that was just created.
10. Different controls are automatically created in the page. These controls are automatically linked to the item in the data file.

To check this, display the "Link" tab found in the description of one of the controls ("Description" from the popup menu).



Close the description.

We are going to display a title in this page. By default, the title is "List of RADFileA". 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 the books" instead.
1. Click "List of RADFileA" and display the popup menu.
2. Select "Overload the control".
3. Modify the caption of the control: the new caption is "Managing the 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 ( ).

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.

Two buttons will be added to 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 add button

▶ To create the add button:

1. Create a new button ().
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:

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

Let's take a look at this code:

- The function named ***ScreenToFile*** is used to initialize the items with the values of the 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!
- The function named ***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 ().

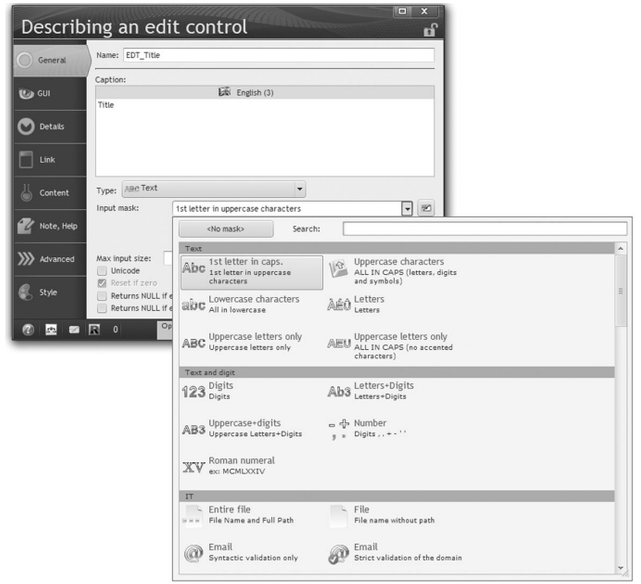
▶ Run the test of the page (). Fill the controls and click the "Validate" button.
Several remarks:

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

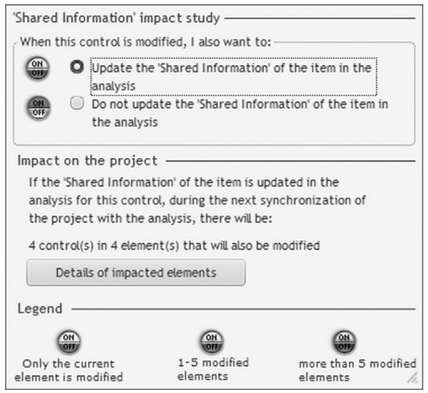
We are going to perform the necessary modifications.

Adding an input mask

▶ First of all, the input mask. An input mask will be defined for the "Title" control. Select "1st letter in uppercase characters" in the control description.



Once the mask has been modified, the appearance 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 when the analysis is regenerated.
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 input of the date will be easier. If necessary, adapt the size of the control in the editor.

Erasing the data after the addition

- We are now going to manage the erasing of the data in the controls once the "Validate" button has been used. 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 instance).
  2. Complete the code as follows:

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

The function named ***Reset*** resets all the controls for the next input. This is also used to specify to the user that the record was added.
The function named ***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 ( ). Fill the controls and click the "Validate" button.

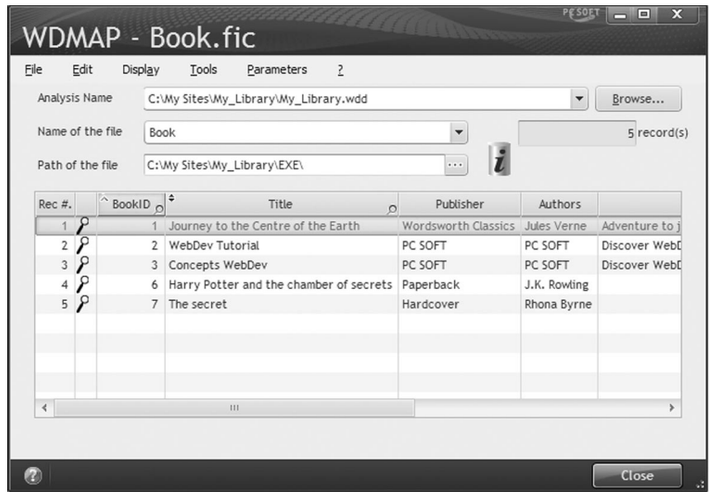
Several controls can now be entered one after another. Some examples:

Title	*Publisher*	*Author*	*Entry Date*	*Theme*
Harry Potter and the chamber of secrets	Paperback	J.K. Rowling	09/01/2008	Novel
The secret	Hardcover	Rhona Byrne	10/01/2008	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 (if 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. Select "Tools .. WDMAP - Data editor".
  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 the controls were created 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 a new edit control by clicking the arrow found to the right of . In the list of preset controls, select "Upload".
  2. Display the description window of the control, specify the name ("EDT_Cover") and the caption ("Image (Cover)"). Validate the control description.
  3. 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.
  4. 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 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:

- the function named **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.
- the function named **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.
- the function named **HLinkMemo** is used to load the image in the memo item of the HyperFileSQL "Book" data file.

▶ Let's 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 ().
2. The caption of this button is "Return" and its name is "BTN_Return".
3. 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 enables you to associate some preset actions to the buttons. To manage a simple closing button (without management of the 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 the test of the page.

Managing the search and the modification

We've managed the addition of records into the "Book" file. It's fine. 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 ().
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. Specify the name and caption of 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 ().
2. Position the button beside the combo box that was just created.
3. Specify the name and the caption of this control ("BTN_ExactMatch" and "Exact-match search").

4. Enter the following server click code:

```
HReadSeekFirst(Book, BookID, Combo_Title)
IF HFound(Book) THEN
   FileToPage()
END
```

The function named ***HReadSeekFirst*** is used to perform an exact-match search. In this example, the search is performed on the Book file and on the IDBook 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).
The function named ***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

The function named ***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 (Ab).
  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 (Ok).
  2. Position the button beside the edit control that was just created.
  3. The name of this control is "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
```


The function named ***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.

The function named **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 information displayed. Modifying the values of the 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 (Ok).
  2. Position the button below the "Validate" button.
  3. The name of this control is "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 in the screen. The function named ***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 113).

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
```

The function named ***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.

The function named ***HReadLast*** is based on the same principle, but this function reads the record with the greatest key value.

The function named ***HReadNext*** reads the record whose key value is immediately greater than the one of the current record.

The function named ***HReadPrevious*** reads the record whose key value is immediately less than the one of the current record.

In any case:

- the function named ***HOut*** is used to find out whether the data file is empty.
- the function named ***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. MULTI-CRITERIA 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

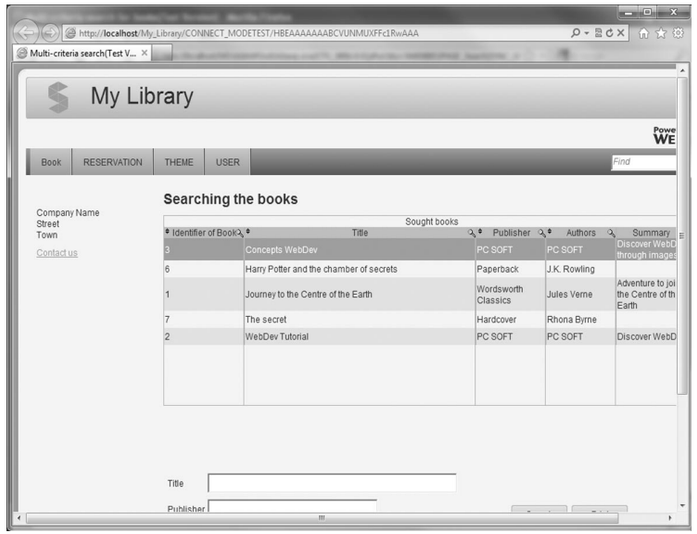
Overview

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, we have already performed a search 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:



Note

This lesson "gives an overview" of query creation. An entire section of this tutorial is devoted to the management of queries (“Create your first query”, page 273).

Creating a query with parameters

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

1. Select "File .. New .. Query". 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.

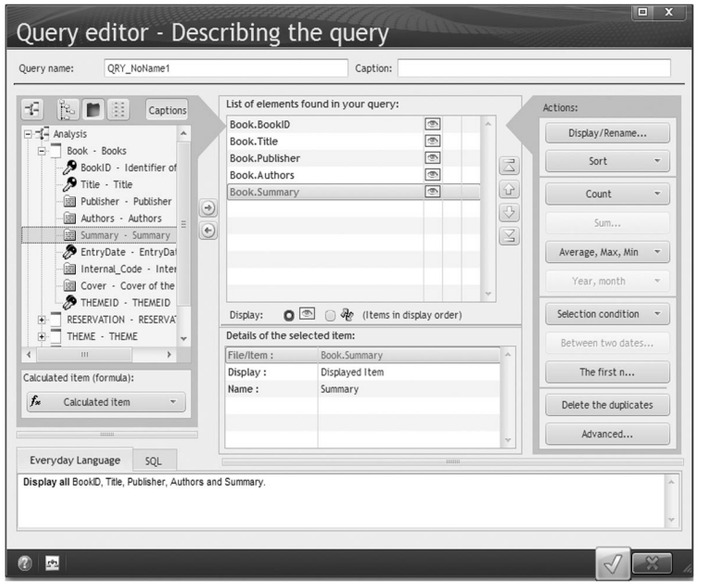
4. The list of data files appears on the left of the screen. To select an item of a data file, you must:

- 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 publisher
- the authors
- the summary.

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.
- the selection condition defined for Book.Summary is: "contains" the "Param_Summary" parameter.

8. Our query is over. Give a name to the query (QRY_Search_Book for example, top left of the screen). Validate the description window of the query.

9. Save the query.

10. Let's now run the test of the query: click  . The window that opens asks for the parameters of the query. 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 based on the query

This page will allow the user to enter some information used to perform the search for the requested books.

▶ To create the page:

1. Create a new blank page ("File .. New .. Page").

2. In the wizard for page creation, select the page template named "PAGETPL_TPL_Main" to create your page.

3. Validate. The new page is displayed in the editor.

4. Enter a name (PAGE_Search for example), a title and a description for your page ("Multi-criteria search" 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: "List of RADFileA". To modify this title in the current page, all you have to do is overload this control in our page to display "Multi-criteria search for books" instead.

1. Click "List of RADFileA" and display the popup menu.

2. Select "Overload the control".

3. Modify the caption of the control: the new caption is "Multi-criteria 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. Create a table control (). The wizard for table creation starts.

2. We are going to create a table 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: "QRY_Search_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: 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 ("TABLE_Search_Book" and "Sought books" for example). Validate.

9. The table 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
- Summary

4. Drag and Drop these items toward 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. For all the created edit controls:
- Display the "Link" tab of the control description ("Description" from the popup menu).
- Select "None" in the linked item.
- Validate.

▶ We are now going to enter the code that will be used to initialize the table. This code will be entered in a specific button.
1. Create a new button. The name of this button is "BTN_Search" and its caption is "Search".
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_Search_Book.Param_authors = EDT_Authors
QRY_Search_Book.Param_publisher = EDT_Publisher
QRY_Search_Book.Param_summary = EDT_Summary
QRY_Search_Book.param_title = EDT_Title
HExecuteQuery(QRY_Search_Book)
TableDisplay(TABLE_Search_Book)
```

This code is used to initialize the parameters of the QRY_Search_Book query with the values entered in the page controls.
Then, the query is run (**HExecuteQuery**). The table 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. An entire section of this tutorial is devoted to the management of reports (“Your first report”, page 297).

Creating a report

We are going to create the report used to print the list of books displayed in the table found in the "PAGE_Search" page.

▶ To create a report:

1. Select "File .. New .. Report". 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. Select the "PAGE_Search" page. Go to the next wizard screen.

5. Select the items displayed in the report. Uncheck the identifier of the book (BookID). Go to the next screen.

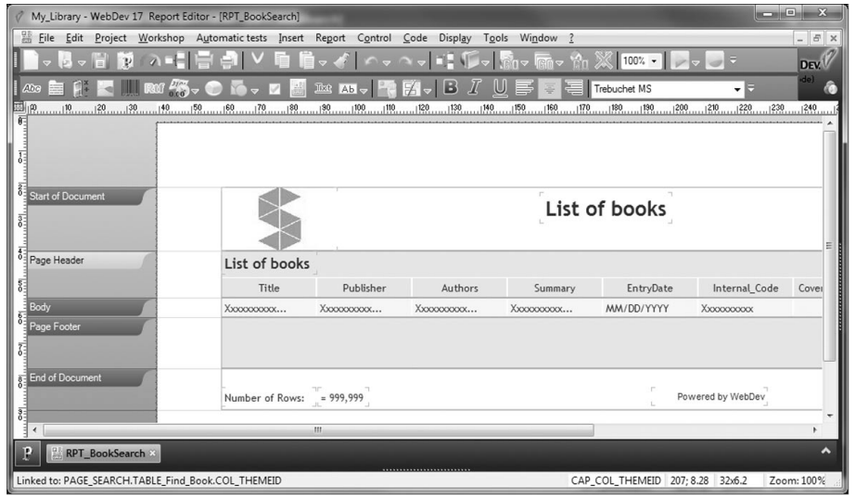
6. Keep the proposed paper format. Go to the next screen.

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 books" for example).

9. Validate. Select (if necessary) the landscape mode to display the entire table.

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 to an HTML page or 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 propose the printing of the books found in the table:

1. Open the "PAGE_Search" page in the editor (if not already done).

2. Create a button in the "PAGE_Search" page (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.
- the function named ***iDestination*** is used to define that the report will be printed in a PDF file and ***iPrintReport*** starts printing the report.
- the function named ***FileDisplay*** is used to display the PDF file generated in the browser.
- the function named ***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.

LESSON 2.7. THE USER GROUPWARE

This lesson will teach you the following concepts ...

- What is the user groupware?
- Including the user groupware
- Configuring the user groupware
- Running the test of the user groupware



Estimated time: 10 min

Overview

A site can be used by different contributors with different profiles. It is often necessary to define several access levels according to the Web user.

Let's take a simple example: when implementing an e-business site, the site proposes the following features:

- Seeing the price list
- Modifying the price list
- Entering the orders
- Managing the 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.

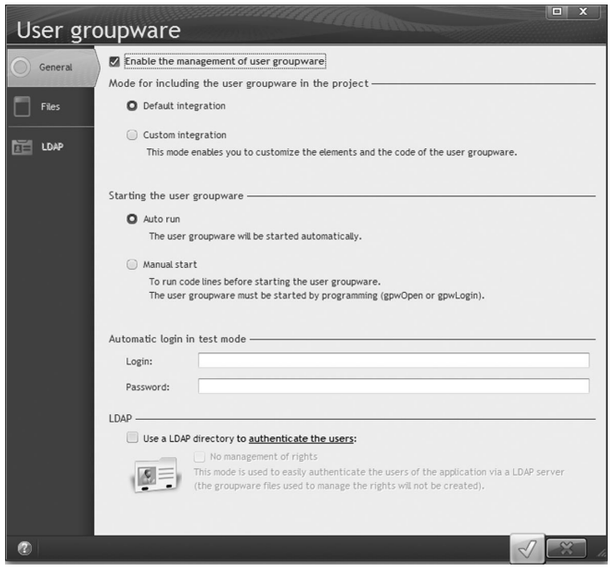
WebDev enables you to manage these different access levels via the user groupware.

We are going to include the user groupware in our site ("My library") and to configure it.

Including the user groupware

▶ To include the user groupware in the "My library" project:

1. Select "Workshop .. Configure the user groupware". The window for configuring the user groupware is displayed.



2. Check "Enable the management of user groupware".

3. Keep the default options of the "General" tab:

- Default integration: the login page is automatically included in the application.
- Auto run: the user groupware is run as soon as the site is started.



Note

The custom integration enables you to customize the pages and the source code of user groupware. This option is intended for the developers who want to translate or customize the pages of the user groupware.

If the end user uses an **LDAP directory**, it can be used to authenticate users.

When installing the site, the user will have the ability to enter the parameters of his LDAP directory.

4. Select the "Files" tab. This tab is used to define the format and the location of the data files for the user groupware. In our case, we will be using the HyperFileSQL Classic data files, in the location specified by default.

5. Select "Enable the history of connections". This option allows the supervisor to get information about the connected users.

6. Validate. The user groupware is included in the application.

▶ Let's now run the test of our site:

1. Run the test of the project ( ). A login page is displayed.



Note

A single user exists by default, the supervisor. To connect yourself as supervisor, use:

- the name: SUPERVISOR
- the password: SUPERVISOR

2. Login as supervisor.

3. A new menu is displayed, allowing you to connect to the site or to configure the site.

4. Choose "Configure". We are going to configure the user 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 of the site.

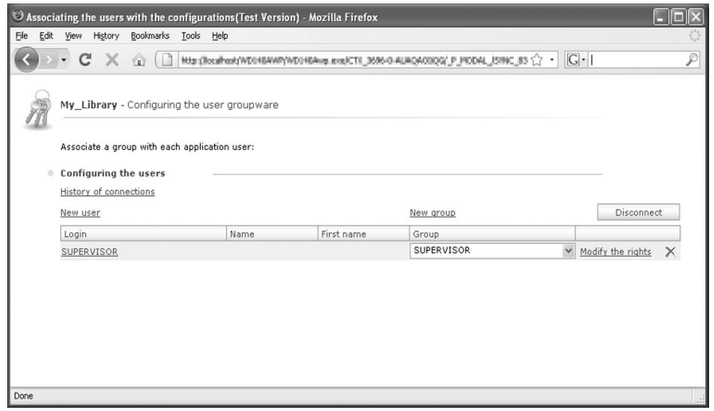


Note

The configuration of the 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.

- To configure the user groupware:



1. Create the users who will be able to access to the application ("New user" link). Only the supervisor exists by default.
Create the new TEST user. His password is "test".
2. Associate (if necessary) the users with a group. The "New group" link is used to create groups.
3. Configure the rights of the users via the "Modify the rights" button. The page that is displayed is used to select each page of the site. For each page element, you can define whether the control will have the behavior of the site (default) or whether it will be disabled, invisible or grayed.
4. Exit from the test.
5. The user groupware will no longer be used in the rest of this tutorial, so we are going to disable it: select "Workshop .. Configure the user groupware" and uncheck "Enable the management of the user groupware".

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 a main 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 353.

▶ To display the dashboard of the project (if not already done), click  .

Automatic tests

One of the most interesting features of the dashboard is to give information about the tests that were run on the project.

Several tests have already been run while developing our project.

▶ Click the Test light. A window displays the validation rate of the application. **This window is empty at the moment.**
This window returns the list of all the tests that have been run on the procedures and classes of the project. Most of these tests are automatic tests.

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 instance 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: "Set_MyLibrary" and validate.
4. In the project explorer, select the set of procedures that was just created.
5. Select "New global procedure" from the popup menu.
6. Specify the name of the procedure: "CheckInternalCode" and click "Add".

7. The procedure is displayed 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.
- The function named ***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 than 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.

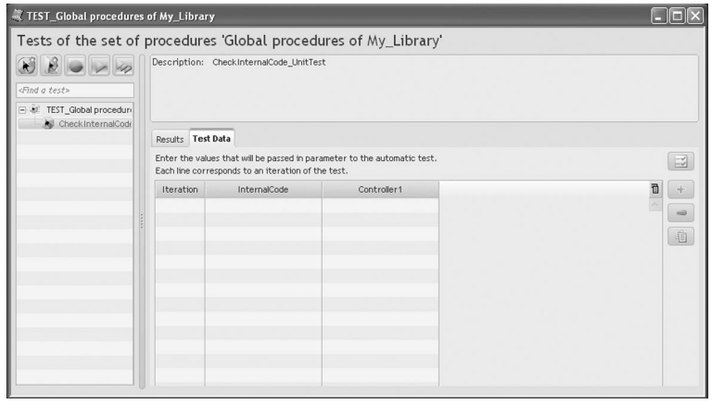
A wizard is available for several WLanguage functions.

- If ***MatchRegularExpression*** is set to True, the internal code will be validated: the procedure returns True. Otherwise, the procedure returns False.

8. 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 "Create a unit test".
3. The editor of automatic tests is displayed.



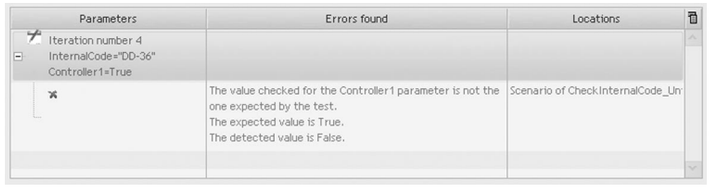
4. Three columns are found in this table:
- Iteration that corresponds to the iteration in the test. Indeed, a set of tests will be used to run the test of our procedure. Each iteration will correspond to a test.
- InternalCode that corresponds to the value of the InternalCode parameter of our procedure. 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. To do so, use the icon  . 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 iteration 4: click the "Test data" tab and modify iteration 4: the controller must be set to False.
8. Run the test again. The available test. The test can now be run at any time.
9. Close the test editor and record the test if necessary.

You have handled a simple example for creating some unit tests (also called automatic tests) in WebDev. See the online help (keyword: "Automatic test") for more details.

LESSON 2.9. REFERENCE A SITE

This lesson will teach you the following concepts ...

- **How do I optimize the referencing of a site?**



Estimated time: 20 min

Overview

In order for your site to get visitors, the Web users must be able to find it. The easiest way to find a site is to use a search engine. The Web user enters keywords and the search engine proposes all the sites that correspond to the search criteria.

Therefore, your site must be referenced by the search engines.

To help you, WebDev proposes:

- **a referencing for each static and AWP page**: a description and a set of keywords can be defined for each page. These keywords will allow Web users to access the page directly. You can also choose not to reference a page.
- **a referencing wizard**, indicating the improvements that can be performed to optimize the referencing of each page.
- **to include or to use a specific page as home page**. This solution allows you to reference the dynamic pages for instance, by associating some keywords with the home page. This home page contains all the keywords that will allow Web users to find your site.

Our example being a dynamic site that uses no AWP page, we are going to create a home page in order to reference it.

Adding a home page

The first page of a site corresponds to the Index.htm or Default.htm page. This page is the first page of the site displayed to the Web users when they enter the address of the site. Furthermore, this page is used by the various search engines to reference the site.

This home page can be created at any time when developing your site.

This page can correspond to:

- Any page of your site.
- A specific page, generated by WebDev. This page can be modified thereafter.

Let's now create this home page.

▶ To create the home page in the "My_Library" project:

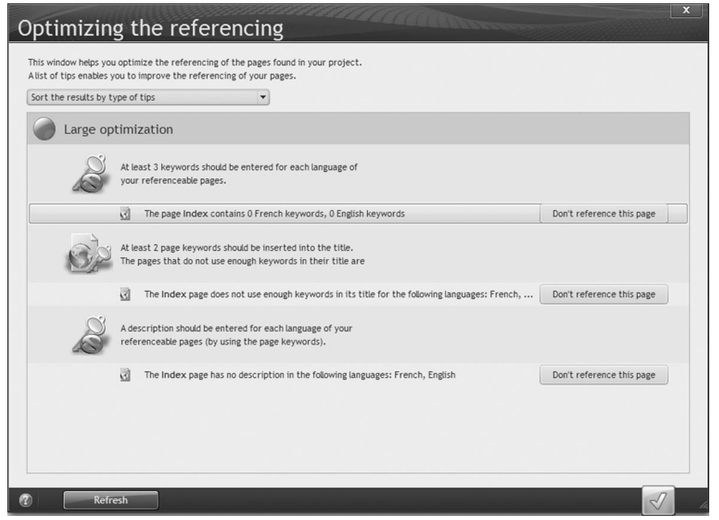
1. In the WebDev editor, select "File .. New .. Page".
2. In the wizard, select "Page" on the left of the window, then select "Home" and validate.
3. Select one of the preset home pages ("Advert_Fast" for example). Validate.
4. The home page was added to your project. This page is named "Index".

▶ We are now going to modify the home page in order to start the first page of our site.

1. Open the "Home" page (from the "Project explorer" pane for example).
2. This page contains the link named "Direct access to the site". Display the description window of the link.
3. Modify the information of this control. Change the caption and enter "Reserve your next book". In the "Action" combo box, you will notice that the link will display the first dynamic page of the project. That's what we want.
4. Validate the description window of the link and resize the control.
5. Run the test of the site by using the home page. To do so, use  .

▶ Let's see how we can optimize the referencing of this home page. We are going to start the wizard for optimizing the referencing.

1. Select "Page .. Optimize the referencing of the page".
2. If necessary, choose "Sort the results by page". The optimization wizard returns the following result:



▶ We are now going to modify the home page consequently.

1. Open the "Home" page in the editor.
2. Display the description of the page.
3. In the "General" tab, enter a description and a title ("Welcome to My_Library" for example).
4. In the "Details" tab, click "Edit the expressions and the keywords".
5. In the window that opens, enter the description in the "Description" tab ("Welcome to the My_Library" site) and enter the keywords in the "Keywords" tab (Library, Book, Reservation).
6. Validate the referencing window.



Note

A "Sitemap" file can be generated for a static, PHP or AWP site. This file allows the Google search engine to better analyze the content of the sites.

7. In the "Language" tab, delete the French language, our site being available in English only (click "Synchronize with the project").
8. Validate the description window.

The "My_Library" site is now ready to be deployed. This deployment will be presented in "Deploying a dynamic site", page 425.


PART 3
Pages and controls
DEVELOP 10 TIMES FASTER
PC SOFT

LESSON 3.1. THE EDITORS

This lesson will teach you the following concepts ...

- The editors
- The panes
- The shortcuts
- The environment configurations
- The custom menu
- Customizing the wizards



Estimated time: 10 min

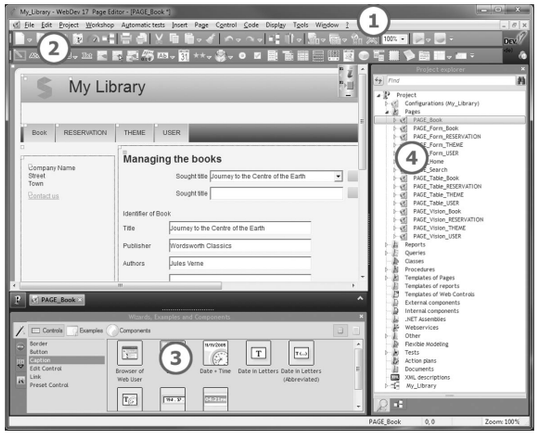
The environment

In the first lessons of this tutorial, we've created and handled pages. These operations have been performed in the page editor.

The page editor enables you to easily create outstanding pages, linked (or not) to the data.

WebDev contains several editors, all included in the same environment: code editor, query editor, report editor,...

All the editors use the same environment:



1. Menu bar
2. Toolbars
3. Panes
4. Project explorer

A few tips to help you fully benefit from the WebDev environment.

The panes

Several panes are available in WebDev. These panes can be displayed or not, grouped in suitable order. You have the ability to define the configuration that best suits your environment and to restore it at any time.

The different panes

Several panes have already been used in this tutorial:

- the project explorer, that is used to easily access the different project elements. The project explorer can also be used to perform a search in all the project elements, to easily create new elements,...
- the "Wizards, controls and components" pane. This pane gives direct access to:
  - a set of preset controls that can be directly used in your pages,
  - full or unit examples supplied with the product.
  - the components supplied with the product and your own components,...

WebDev proposes several panes. To display the full list, select "Display .. Toolbars".

Handling the panes

To display the panes:

1. Select "Display .. Toolbars .. Panes .. ".
2. Select the pane to display or to hide (a checkmark is found in front of the name of the panes currently displayed in the editor).



Tip

A specific toolbar is available for managing the panes. To display it, select "Display .. Toolbars .. Checking the panes".

Each pane found in the interface can be:

- ***floating***: the pane can be resized and moved independently. The floating panes are convenient if you are using two screens.
- ***grouped with other panes***: all you have to do is move a pane toward another pane. When the icon representing the tabs is displayed (), position the pane over this icon and release the mouse button.
- ***docked to a side of the interface***: all you have to do is move a pane toward a side of the interface. When the icon representing one of the sides of the interface is displayed (), position the pane over this icon and release the mouse button.
- ***invisible***: if the pane is floating, all you have to do is click the closing cross (X). If the pane is grouped, all you have to do is right-click the corresponding tab and select "Close the tab".

The environment configurations

The configuration of your panes suits you and you want to store it? Nothing's easier: up to 4 configurations can be saved with [Ctrl] + [Alt] + [1], [Ctrl] + [Alt] + [2], ... To display the requested configuration, press the [Ctrl] key followed by the number of the requested configuration.



Tip

[CTRL]+[W] is used to display or hide the grouped and anchored panes.
This instantly gives you a larger workspace.

The custom menu

Who has never wanted to have their "own" custom menu?

WebDev proposes it as a standard feature.

Let's take an example to understand the operating mode.

We are going to create a custom menu used to:

- display the calculator of Windows
- display the project description.

Note

Required configuration

To use the "Custom-Menu", this feature must be re-enabled.

To do so, perform the following operations:

1. Select "Tools .. Options .. Options of the environment".
2. Check "CustomMenu" .
3. Validate your choice.

- First step: we are going to create the menu option used to start the calculator of Windows from the WebDev editor:
  1. Select "CustomMenu .. Customize <CustomMenu>". The wizard for creating your custom menu starts.
  2. Click the "Add an option" button.
  3. Enter its name in the "Menu caption" control ("Calculator" for example).
  4. Click the "Browse" button and find the executable of the calculator (Calc.Exe in "C:\Windows\system32" for example).
  5. Enter the message that will be displayed in the status bar when the option is hovered by the mouse cursor ("Starts the calculator" for example).
  6. Select the icon that will be displayed to the left of the option in the menu: to do so, select the executable of the calculator once again in the "icon" control.

- Second step: we are now going to add a second option used to directly start the "<Project Name>.RTF" file that was automatically created when creating a new project. This file contains the project description defined in the wizard for project creation.
  1. Click the "Add an option" button.
  2. Enter its name in the "Menu caption" control ("Project description" for instance).
  3. This file is found in the directory of the current project. Click the "Macro" button and double-click "{$ProjectDir} Path of the current project directory".
  The "{$ProjectDir}" string is automatically displayed in the "Command script (DOS commands)" control.
  4. Enter the "\" character at the end of this string.
  5. Click the "Macro" button and double-click "{$ProjectName} Name of the current project".
  The "{$ProjectName}" string is automatically displayed in the "Command script (DOS commands)" control.
  6. Enter the ".RTF" characters at the end of this string.
  7. Enter the message that will be displayed in the status bar when the option is hovered by the mouse cursor ("Opens the project description" for example).
  8. Validate the window for configuring the custom menu.

- Your custom menu is created. Run its test!

Customizing the wizards

To go even further in the customization of your work environment, WebDev enables you to customize the different wizards proposed by the product. How?

When using a wizard (wizard for creating a Combo Box control for example), all you have to do is click the image with the right mouse button. Several images are displayed: all you have to do is choose the most convenient one.



LESSON 3.2. 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:

- creation of blank pages based on a template (part 1 of this tutorial)
- creation of different types of pages (form, table,...) based on the 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, started via "File .. New .. Page".

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 looper, ... 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 contain the controls and the buttons only as well as 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.

Description of a page: 7 tabs are available

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 a sample project created for this purpose.

1. Start WebDev 17 (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 project named "Pages and controls". The project is loaded.
   Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. Pages and controls".
4. Open the "PAGE_Button" page.



Notes

Opening a page in the editor

Several methods can be used to open a page in the editor. Summary of the different methods:

- "File .. Open" and select the file to open.
- Press [CTRL] + [E] and select the page to open. The benefit of this method is the preview of the page.
- In the "Project explorer" pane, double-click the name of the page to open.

5. Display the description of the page ("Page .. Description"). The tab displayed by default is the last tab displayed.

"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 (or "**caption**") 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 use 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-ROM 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 contains 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 type of the page, options can be entered in the "Options" tab.

- The target platform and the type of 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 **edit mode** of the page. Two edit modes are available:
  - the edit in zoning mode: this edit mode allows you to use the simplified anchoring (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 **position of the controls**.
  - If "Fix the controls in the page" is checked, the controls will not move when the browser window is resized. Otherwise, the controls will be moved according to the size of the browser.
  - If "Center the page in the browser or in the frame" is checked, the entire page will be centered in the browser window or in the frame containing the page (horizontal centering).
- The cursor of the page.

"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 for more details (keyword: "Ajax").

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 the pages.
- The independent HyperFileSQL 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 413 for more details

"Note" tab

This tab enables you to enter 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 can:

- 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***
- optimize the generated HTML code (the optimization used for the page may differ from the one used for the project).

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 "Devices" tab is used to define the parameters that will be used when generating a site for iPhone or a site for BlackBerry.

"Style" tab

This tab groups the parameters regarding the style of the page. You have the ability to select:

- The skin and the color palette.
- The background image of the page and its display mode.
- The background color.
- The default color of the text.
- The color and the style of the page links.
  These two last options can be modified control by control.
- The WebDev style sheet associated with your page (".WWS" file).

Processes associated with a page

In the page editor, the "Code" option (from the popup menu of the page) gives direct access to the code editor. The different processes linked to the page are displayed.

Global declarations (dynamic pages and PHP pages only)

Run on the server when the page is opened, before this page is returned to the browser. This process contains the declarations of the global variables, ...

Initialization (dynamic page and PHP page only)

Run on the server when the page is opened, before the page is returned to the browser but AFTER the initialization process of the page controls.

Load (dynamic and PHP page, semi-dynamic and static)

Run in the browser when the page is displayed in the browser of the Web user. This process can contain some calls to the browser functions, ...

Unload (dynamic and PHP page, semi-dynamic and static)

Run in the browser whenever the page is unloaded, which means when it is replaced by another page or when the browser of the Web user is closed.

Synchronization (dynamic page only)

Used to manage the synchronization of the page on the server when using the browser "Back" button. See the online help (keyword: "Synchronization") for more details.

Closing (dynamic page only)

Run on the server whenever the context of the page is destroyed on the server (end of session, ***PageDisplay*** used in server code,...).



Notes

MyPage is a keyword that returns the name of the page that contains the current process (where **MyPage** is called). It can be used in server code and in browser code, in all the processes of the page controls.



Tip

Other specific processes can be added via the toolbar at the bottom of the code editor:



In addition to the standard processes for mouse and keyboard management, two other processes can be used for the pages ("..." button):

- **Reset form** ("onreset" browser code): Run when a "Page reset" button is clicked or when ***Reset*** is used in browser code.
- **Submit form** ("onsubmit" browser code): Run when a "Submit to server" button is clicked or when ***PageSubmit*** is used in browser code. This process can be used to check the data entered in the page and to prevent sending data to the server with **ReturnToCapture**.

LESSON 3.3. 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

Several WebDev controls allow you to communicate with the Web user. 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 values entered 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 will be using a sample project created for this purpose.

1. Close the current project if necessary. The home window is displayed.
2. In the home window, click "Tutorial" and select the project named "Pages and controls".
   Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. 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
- Edit control
- Button
- Link
- Image, clickable image
- Radio button, check box
- List box
- Combo box
- Table
- Looper
- TreeView
- Chart
- Cell
- Rating

These controls will be presented in details.

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 any control).

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,...

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_Retrieve" ("Retrieve" button), the syntax is:

```
Str is String
Str = "Content of static control "+ CR + STC_Retrieve
// or
// Info("Content of static control "+ CR + STC_Retrieve)
```

You can even 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 enable you 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 WebDev (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. Now let's take a look at their operating mode.

Regardless of the information to enter or to 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 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 why masks are useful when developing the site named "My_Library".



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. 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 the 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 button (server or browser) 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
// Retrieves the content of the numeric control
Value= EDT_Numeric
```

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, 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 may be useful:

- **Max input size:** Maximum number of characters that can be entered or assigned into 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 "memo text" 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 contains 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 multi-criteria searches.

"GUI" tab

The GUI tab is used to define the characteristics regarding the interface of the control.

The initial status of the control corresponds to the status of the control when displaying the page. This characteristic is available for nearly all the types of controls.

When running 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 the 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 with **..State** (for the "Active" and "Read-only" options) and with **..Visible** (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 position on the control by typing a combination of keyboard keys ([ALT] + [Letter] for instance).
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) enables you to "group" several controls together in order to be able to modify the status of these controls in a single operation.

"Details" tab

The Details tab is used to define the characteristics regarding themanipulations 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 controls of the page. 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.

Euro management: 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.The process associated with the button will be run when the button is enabled (when the button is clicked by the Web user).

Several "styles" of buttons are available in WebDev:

- **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 ("Style" tab of the button description).
- **Three-state button:** normal status, rollover status and pressed status. All you have to do is associate the button with an image containing the three states of the button ("Style" tab of the button description).
- **Standard button** ("text" button). The appearance of the button is defined in the style sheet of the project.

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 the controls to the server (submit)
2. Reinitialize the page controls (reset)
3. No specific operation

"Send the value of the controls to the server" button (submit)

The "Send the value of the controls to the server" button are the most common ones.

When a click is performed 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 "Send to the server - (Enter)" button. The message corresponding to the browser code ("Send to the 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 when some data coming from a database must be displayed in the browser.

Button whose type is "Reinitialize the page controls"

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 instance.

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 catalog of WebDev (in the "Cliparts" or in the "Graphic buttons" for example).



Tip

We advise you to:

- give no caption to a graphic button whose image contains 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 = FolderWeb() + "/VSXP15022_MODIF.gif"
```

Note: ***FolderWeb*** is used to find out the path of the directory containing the images in browser code.

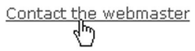
Type of control: Link

(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 when the link is clicked (display a page, perform a process, open the messaging software of the Web user,...).

In most cases, the links are used to allow the Web user to send an email to the site manager, to open a new browser toward an Internet address, ...



The content of a link and its action can be defined in the editor when the link is described or by programming (WLanguage property named **..URL**).

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 when an image is clicked.

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 does not change.
  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 code 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.

If "Locate the images in the _WEB directory in browser code" is checked ("Details" tab), there is no need to use ***FolderWeb*** in browser code to localize the path of the image.

The "Catalog" button is used to access the image catalog supplied with WebDev (“The image catalog”, page 166).

Display mode ("General" tab)

WebDev enables 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 may 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.
Homothetically extended		The image is proportionally enlarged so that the smallest side of the image is entirely displayed in the image control
Homothetically 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.
Homothetically 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 increasing size		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 increasing size		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 when a click is performed on the image ("General" tab)

When a click is performed on an image, you have the ability to trigger an action (run the code of a button for instance).

The "Other actions" button is used to define some preset actions when a click is performed on the image.

You also have the ability to choose the target of the action (new browser, current page,...).

Other characteristics

You can also define:

- **the border of the image** (outline) in the "Style" tab. You can 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 the height of the image ("GUI" tab).
- the **alternative text** ("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

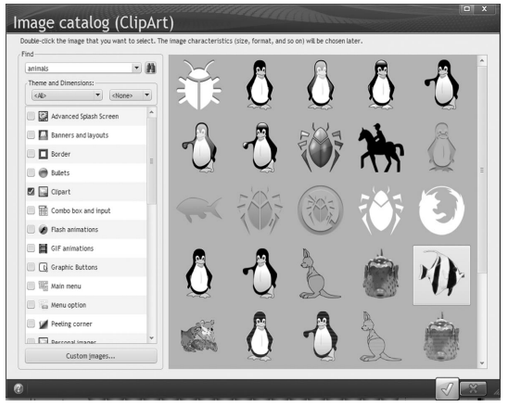
To improve the referencing of your pages, an alternative text should be entered in all your Image controls.

The image catalog

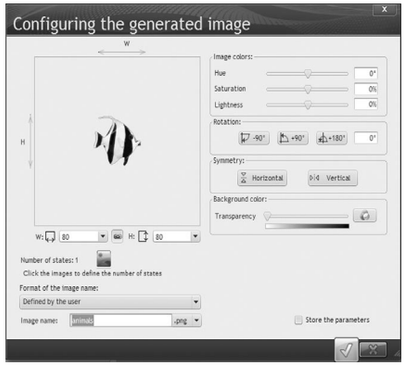
As soon as an image can be displayed in a control or in a page, WebDev proposes to use the image catalog via the "CATALOG" button. This catalog contains hundreds of images, cliparts,...

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 instance)
- 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 fish 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

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 can:

- define the position of the mouse when a click is performed 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 available for 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 is used to define the type of clickable image to use:

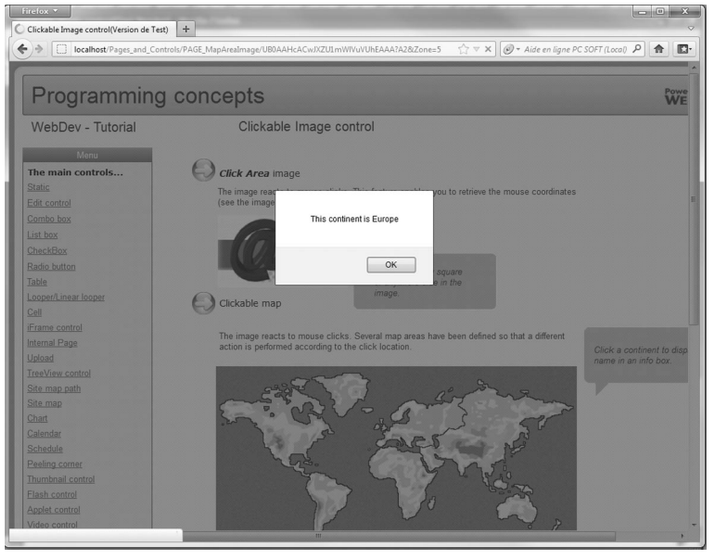
- ***Clickable image***: the image is a simple clickable image. An action can be associated with the click performed on the image. However, you cannot retrieve the coordinates of the mouse or define click areas.
- ***Click area*** (dynamic page only): the image is a clickable image and the coordinates of the mouse can be retrieved (***PageParameter***).
  You will find an example of server code used to retrieve the coordinates of the mouse when a click is performed on a click area.
- ***Map area:*** clickable image with definition of clickable map areas. An action can be associated with the click performed on the image and on each graphic 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.

- Test page "PAGE_ImageMapArea":



- 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 will 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. To do so, select the value of the zoom in the relevant combo box found in the toolbar of WebDev.



Tip

If you choose to draw a "polygon" area, some points can still be added even when 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).



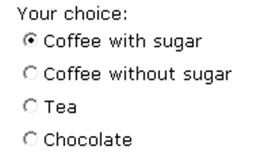
Several map areas can be defined on the same clickable image. Different processes can be run according to the location clicked by the Web user.

Type of control: Radio button — ***(static page, semi-dynamic page and dynamic page)***

Summary

The 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).





Tip !

How do I differentiate a radio button from a check box?

An easy way to remember: think of the old radios: a single frequency could be selected via the button!

The radio button allows you to select a single option.

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 the 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
```

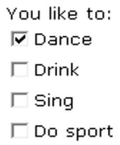
Type of control: Check box

(static page, semi-dynamic page and dynamic page)

Summary

The check boxes are used to choose whether the value of each option is "True" ("box checked") or "False" ("Box cleared").

The caption of the options must be entered when describing the check box in the editor. It can be modified by programming (server code).



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 the check boxes

Each option found in a check box is "identified" by a subscript (that can take a value from 1 to N, N being the total number of options 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 that "Drink" is checked
- "CBOX_Choice[3]=False" means that "Sing" is not checked

When several options are found, the check box operates 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 the [ ] 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 the option 2 "Drink"
```

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 handling boolean expressions may omit 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.

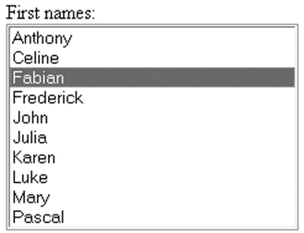
Type of control: List box



(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).



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 proposed directly. 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 the 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. The code corresponding to each button is displayed in the page.
- 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 [Enter] 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 memory 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 the control is created, you have the ability to create the query at the same time as the control. The query will be embedded into 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 sorting 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 practical example, see the "Content" screen of the "Images" list box to check the characteristics of the browsing list box used.



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 displays the content of the data file or query directly. 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 practical example, see the "Content" screen of the "Month" list box to check the characteristics of the list box on source used.

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 of 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 in a list box (server code):

- the function named ***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 fist 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. The function named ***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, use the name of the list box to find out 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.

```
// Rows selected in a multi-selection list box
i is int = 1
SelectedRow is int
SelectedRow = ListSelect(MultiList,1)
WHILE SelectedRow <> -1
  Trace("The row #"+ SelectedRow +" is selected")
  i++
  SelectedRow = ListSelect(MultiList,i)
END
```

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, use ***ListDelete***.

To delete all the values from the list box, use ***ListDeleteAll***.

```
// Deletes the selected element
ListDelete(LIST_Memory)
```

Search (server code/browser code)

To find an element in a list box, 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 will be applied 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 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. The code of the different buttons is displayed in the page.
- 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 many other interesting options.

The "Details" screen is used 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".

The term of **Table** refers to 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 may 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_Menu,"Monday","Scrambled eggs",...
    "Irish stew", "Vanilla ice cream")
```

Table linked to a data file or a to a query

Various 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 the control is created, you have the ability to create the query at the same time as the control. The query will be embedded into 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 row of the table.
- 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 elements of the file 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 many other interesting options.



Notes

The description window of a table contains two sections:

- the upper section, presenting the name of the table, the name of the 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 useful 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:

- Adjustable by the user (via the mouse)
- with search: a magnifier will be displayed, allowing you to search for a value in the column.
- sortable. Arrows will be displayed in the title of the 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 used.

- A memory table can be handled row by row or column by column.
- A memory table can be editable 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 useful functions for handling 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. Different types of information can be displayed in the looper controls. This enables you to display, in each row, 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.

Several characteristics can be modified when the looper controls are repeated: value, caption, color of the text, 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 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.

LooperAddLine must be used to add a row into a looper (server code). 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_Product, ...
    "Product #1", "shirt1.png", "Blue shirt." + CR + ...
    "Red, orange and green patterns" + CR + ...
    "Available in several sizes", 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 sorting when filling the looper, so it is used to specify the order in which the values will be displayed in the looper. For example, a looper can be used to display:

- the photo of a product
- the description of a product
- the price of a 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: TreeView  *(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:

- the property named **..Color** to modify the color of a row,
- the property named **..Brush Color** to modify the background color of a row,
- the property named **..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)

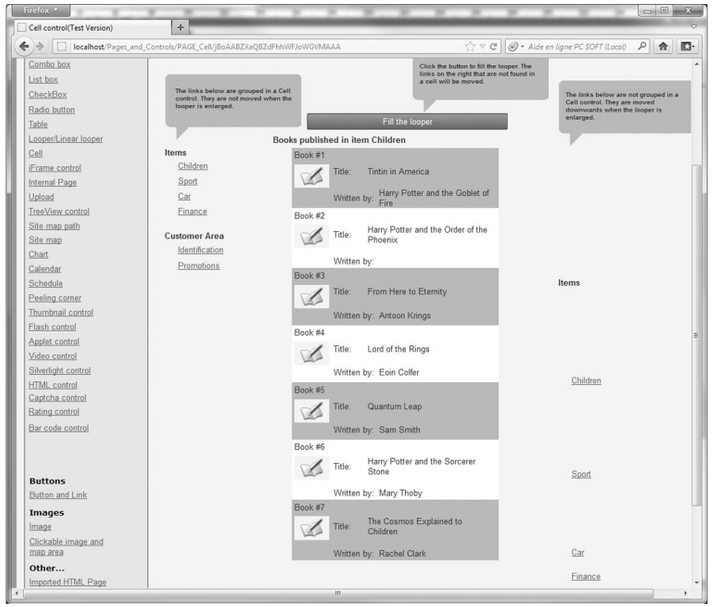
Summary

A cell is used to avoid the unexpected effect when moving a group of controls downwards when a looper or a table is located beside a group of controls.

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. This page presents several links positioned beside a looper.
  The links positioned on the left are grouped in a cell, the links positioned on the right are not grouped in a cell.
  When the looper is resized (switch from page 1 to page 2 via the pager), the controls positioned in the cell are not moved.



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 allows you to easily 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 mark.

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 specific features. These controls are as follows:

- Calendar
- Scheduler
- Captcha
- Bar code
- HTML table
- Horizontal rule
- 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 "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.

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: Captcha *(static page, semi-dynamic page and dynamic page)*

Summary

The captcha controls are the strange texts that must be read and entered by the Web user 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 a security in this type of 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 may contain several control 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 will not go into details about HTML tables in this tutorial. See the online help (keyword: "HTML table") for more details.

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 will not 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 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: Check whether all the files required for the applet to operate (".class" files, images, and so on) are found at the same location as the applet itself.

Practical example

- Open the "PAGE_AppletControl.WWH" page and run its test.
- Go back to the editor and display the description of the control ("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

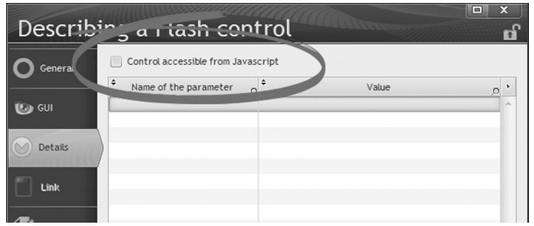
- Open the "PAGE_FlashControl.WWH" page and run its test. This page presents how to use the Flash control and how this control can interact with the WebDev page.

- Go back to the editor and display the description of the control ("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.
- **Flash 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 (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 will not 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 in 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 will not 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 will not 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 will not 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) may 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):

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 looper:



In this example, 32 records have been added to a looper. The maximum number of rows per page is set to 4. A pager was associated with the looper in order to display 4 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. When a click is performed 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 will not go into details about this type of control in this tutorial. See the online help (keyword: "Thumbnail control") for more details.

LESSON 3.4. ERGONOMICS OF THE SITE

This lesson will teach you the following concepts...

- Improving the ergonomics of the 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.

Handling the styles of the controls

A style is a set of parameters (colors, fonts,...) affecting the appearance of a control. A set of styles is associated with each type of 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 the CSS extension (Cascading Style Sheet).

Each WebDev project is associated with a style sheet (CSS file).

When developing a site, we advise you to create all the necessary styles. This creation can be performed:

- before creating the pages of the site
- whenever required.



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. The WebDev styles are proposed by default. These styles group a set of CSS styles.
- use the CSS styles directly. If several controls of a project use the same CSS style, the style modification performed from a control will be applied to all the other project controls that use this CSS style.



Caution !

If the style is modified for the control, then it is overloaded. When the style is updated, the control will keep its overload.

If the style is dissociated, only the style of the current control will be modified. This style will no longer be part of the CSS style sheet but it will be included in the HTML page. 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 page load).

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 (.CSS file) from a project to another one.

The style sheet to use is specified in the project description ("Project .. Project description", "Skin/Styles" tab).



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 the controls

The tab order of the 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, select "Page .. Tab order .. Define automatically".
- To edit the tab order of the controls, select "Page .. Tab order .. Edit":



You also have the ability to let the browser manage the tab order.

- To define a tab order by selection:
  1. Select the controls in the requested tab order while keeping the [CTRL] key down.
  2. Select "Page .. Tab order .. Define from selection".

Keep the [F5] key down to view the tab order of the controls.

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"
```

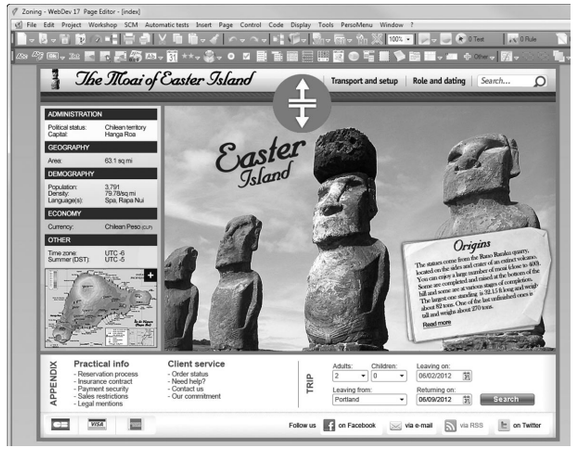
Create your pages by using the zoning mode

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: Enlarging a header zone automatically moves the zones located below.





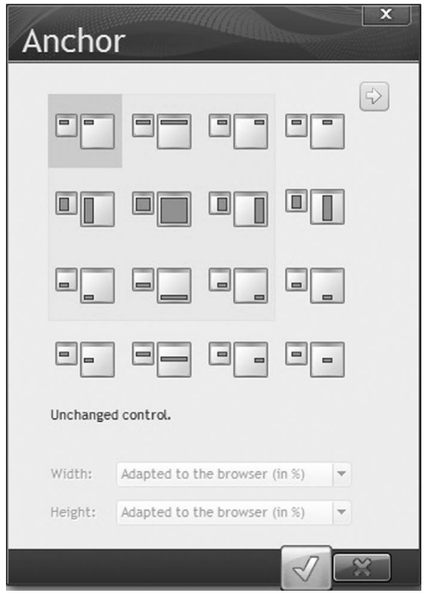
The zoning mode is used to easily design the pages and it brings several features (among which the anchoring of controls).

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 style 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 were displayed 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, select "Display .. Options .. Modify the options" and select the "Magnetism" tab.

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 (once they have been moved for example). You have the ability to use the alignment options. These options are grouped in a toolbar. To display it, select "Display .. Toolbars .. Alignment".



To use the features of this toolbar, 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.

If none of the preset alignments suits you, you have the ability to define a custom alignment: the icon in the toolbar 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

You 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 ( ).
2. Create as many panes as necessary and give a caption to each pane ("General" tab of 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 to the tab.
   - Right-click the control to associate and choose "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, so the active pages can be easily viewed 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.

Interface of the site: Use the page templates

The first purpose of a site is to provide a service (e-commerce, search, and so on) 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.

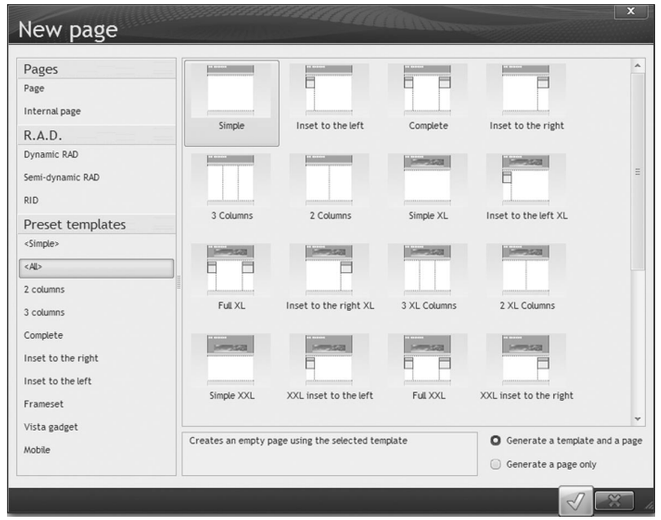
What is a page template?

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 pages of a site. For example: a logo, an identification area, a link area, menus in tab format or drop-down menus, ... A page template enables you to implement a style book.

A page template can contain some 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 already been handled in part 2 of this tutorial (see “Customizing the generated site”, page 105).

How do I create a page template?

- To create a template:
  1. Select "File .. New", hover "Page" and click "Page template". Select the source of the template and validate.
  2. The template is currently created. Create the controls, images, procedures ... required by the template.
  3. Save the template and specify its name.



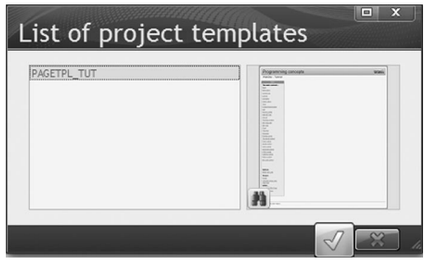
Tip

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. Select "Insert .. A template (Inheritance)". The following screen is displayed:



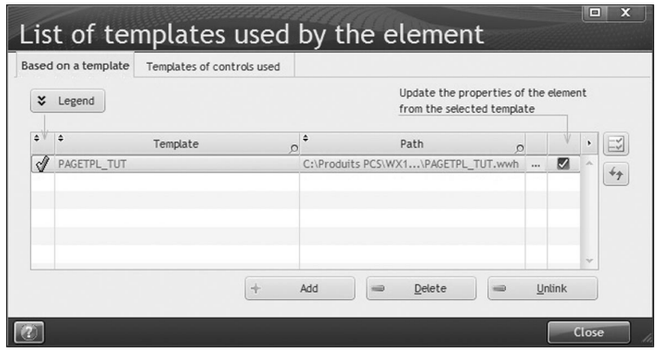
  3. Select a template to import and validate.
  The controls, the code of the controls and the procedures of the template are imported (a small yellow square is displayed on the left side of each template control).
  4. The controls, codes and procedures of the template cannot be modified from the page 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

All the pages associated with the template can be updated whenever the template is modified.

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. Select "Page .. List of templates used". A selection window is displayed:



  2.Select the template that must be dissociated and click "Unlink".
  3. Validate. The template controls are free 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 WebDev 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 3.5. 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 appearance of a site, were presented int he 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 one of these tools to help you choose the most suitable one.

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	**Creating a supercontrol:** 1. Create the supercontrol (S). 2. Include the controls in the supercontrol. **Refactoring**: Select the controls and select "Control .. Refactoring .. Create a supercontrol from the selection".
Reusability	Via the dictionary
Update	Via the dictionary (via the subscription mechanism)
Teamwork	Sharing the dictionary by network. 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	**Creating an internal page:** 1. Create the internal page ("File .. New .. Page .. Internal page") 2. Define the controls and the code. 3. Save. **Refactoring**: Select the controls and select "Control .. Refactoring .. Create an internal page from the selection".
Reusability	Via the "Internal page" control.
Update	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	- No improvement is allowed in the interface: no overload, no move of controls. - Rectangular area.

Example: Page whose interface will never change regardless of the project: option page.

Practical example

- Open (if not already done) the "Pages_and_controls.WWP" project ("? .. Tutorial .. Pages and controls").
- 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	**Creating a control template:** 1. Create the control template ("File .. New .. Page .. Control template") 2. Define the controls and the code. 3. Save. **Refactoring**: Select the controls and select "Control .. Refactoring .. Create a control template from the selection".
Reusability	Via the "Control template" control.
Update	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 (if not already done) the "Pages_and_controls.WWP" project ("? .. Tutorial .. Pages and controls").
- 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 3.6. 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 window of the browser:
- 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("WWSite", "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 "Default target" 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
```

The function named ***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 groups of controls.

Question How do I create a vertical menu in a WebDev page?

You now have the ability to display the menu vertically in the menu editor.
To access this feature, select "Insert .. Menu".

▶ To create a menu:

1. Create a new page if necessary.
2. Select "Insert .. Menu". A wizard for menu creation is displayed, allowing to choose the style of the menu to create. Select "MENU_Btn_V". 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 to 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 standardize the appearance of my pages?

To standardize the style of the project pages, all you have to do is create a page template ("File .. Save as template") and use this template for all the pages found in your project.
See the online help (keyword: "Page templates") for more details.

Question **How do I add a popup menu to a table or to a looper?**

A popup menu can be added to a table or to a looper. 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 Global Declarations code of the page, the syntax of the WLanguage code to enter is as follows:

```
-- 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.

PageParameter enables you to find out the value of a parameter passed to the current page.

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, 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?**

"Control .. Swap .. Radio Button/Check Box" is used to swap these two types of controls.

Question **How do I align the controls?**

Several methods can be used to align the controls:
- The alignment rulers ("Display .. Rulers").
- The real-time interface checker.
- The alignment options ("Display .. Toolbars .. Alignment").

The rulers are enabled by pressing [CTRL] and [R] simultaneously. "Markers" (or "tabulation marks") can be defined on the ruler and moved (by clicking the ruler at the requested location). Then, when the controls are moved in the page, they will be "magnetized" when they come near these markers.

The real-time interface checker is enabled by default. To disable it while moving a control, all you have to do is press the [SHIFT] key.

The alignment options are used to align, position or resize the controls according to other controls.



After practising a few minutes, you will soon realize what a good thing proper alignment is!

Question **How do I give the same size to the buttons?**

- Select the button whose size (width and height) will be used as reference, then the other buttons to resize.
- Select "Same width, same height" in the alignment tools ("Control .. Alignment").

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").

"Fixed background image" enables you not to scroll the image if the page cannot be entirely displayed in the browser.

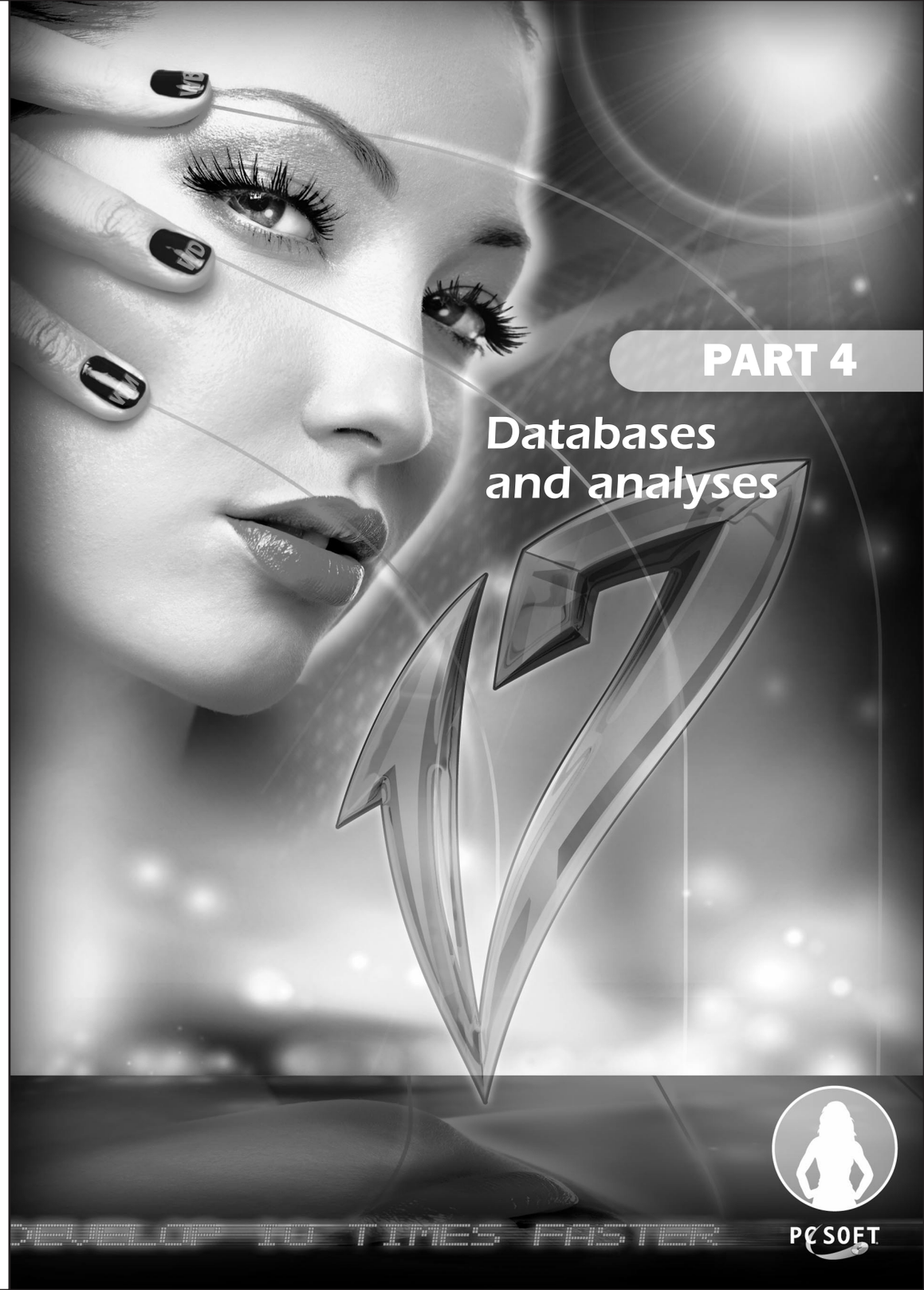
Note: A larger image will increase the time required to display the page.

Question **How do I display the date of site update?**

To automatically display the date of site update:

1. Create a formatted display control ("Insert .. Control .. Formatted display").
2. Enter the following code in the initialization process of the control:

```
MySelf = fDate(CompleteDir(fExeDir())+...
               "<Library Name>.WDL", "", fModify)
```

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 HyperFileSQL databases are supplied with WebDev. They can be accessed from the HyperFileSQL Control Center.

Several database formats are supported by WebDev. The most common ones are:
- HyperFileSQL, the database system supplied with WebDev. The HyperFileSQL 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:
- HyperFileSQL 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 you development computer by selecting "ODBC administrator" (or ask the site manager for the deployment computer).

Caution, not all the databases are necessarily 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.

Which access should I use?

- To get more information about the different database access modes and to find out which type should be used, select "? .. Help for using the databases" from the WebDev menu.

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

HReadFirst is used to browse a data file 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 browse loop is used to test the value of **HOut**. The function named **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

HReadSeek 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). The function named **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 test the value of **HFound**. The function named **HFound** is used to find out whether a record corresponding to the condition was found.



Note

The function named **HReadSeek** performs a generic search by default: 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 entirely browse a data file, a query or a HyperFileSQL view.
The browse can be performed:

- on the best key of the data file, automatically detected by the HyperFileSQL 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 reading of the file 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 HyperFileSQL data file according to a condition.
The browse can be performed:

- on the best key of the data file, automatically detected by the HyperFileSQL 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 the 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 browsing command should I choose?

All the types of browsing commands 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.

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 test the duplicate errors and the integrity errors.

▶ To open the example project:
1. Start WebDev 17 (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 project named "Management of HyperFileSQL errors". The project is loaded.
Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. Management of HyperFileSQL 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.

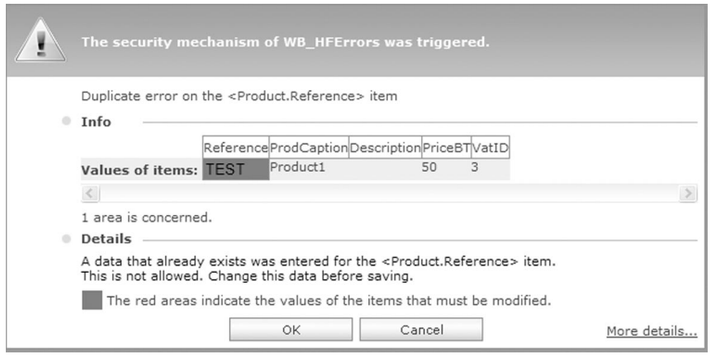
Let's now test the type of error that can be displayed in our simple example.

▶ Click the "GO" button to run the "WB_HFErrors" project.

▶ Click "Automatic".

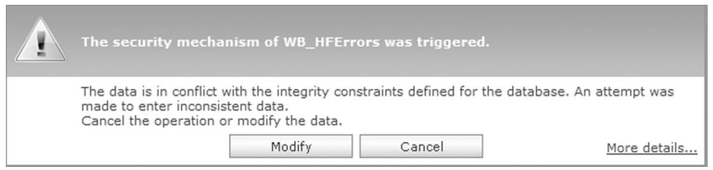
▶ We are now going to test the duplicate errors. To do so, enter the following information:
- Reference: TEST
- Caption: Product1
- Description: Product1
- Price: 50
- VAT rate: 5.5%

- Click the "OK" button. 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 test the integrity errors. To do so, enter the following information:
  - Reference: TEST2
  - Caption: Product2
  - Description: Product2
  - Price: 50
  - VAT rate: No VAT.
- Click the "OK" button. 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.

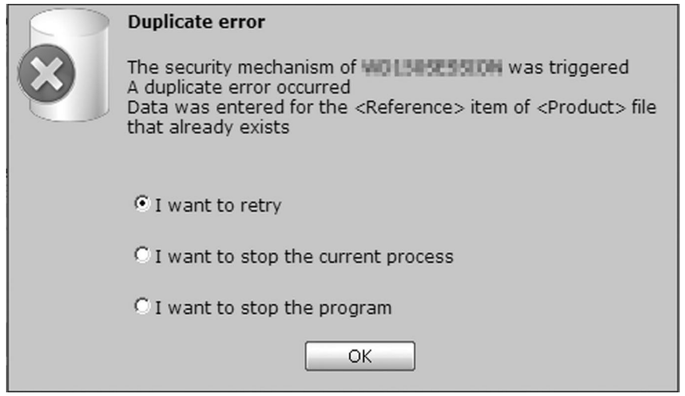
- Click "Cancel".

Managing the errors in advanced programmed mode

The management of errors in advanced programmed mode consists in customizing the error process 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.

- Click the "GO" button to run the "WB_HFErrors" project.
- Click "Advanced programming".
- We are now going to test the duplicate errors. To do so, enter the following information:
  - Reference: TEST
  - Caption: Product1
  - Description: Product1
  - Price: 50
  - VAT rate: 5.5%
- Click the "OK" button. 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 test the integrity errors. To do so, enter the following information:
  - Reference: TEST2
  - Caption: Product2
  - Description: Product2
  - U.P. BT: 50
  - VAT rate: No VAT.

- Click the "OK" button. 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 line:

```
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 indicate 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. THE LOG PROCESS

This lesson will teach you the following concepts ...

- What is the log process?
- Using the log process.



Estimated time: 15 min

What is the log process?

This lesson presents the log process for the HyperFileSQL databases. For the external databases (Oracle, ...), see the relevant documentation.

What is the log process used for?

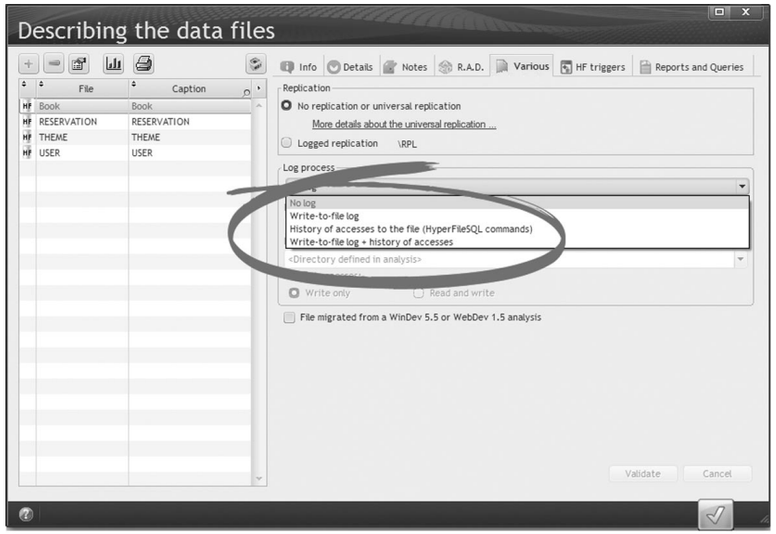
The log file is a specific file that will allow you to fulfill the following requirements:

- restore the database to its previous status further to a physical problem, errors in processes, ...
- 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 241).

The log file acts as a security mechanism for your database. It enables you to quickly restore your database to its last known state before the problem occurred, since the last backup, without having to re-enter anything.

How do I implement the log process?

WebDev enables you to easily implement the log process. For each data file, all you have to do is specify the type of log process to apply ("Various" tab in the description window of the data file).
For example, for the "Book" file:



All you have to do is choose the requested type of log process:

- Choose "Write-to-file log" if you want to quickly restore your database following a physical problem or a processing error.

- Choose "History of accesses to the file" if you want to keep track of the accesses to the database.



Caution !

The log process whose type is "History of accesses to the file (HyperFileSQL commands)" does not allow you to restore your database if a problem occurs. This option is used to perform checks on the accesses to "sensitive" data files or to perform statistical calculations.
This type of log process is not recommended.

- Choose "Write-to-file log + history of accesses" if you want to combine the two options.

How does the log process physically operate?

All the accesses to all the logged files found in an analysis are archived in a file named "Table of operations" (one per analysis).
This unique file guarantees that the operations are recorded sequentially.
All the modifications made to a data file are stored in the corresponding log file (one per logged file). For example, for a data file named "PRODUCT", the log file will be "PRODUCTJNL".

By default, these "log" files are found in the directory defined when describing the data files of the analysis:

- If nothing was specified, they will be found in the "JNL" sub-directory of the directory of data files.
- If a directory was specified, they will be found in this directory.

The name of the directories is stored in the file itself.

The directories of the log files can be modified once the log process is implemented:

- by WDLog, supplied with WebDev ("Information about a file").
- by programming. You have the ability to modify this location by using ***HChangeLogDir and HChangeRplDir***, ... (See the online help (keyword: "Log process") for more details).



Caution !

Modifying the directories of log files is a "critical" operation.
When managing your site, you must necessarily follow the operating mode below:

1. Forbid the accesses to the database (all the files).
2. Move the log files and/or the operation tables to the new directories.
3. Modify the name of the directories in ALL the relevant data files ("Various" in the description window of the files).
4. Re-allow the accesses to the database.

How do I successfully implement the log process?

Implementing log files for a database is closely linked to the management of backups for this same database (backups of data files).
Indeed, a log can only be "applied" to a given status of the database (the status that corresponds to the creation or to the re-creation of the log).
These operations can be performed without programming via WDLog supplied with WebDev.

We recommend that you follow the operating modes below:

- Implementing the log process:
  1. Saving the initial data files (if they exist).
  2. Implementing the log process.
- Saving an application with logged files:
  1. Saving the data files.
  2. Saving the log files if necessary.
  3. Reinitializing the log files.
- Restoring following a problem:
  1. Restoring the last backup.
  2. Running the current log up to the requested operation.
  3. Performing another backup and reinitializing the logs.

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



Notes

Log process and HyperFileSQL Client/Server
The log process is available in HyperFileSQL Client/Server. See the online help for more details.

LESSON 4.5. THE TRANSACTIONS

This lesson will teach you the following concepts ...

- What is a transaction?
- How should I use the transactions?



Estimated time: 10 min

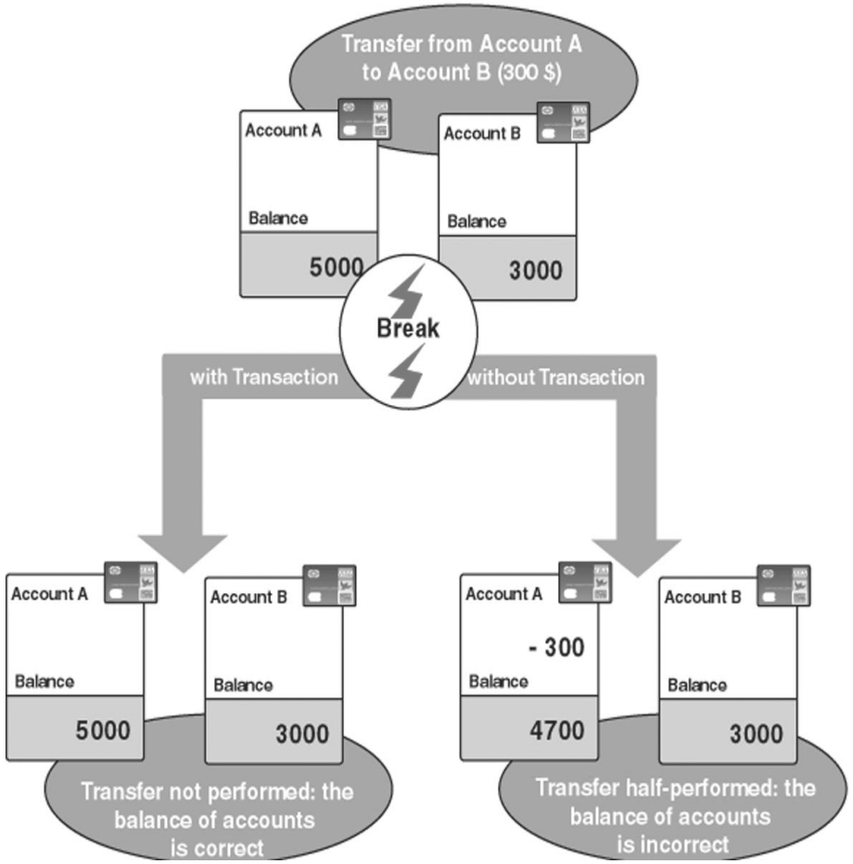
What is a transaction?

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.

Managing the transactions is the best way to ensure the integrity of a set of indissociable write operations performed on HyperFileSQL data files.
The transactions are also a reliable way for securing your processes on HyperFileSQL 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, and so on). 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, you don't end up with an account being debited without the other one being credited!



The transactions on HyperFileSQL

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 the transaction will be canceled.
The transactions are 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.

Using the transactions

Implementing the management of transactions

1. If your data files are password protected, open all the data files used during the transaction (***HOpen***) before the beginning of the transaction or specify the passwords with ***HPass***.
If your data files are not password protected, the data files used after the call to ***HTransactionStart*** will automatically belong to the transaction.
2. Start the transaction with ***HTransactionStart***. This function can also be used to define the name of the transaction log.
3. Perform your operations. All the write operations performed on the files in transaction are automatically saved in the transaction file.



Caution ! The processes performed are slower (because each operation is recorded in a specific file).

4. Cancel (if necessary) the operations performed during the transaction (***HTransactionCancel***).
5. Specify the end of transaction with ***HTransactionEnd***: the transaction is validated.

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

Handling the records during a transaction: the rules to follow

The records modified during the transaction can be read before or after the beginning of the transaction: they will always be taken into account in the log of transactions.

Managing the transactions does not exclude managing the locks for the records in transaction.
Indeed, the records handled during the transaction are automatically locked in write mode.

Therefore, the transaction must be as short as possible to avoid locking the users. Don't forget to limit the number of statements for browse, modification ... on the data file between the calls to ***HTransactionStart*** and ***HTransactionEnd***.

What should I do if a transaction is interrupted?

If a breakdown (power outage, reboot, ...) occurs during a transaction, the data files may become corrupted: the transaction was neither validated nor canceled. The transaction file is still found on the computer.
In this case, the consistency of the database must be restored. This operation can be performed automatically:

- during the next call to ***HTransactionCancel*** or ***HTransactionStart***.
- by WDTRANS, a tool for managing the transactions supplied with WebDev.

To find out whether the integrity of the database must be restored, check the result of ***HTransactionInterrupted*** in the initialization code of the project.

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



Notes

Transactions and HyperFileSQL Client/Server
Transactions are available in HyperFileSQL Client/Server. See the online help for more details.

LESSON 4.6. 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 is to guarantee the confidentiality of the data found in the HyperFileSQL files.

This confidentiality is guaranteed by the encryption of the data files.

The encryption of the 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.

Different types of encryption are available:

- 128-bit encryption (high performance)
- RC5 encryption (128 bits, 12 rounds, less performance but more secured)
- RC5 encryption (128 bits, 16 rounds, less performance but more secured)

The specified key (or "password") will be valid for the data file itself, 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 ...

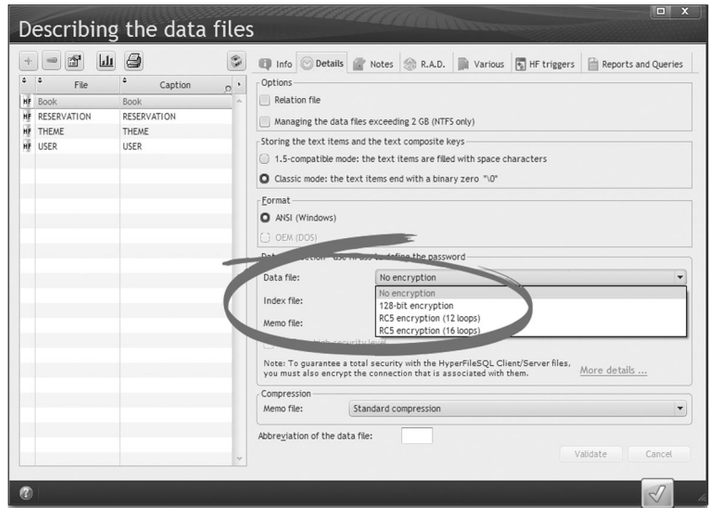
The password must be managed by the developer, in other words ... you! WebDev will encode or decode according to the specified password.

How do I encrypt the data files?

Implementing the encryption of the data files

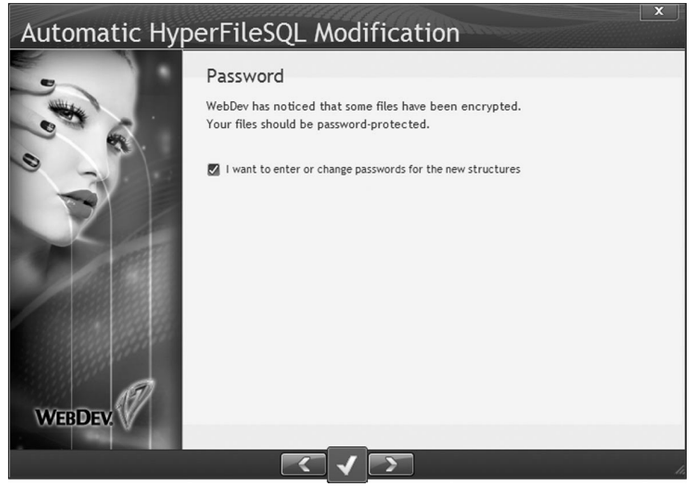
▶ To use the encryption on a data file:

1. Load the analysis of your project with the data model editor ("Project .. Load the analysis").
2. Select the data file that must be encrypted.
3. In the description of the data file ("Structure of files .. Description of data files", "Details" tab), choose the type of encryption for your data file, your memo or your index.

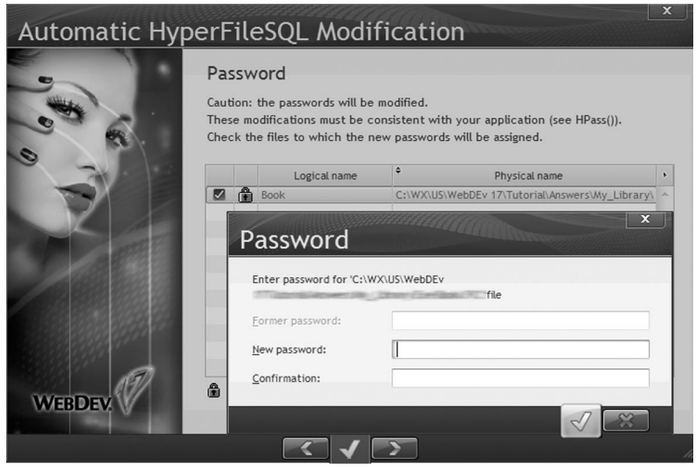


You also have the ability to enable a higher security. In this mode, the encryption password must be entered whenever an automatic data modification is performed (new setup, update, ...).

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:

- indicate the password with the functions for opening and creating the data files (***HCreation***, ***HCreationIfNotFound***, ***HOpen***).
  Example for using the ***HCreationIfNotFound*** function:

```
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 HErrorPassword() THEN
   Trace("HyperFileSQL error: " + HErrorInfo())
   RETURN
END
```

See the online help (keyword: "HPass", "FicCryptMethod", "MmoCryptMethod", "NdxCryptMethod") for more details.

LESSON 4.7. THE REPLICATION

This lesson will teach you the following concepts ...

- What is the data replication used for?
- Implementing the data replication.



Estimated time: 20 min

What is the data replication used for?

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 enables 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 HyperFileSQL 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 HyperFileSQL database with an Oracle database for example).

We shall only present the universal replication.

Implementing the data replication

The purpose of the universal replication is to keep several databases synchronized. These databases can have different types. For example, a replication can be performed between a HyperFileSQL database and an Oracle database, or between two HyperFileSQL databases.
The universal replication uses a centralized model: all the databases are synchronized with a master database. Then, the master database carries over the modifications to the other databases.

The universal replication uses several types of files:

- **.RPM file**: file used to describe a master database as well as the databases that subscribe to it.
- **.RPL file**: file used to describe a subscriber database. A RPL file is created for each subscriber database. This file is found on the subscriber computer.
- **.RPA file**: log file containing the replication information. This file is exchanged between the master database and the subscriber database.
- **.SYN file**: file containing information about the situation on the remote database. This file is used to optimize the size of the synchronization files. This file is found on the master computer and on each subscriber computer.

Note: To implement the universal replication on databases other than HyperFileSQL, a ***DateTime*** item must necessarily be created in each data file. This item will be updated by the application when a record is modified or added. If the databases use different time zones, we recommend that you use a universal format (GMT date and time for example).

Implementation

1. Activation

To enable the universal replication, all you have to do is use ***HSetReplication*** associated with the ***rplReplicationUniversal*** constant.
This function is used to disable the standard replication mode (if it was enabled) and to enable the universal replication.

2. Declaring the master database

This operation must be performed once only on the master database.
To declare the master database, all you have to do is use ***HCreateMasterReplica***.
Note: If data can be stored in the current directory, the following code line can be used:

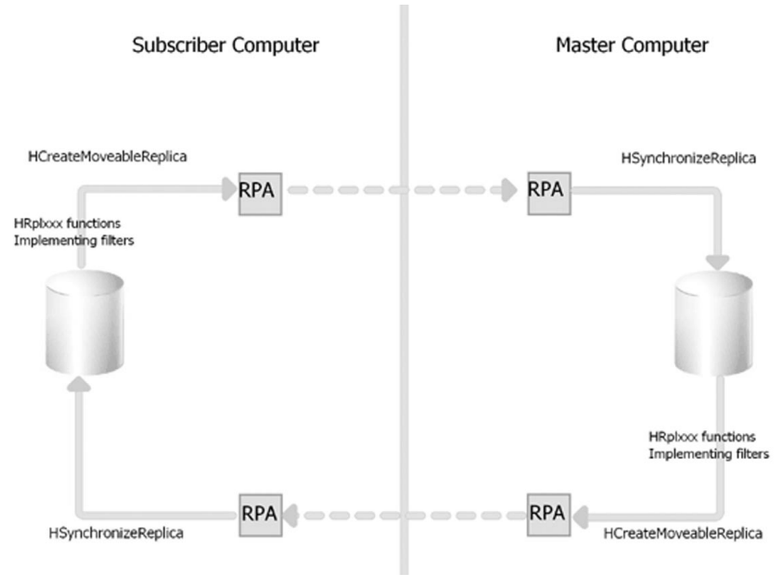
```
HCreateMasterReplica("")
```

This code line creates the MasterReplica.RPM file on disk. Then, all you have to do is write the subscribers into this file.

3. Declaring the subscriber databases

This operation must be performed once only for each subscriber database. This operation must be performed on the master database.
To declare a new subscriber, use ***HCreateSubscriberReplica***. This function creates a subscriber (RPL file) with the specified name. This functions also returns a subscriber number.
Note: ***HCreateSubscriberReplica*** uses specific parameters for the universal replication. See the help about the function for more details.

4. Replication



HCreateMoveableReplica creates a log file (.RPA file).

This file is transmitted and run by ***HSynchronizeReplica***.

Caution: By default, the master has priority when the synchronisation is performed (*HSynchronizeReplica*): if a replication is performed from the subscriber to the master, the data found in the master database will not be updated. We recommend that you use another constant (rplMostRecentFirst for example).

Two specific functions can also be used:

HRplDeclareLink	Used to signal to the replication engine that a link was found between two data files. The engine will follow the link to get the list of records that must be replicated in the second file.
HRplFilterProcedure	Used to specify a specific filter procedure when a given file is replicated.

LESSON 4.8. THE TRIGGERS

This lesson will teach you the following concepts ...

- What is a trigger used for?
- Using a trigger.



Estimated time: 10 min

What is a trigger used for?

Definition of a trigger

A trigger is used to automatically trigger an action when an addition, a modification or a deletion is performed in a HyperFileSQL data file.

During an operation (addition, modification or deletion) on a HyperFileSQL 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.

Implementation

The triggers are defined by programming. The following WLanguage functions can be used:

- ***HDescribeTrigger***: describes the trigger. This function is mainly used in the initialization code of the project. The trigger is used to call a procedure. This procedure is associated with the trigger when using ***HDescribeTrigger***.
- ***HActivateTrigger***: Enables a trigger described by ***HDescribeTrigger.*** The trigger is enabled as soon as it is created.
- ***HDeactivateTrigger***: Disables a trigger described by ***HDescribeTrigger.***



Caution !

The function named ***HDeactivateTrigger*** does not destroy the trigger. The trigger is disabled but it still exists.

- ***HDeleteTrigger***: Destroys a trigger described by ***HDescribeTrigger.***

Step 1: describing the trigger (in the initialization code of the project)

Example:

```
- - initialization code of the project - -
HDescribeTrigger("CUSTOMER,SUPPLIER", "HDELETE", ...
    "ProcConfirmDeletion", hTriggerBefore)
```

In this example, the trigger will call the "ProcConfirmDeletion" procedure when a record is deleted from one of the files ("CUSTOMER" and/or "SUPPLIER"). This deletion may occur anywhere in the project.



Caution !

The "ProcConfirmDeletion" procedure will be called BEFORE (**hTriggerBefore**) the deletion is performed.

Step 2: creating a global procedure linked to the trigger

Example:

```
PROCEDURE PROCCONFIRMDELETION ()
// Asks the user for confirmation
IF YesNo("Do you confirm the deletion? ") = No THEN
    // Initialize H.ToDo to cancel the operation
    H.ToDo = "A"
END
```

We have presented the operating mode of triggers. See the online help (keyword: "HDescribeTrigger") for more details.

LESSON 4.9. EXTERNAL DATABASES

This lesson will teach you the following concepts ...

- Configuring the connection to a database in the analysis
- Configuring the connection to a database by programming

Estimated time: 15 min

Overview

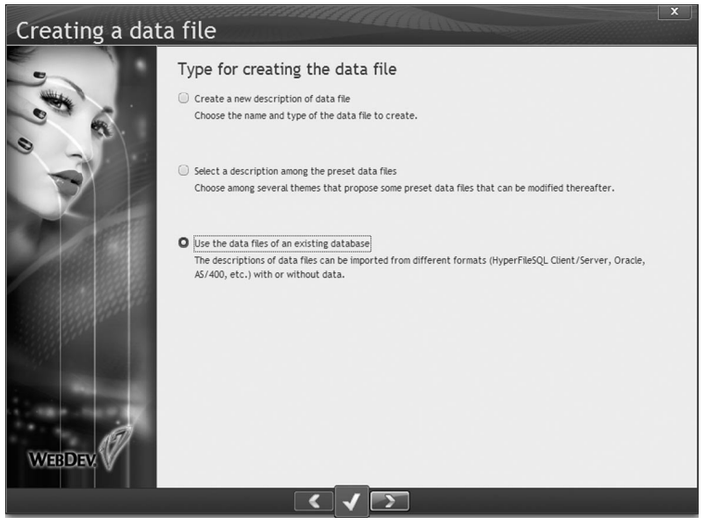
Any system for managing databases other than HyperFileSQL will be called an "External database". Any program created in WebDev can easily access a database other than HyperFileSQL.
To do so, all you have to do is describe the database in an analysis (as for HyperFileSQL).

The modes for accessing the databases are:
- native access
- access via ODBC
- access via OLE DB
- ODBC access via OLE DB

The programming will be performed by the functions of WLanguage (HRead*) and/or by using the SQL language. To display the comparison table, select "? .. Help for using the databases" from the WebDev menu.

Connecting to a database in the data model editor

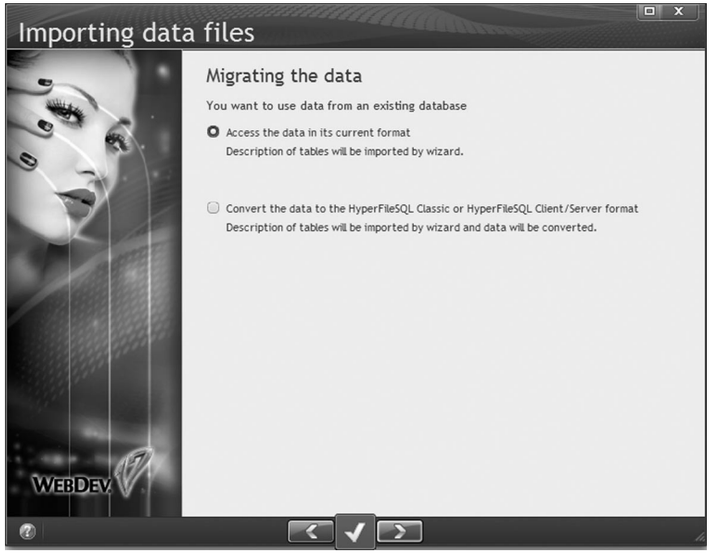
When creating a new data file ("Insert .. Data file" in the data model editor), select "Use the data files from an existing database":



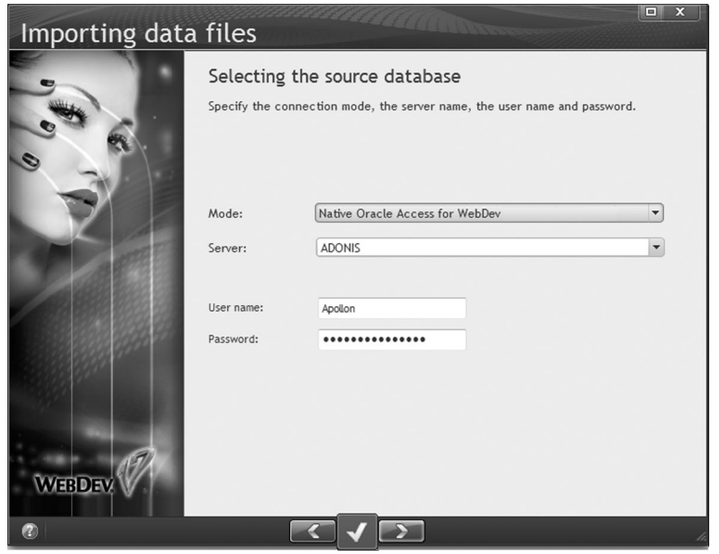
Then, select the type of the database:



Specify whether the data will be converted to HyperFileSQL Classic or whether it will remain in its current format:



The parameters for connecting to this external database must be described (name of server or service, database, login and password, type of access (Native, OLE DB, ODBC)).



This information is used to define a connection in the analysis. This connection will be automatically used when generating RAD (full application RAD or window RAD). All the data files are linked to this connection.

The use of a connection specified in the analysis is very convenient for the development of the application.

However, in most cases, the parameters used during the development correspond to the parameters of an administrator while the client application must use a more secure connection mode (with less rights).

Therefore, the connection must be redefined. Two solutions are available:

- Solution 1: Redefining the specified connection in the analysis. This solution forces you to keep "hard-coded" and non-configurable data in the application.
- Solution 2: Redefining the specified connection by programming. This solution forces you to use ***HOpenConnection*** and ***HChangeConnection***.

Connecting to a database by programming

Connecting to a database by programming is performed by two WLanguage functions:

- ***HOpenConnection***, used to describe the different parameters of the connection and to open it.
- ***HChangeConnection***, used to associate a connection with the different data files of the analysis.

Let's study the operations required to implement the connection by programming.

Creating the connection: HOpenConnection

The easiest method for using **HOpenConnection** is to use the wizard of this function: answer the different questions and the corresponding code is automatically generated in your process.

To use the wizard of **HOpenConnection**:

1. In the code editor, type the name of the function followed by an opening bracket:

```
HOpenConnection(
```

2. Click the "Wizard for HOpenConnection" option that is displayed. The wizard is automatically started.

The wizard asks for the different parameters of the connection:

- Name of the connection
- Provider
- Encryption
- User and password
- Server and database



Caution !

The user and the password specified in the wizard will be displayed in the generated code and they will be "hard-coded".

You may have to create a login window in order to identify the user and to use variables for the user name and password. These operations can be performed afterwards.

3. Follow the wizard and validate. The corresponding code is automatically included in the current process.

The function named **HOpenConnection** was used to define a connection but this connection is not yet associated with the data files.



Caution !

Don't forget to test the result of **HOpenConnection**. This function returns False if the connection cannot be opened.

Associating the connection with the data files: HChangeConnection

To associate a connection with a data file, all you have to do is use **HChangeConnection**. The syntax of this function is quite simple:

- the first parameter corresponds to the name of the data file ("*" for all the analysis files)
- the second parameter corresponds to the name of the connection to use.

Example:

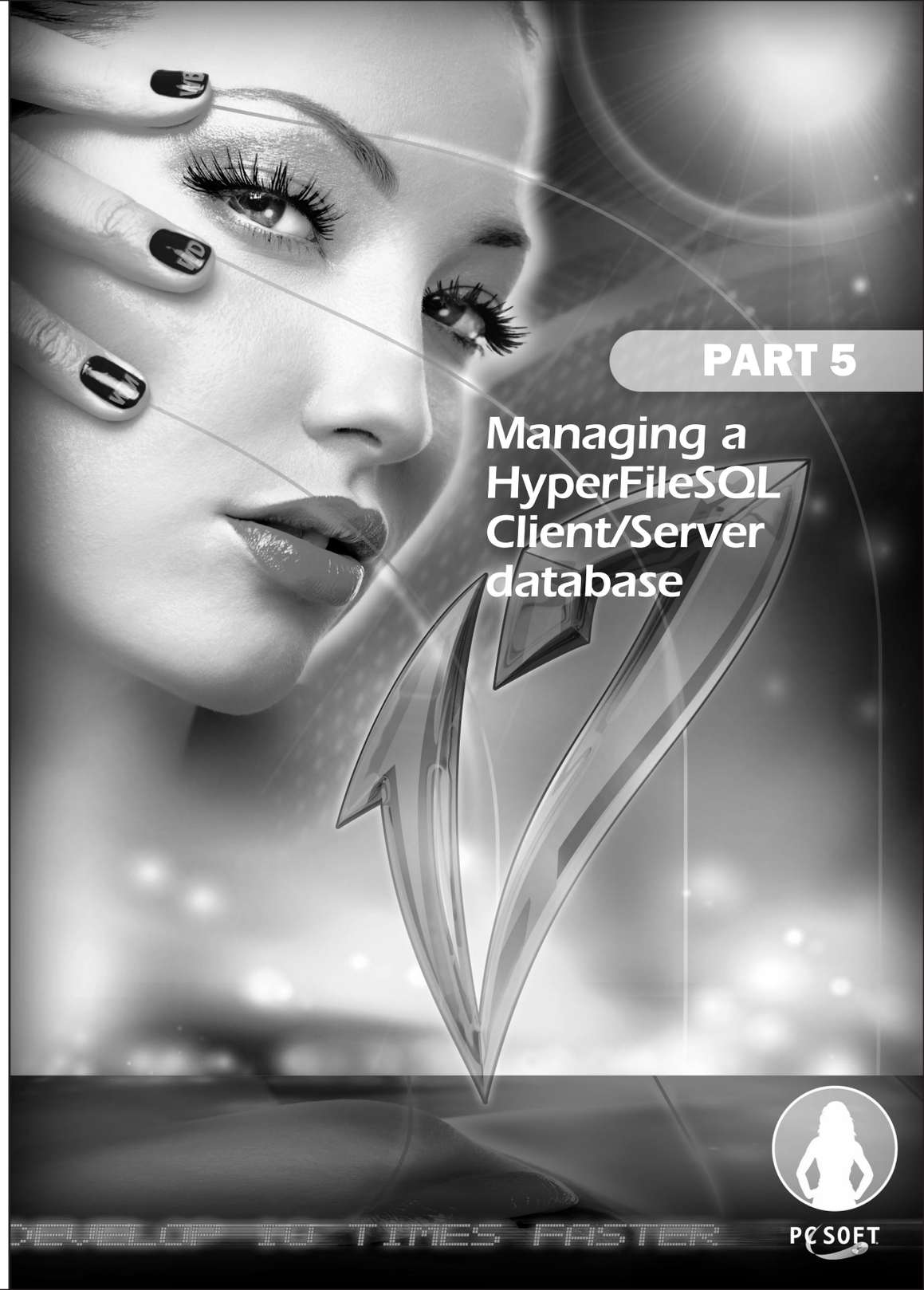
```
HChangeConnection("*","MyNewConnection")
```



Notes

You can describe as many connections as necessary.

A data file can use a connection that differs from the connection used by the other files found in the analysis.



LESSON 5.1. INTRODUCTION

This lesson will teach you the following concepts ...

- Principle of Client/Server
- Why switch a site to HyperFileSQL Client/Server?

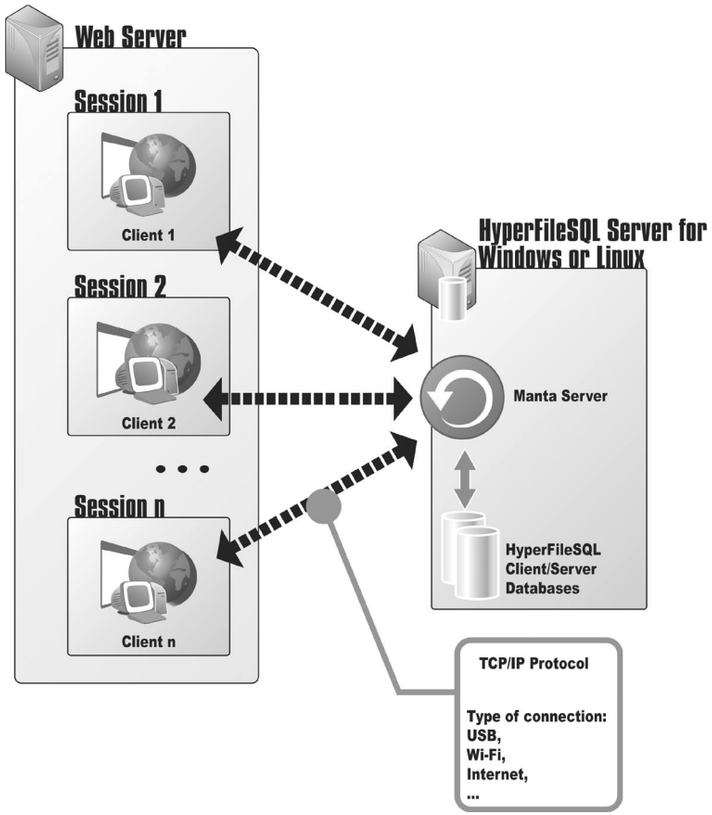


Estimated time: 5 min

Overview

WebDev enables you to create applications that access HyperFileSQL Client/Server databases.

A HyperFileSQL 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 enables you to:

- create a HyperFileSQL Client/Server site from scratch.
- modify an existing WebDev site into a HyperFileSQL Client/Server site.

Why switch a site to HyperFileSQL Client/Server?

The main benefits of a site in HyperFileSQL Client/Server compared to a site in HyperFileSQL Classic:

- The use of HyperFileSQL Client/Server is more secured (use of a login and password and definition of the 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 HYPERFILESQL CLIENT/SERVER SITE

This lesson will teach you the following concepts ...

- Installing a local HyperFileSQL server
- Creating a site in HyperFileSQL Client/Server mode
- Migrating a site in HyperFileSQL Client/Server mode
- Using the HyperFileSQL Control Center
- Features available in HyperFileSQL 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 HyperFileSQL Client/Server site.

Installing a local HyperFileSQL server

The first operation before developing a HyperFileSQL Client/Server site consists in installing a HyperFileSQL 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 HyperFileSQL server is available from the WebDev CD. If you do not own this CD, the setup of the HyperFileSQL server is also available from our Web site (www.windev.com).

To install the HyperFileSQL server locally:

1. Start the setup program of WebDev.
2. Choose "Install or update a HyperFileSQL 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. The HyperFileSQL Control Center can be installed if this one is not found or accessible from your computer.



Caution !

The HyperFileSQL Control Center is required to manage the HyperFileSQL 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 HyperFileSQL Client/Server mode

The method for creating a HyperFileSQL Client/Server site is identical to the method for creating a 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 "HyperFileSQL Client/Server" databases.
3. Specify the characteristics of the connection to the HyperFileSQL Client/Server server that will be used.
4. When a data file is created in the analysis, indicate that this file is in Client/Server mode and specify the connection used.

Migrating a site from HyperFileSQL Classic to HyperFileSQL Client/Server

Overview

Migrating a HyperFileSQL Classic site to the 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 HyperFileSQL 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 (Library)".
Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. Site with data (Library)").

3. Display the analysis of the project ("Project .. Load the analysis"). The data model editor is displayed.

4. In the data model editor, select "Analysis .. Associate a connection with data files". No connection being defined in our analysis, a wizard is opened, allowing you to create a connection.

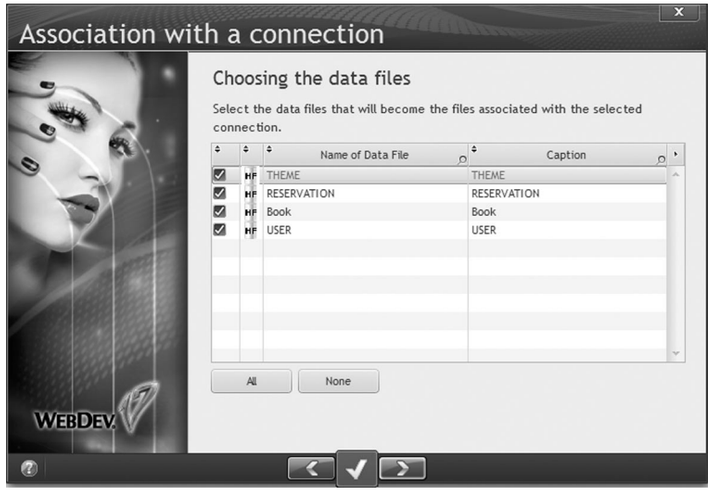
5. Select the type of connection to create: "HyperFileSQL Client/Server". Go to the next screen:



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 (leave this information empty to use the administrator)
- the name of the database (My_Library in our example)
- the name of the connection (keep the proposed name)

7. Go to the next screen. The connection to the database is automatically created. Then, the wizard proposes to associate the different data files found in the analysis with the created connection.



Go to the next screen.

8. The wizard proposes to create the data files on the server. Validate ("Copy now").

9. 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.

10. The data files of the analysis are automatically transformed into HyperFileSQL Client/Server data files and associated with the selected connection.

11. Generate the analysis.



Caution !

Check the code of your project: in HyperFileSQL Client/Server, HSubstDir, HChangeDir, ... are useless.



Notes

According to the parameters specified when creating the connection, the connection defined in the analysis can be modified by ***HOpenConnection*** and ***HChangeConnection***. This operation was presented in “External databases”, page 248.



Notes

The function named **HOpenConnection** can be used to go back to the HyperFileSQL Classic mode: all you have to do is specify the path of the directory containing the HyperFileSQL Classic data files.

12. The development project was migrated. You may also have to migrate the deployed application (if the deployed application uses HyperFileSQL Classic data files for example). This operation is configured when creating the setup program of the application.

Features available in HyperFileSQL Client/Server mode

HyperFileSQL 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

We shall not describe these features here (some of them have been presented in this tutorial in HyperFileSQL 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 HyperFileSQL Control Center**
- **Creating a user in the HyperFileSQL Control Center**
- **Saving the database**



Estimated time: 20 min

Overview

Now that we know how to create and/or migrate a site in HyperFileSQL Client/Server, let's see how it can be managed.

Indeed, a Client/Server database requires:

- a specific configuration of the computers (installing a HyperFileSQL server, ...)
- a management performed via the HyperFileSQL Control Center.

Configuring the computers

To use a HyperFileSQL Client/Server database, a HyperFileSQL server must be installed on the server. Several HyperFileSQL 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 HyperFileSQL server that includes a test database and a production HyperFileSQL server that uses a different port can be installed **on the same computer**.

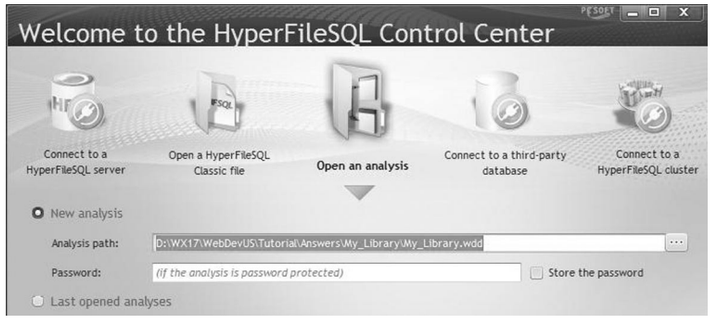
The HyperFileSQL Control Center

The HyperFileSQL Control Center is used to perform all the operations for managing the HyperFileSQL Client/Server servers and databases.

We are going to present the most important features.

To start the HyperFileSQL Control Center and to access the data:

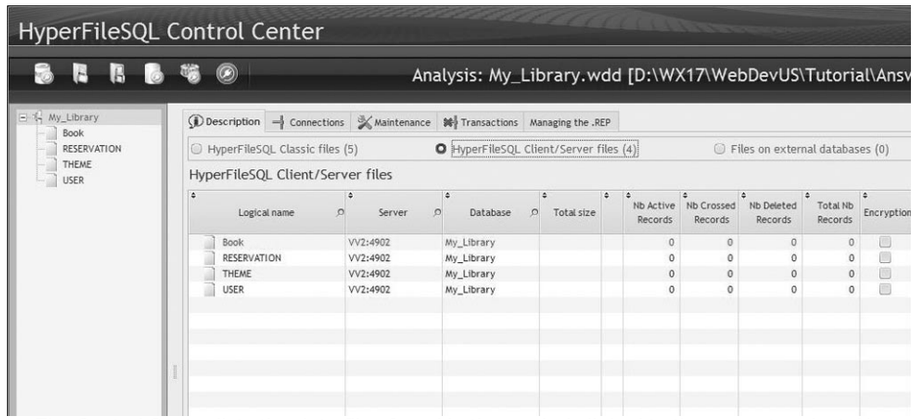
1. In the editor, select "Tools .. HyperFileSQL Control Center". The home window is displayed in the HyperFileSQL Control Center. The analysis of the current project is automatically selected.



Validate the screen. The HyperFileSQL Control Center is displayed.

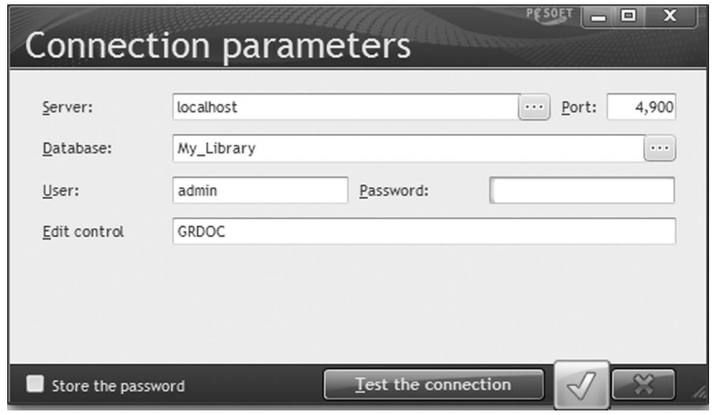
2. Click "HyperFileSQL Client/Server file". The list of data files in HyperFileSQL Client/Server format is displayed

.



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 HyperFileSQL Control Center displays a connection window used to establish the connection to the HyperFileSQL 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 (number of records, ...).

The entire HyperFileSQL Client/Server database can be managed from the HyperFileSQL Control Center.

Creating a user account in the HyperFileSQL Control Center

A single user account is created when installing a HyperFileSQL server and when creating a database: the administrator account ("Admin" login without password).

Using a user account enables 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 HyperFileSQL Control Center are granted for the database (and not for the application).

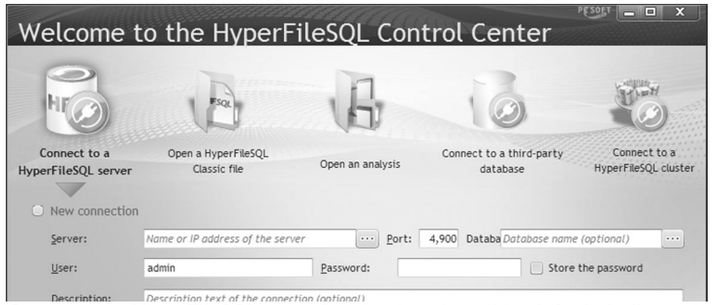
Don't confuse the management of rights for the Client/Server databases with the user groupware.

Some users may not have the rights to write into some 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 HyperFileSQL Control Center, click  and validate the closing of connections if necessary.

2. The home window is displayed in the HyperFileSQL Control Center.



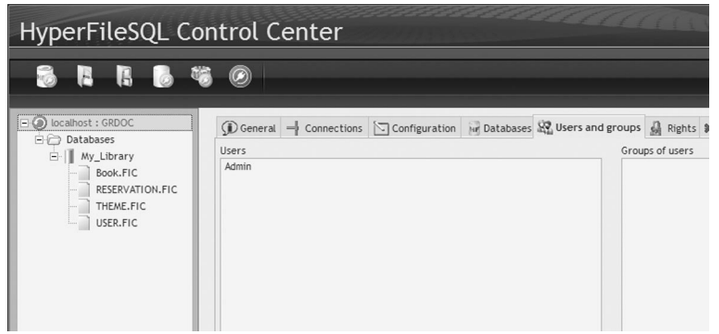
3. The "Connect to a HyperFileSQL server" option is selected by default. Indicate the characteristics of the server installed in the previous lesson.

4. The characteristics of the HyperFileSQL server are displayed:

- in the panel on the left, the computer, the name of the HyperFileSQL 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 HyperFileSQL server.

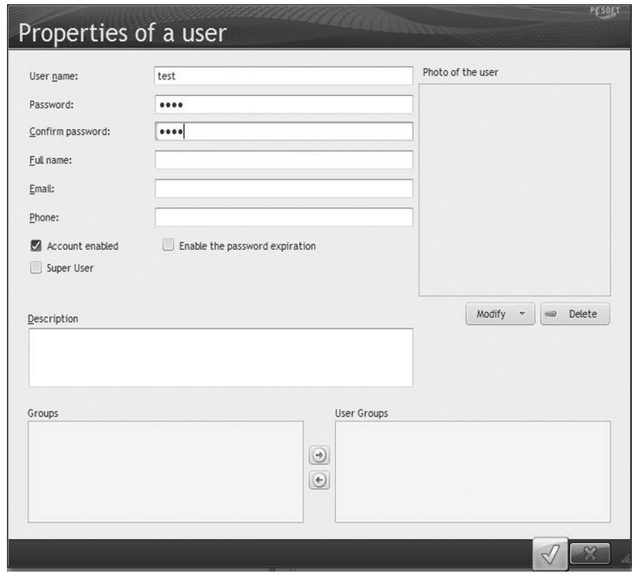
5. In the right section of the screen, select the "Users and Groups" tab. This tab is used to manage the users of the server.

Only the "Admin" user exists at this time.



6. To create a new user, click "New". 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 few days only.

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 rigths to connect to the database:

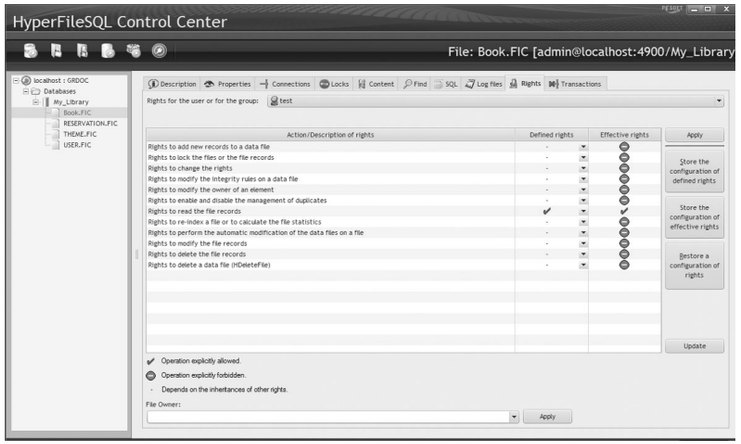
1. In the HyperFileSQL Control Center, select the "My_Library" database.
2. Click the "Rights" tab.
3. Select the "Test" user in the list of users.
4. In the list of rights, select "Rights to connect to the server (encrypted and unencrypted connection)". Select the green checkmark in the "Defined rights" column.
5. Click the "Apply" button. The rights become effective.

To grant the rights to read the Book file:

1. In the HyperFileSQL Control Center, select the My_Library database then 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 HyperFileSQL 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 HyperFileSQL Control Center.

Conclusion

The HyperFileSQL Control Center is a tool for managing the databases, allowing you to:

- stop or restart a server if a problem occurs,
- manage the users and their rights,
- reindex the data files if necessary,
- perform backups of the database.

The HyperFileSQL Control Center is a redistributable tool that can be installed on the computers of the users who work with HyperFileSQL Client/Server databases. The HyperFileSQL Control Center must be used by the administrator of the database.

Notes

Installing a site that uses a HyperFileSQL Client/Server HyperFileSQL requires:

- The setup of a HyperFileSQL 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
Queries
DEVELOP 10 TIMES FASTER
PC SOFT

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 enables you to create (without programming) the following types of interrogations:

- 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 (HyperFileSQL, 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 the current project if necessary. The home window is displayed.
2. In the home window, click "Tutorial" and select the project named "Using queries (Exercise)".
Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. Using queries (Exercise)".

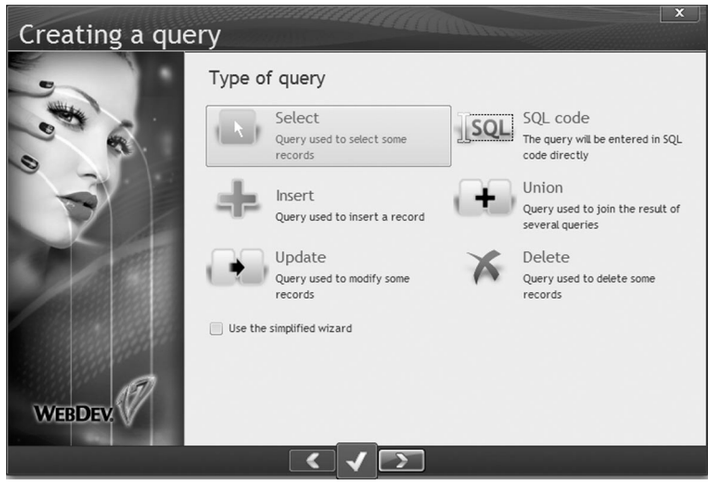
Your first query

We are now going to list the classified adverts for 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".

▶ Start the query editor:
1. In the menu, select "File .. New .. Query".
2. The wizard for query creation starts.

You can use the wizard for query creation or you can enter the SQL code of the query ("SQL code").
The wizard enables you to create the following types of queries:

- "Select": interrogation.
- "Insert": addition into a data file.
- "SQL code": customized query in SQL code.
- "Update": modification in a data file.
- "Union": combination of queries.
- "Delete": deletion from a data file.



Caution !
The direct input of SQL code is intended for 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, ... You will have the ability to improve your knowledge by using all types of queries.

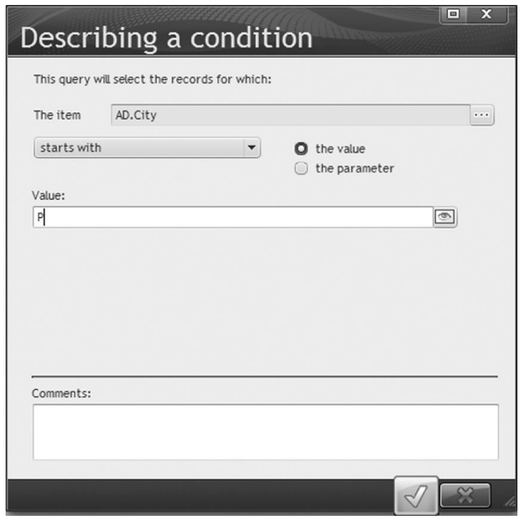
▶ To list the classified ads for a city starting with the letter "P", we must create a Select query. Go to the next plane.

▶ Select the items that will be displayed 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.

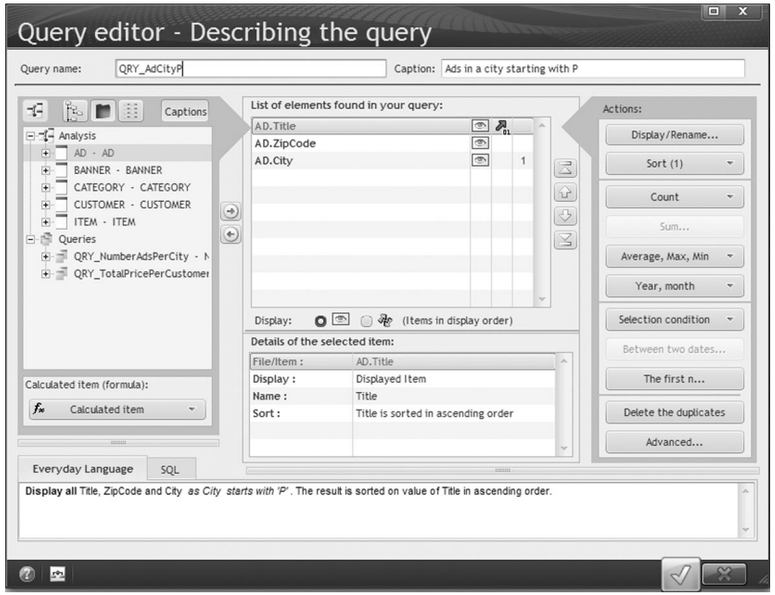


- Validate.
- We are now going to specify the sort criterion. The result will be sorted by "Title".
- Select the "Title" item (in the middle). In the "Actions", click the "Sort" button and select "Sort on the selected item". The sort will be performed in ascending order:

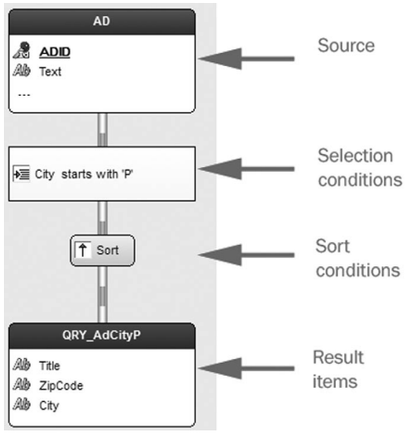


- Validate. A red arrow is displayed on the right of the "Title" item. This arrow indicates that this item will be sorted in ascending order.

- The query is created. It must now be named and saved.
  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:

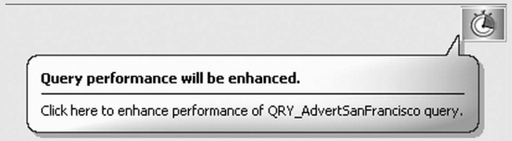


- Click  to save the query.



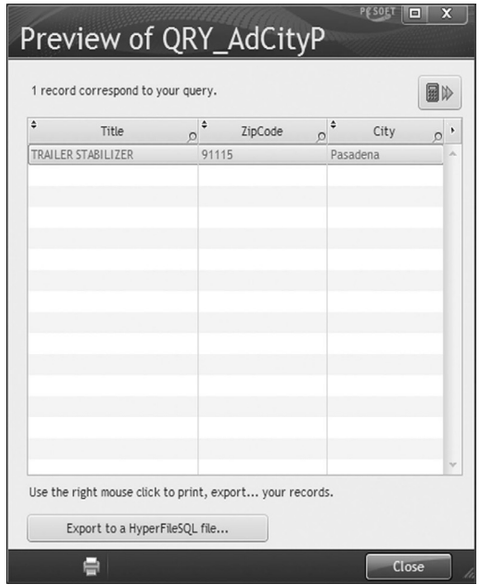
Notes

Once the query is created, the AAD (Aided Application Development) may be triggered to specify that the query can be optimized.



We shall not discuss this feature in the tutorial. See the online help (keyword: AAD) for more details.

Let's now run the test of the query: click  . The result is displayed in a window:



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 (e*Xtensible Markup Language*).
- a Word file, 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 will now 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.

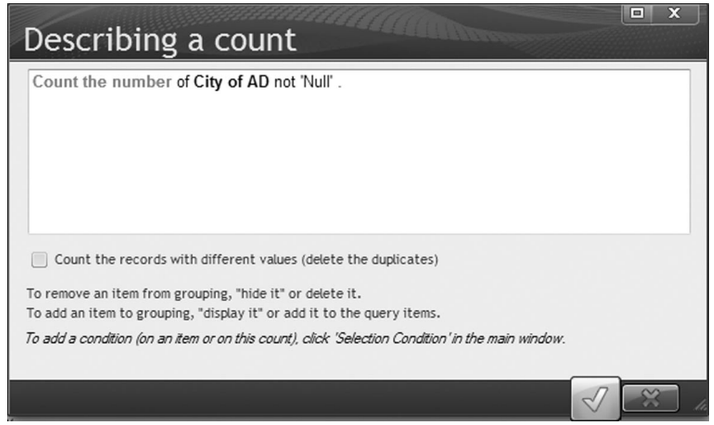
Notes

Reminder: If you don't want to perform the operations in this lesson, the corrected project is available via "? .. Tutorial .. Using queries (Answers)".

▶ Start the query editor:
1. In the menu, select "File .. New .. Query".
2. We are going to create a "Select" query ("Select"). Go to the next plane.

Select the items that must be displayed 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".

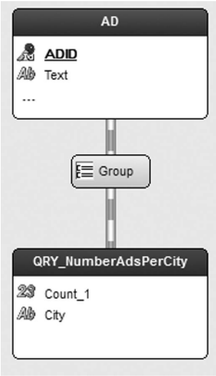


4. Validate.
The "Count(AD.City)" item is displayed in the result of the query.

▶ 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. It must now be named and saved.
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 to save the query. Let's now run the test of the query.
- Click .

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

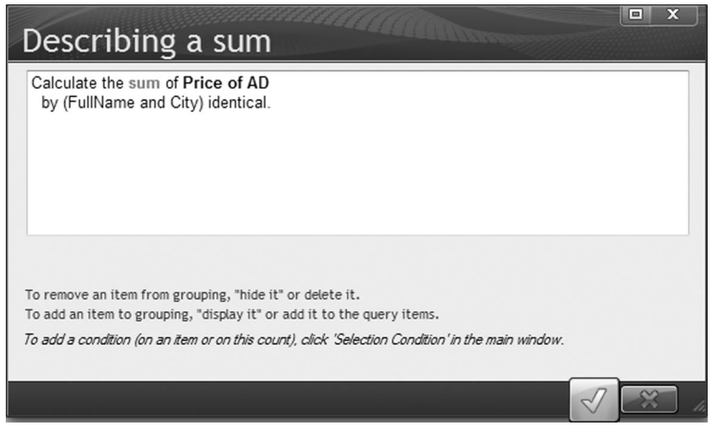
Creating a "Sum" query

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

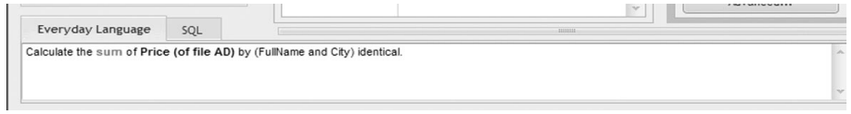
Reminder: If you don't want to perform the operations in this lesson, the corrected project is available via "? .. Tutorial .. Using queries (Answers)".

- Start the query editor:
  1. In the menu, select "File .. New .. Query".
  2. This query is a Select query. Select "Select". Go to the next screen.
- Select the items that must be displayed 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.

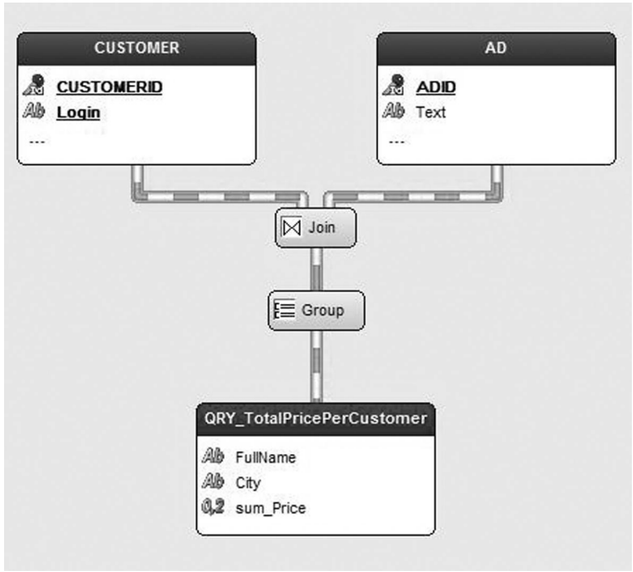
You will notice that the query editor of WebDev creates the queries in everyday language (and in SQL language).



This method for generating queries enables you to check whether your query performs the proper operations.

- The query is created. It must now be named and saved.
  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 to save the query. Let's now run the test of the query.
- Click .

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.

- Open the query named "QRY_TotalPricePerCustomer".
  If this query was not previously created, open the query named "QRY_TotalPricePerCustomer" found in the correction ("? .. Tutorial .. Using queries (Answers)").
- In the menu, select "Query .. 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 contains different keywords:

- the SELECT statement
- the FROM statement
- the WHERE statement
- the GROUP BY statement

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 the calculations will be checked out. 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 statement

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 file and the ORDERS file 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 statement

The GROUP BY statement is used to specify the group items or the group formulas when calculation is performed. For example, in the query that was previously created, 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 statement

The ORDER BY statement 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 directly based on the result of a query (see "The reports" page 299).

Using a query from a table or from a looper

When creating a table or a looper, 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 initializing the query"+...
                                  CR+HErrorInfo())
  RETURN
END
FOR EACH MyQuery
  // Process on the query record
  // read the next record
END
HCancelDeclaration(MyQuery) // Optional
```




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 over, 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 the WLanguage function named ***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

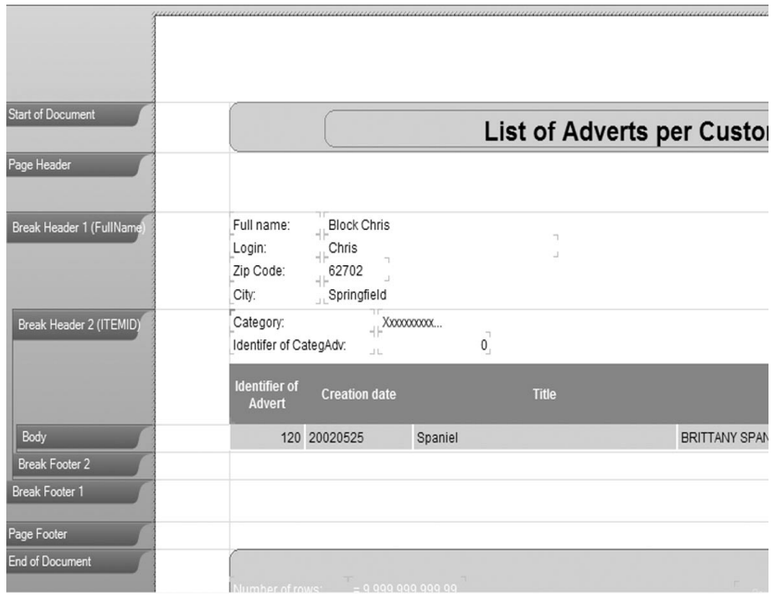
A **Report** is the name given to the representation of a print. A report is created in the WebDev editor. A document is created when the report is run (or edited).

WebDev proposes several types of reports:

- Blank
- Form
- Table
- Crosstab
- Organizer or schedule
- Mailing

- Label
- Form
- Multicolumn
- Composite

A **Block** is the name given to a section of the report. A report includes a set of blocks.



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 when a break is requested on a sort argument. The Break Footer block necessarily exists if this block exists.
  This block is run whenever the value of the sort argument changes, 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 when a break is requested on a sort argument. The Break Header block necessarily exists if this block exists.
  This block is run whenever the value of the sort argument changes, 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 the report is created. 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.
- Close the current project if necessary. The home window is displayed.
- In the home window, click "Tutorial" and select the project named "Printing (Exercise)". The project is loaded.
  Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. Printing (Exercise)".



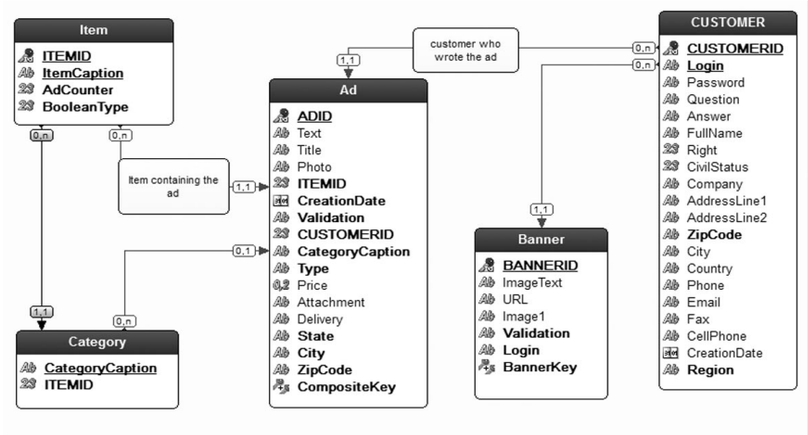
Notes

If you do not want to perform the operations in this lesson, the corrected project is available from the following menu option: "? .. Tutorial .. Printing (Answers)"

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:



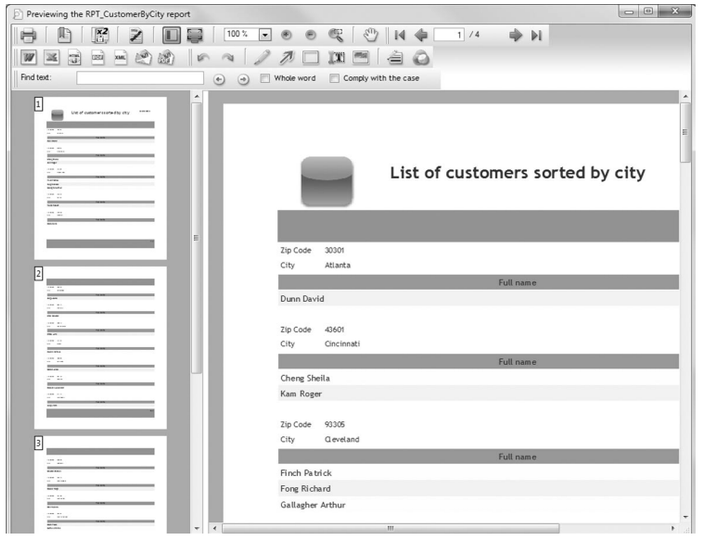
The reports to create

- To learn how to use the report editor, we will 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.

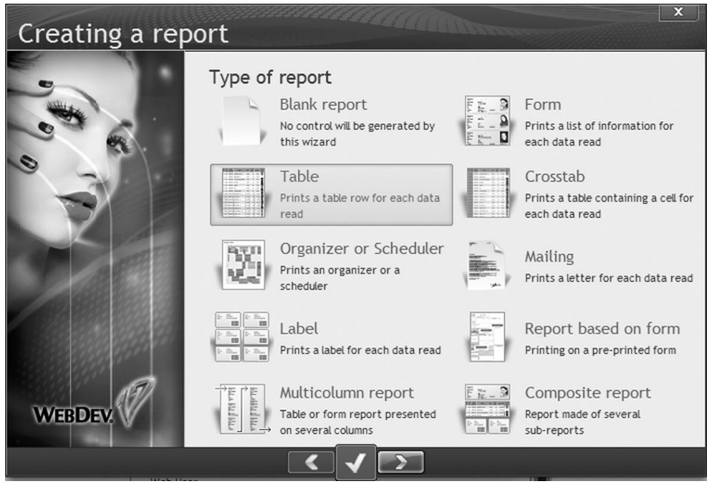
Creating your first report

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 will use a table report in order to clearly represent a list of data.



- To create a report, select "File .. New .. Report". The wizard for report creation starts and proposes several representations for the report:

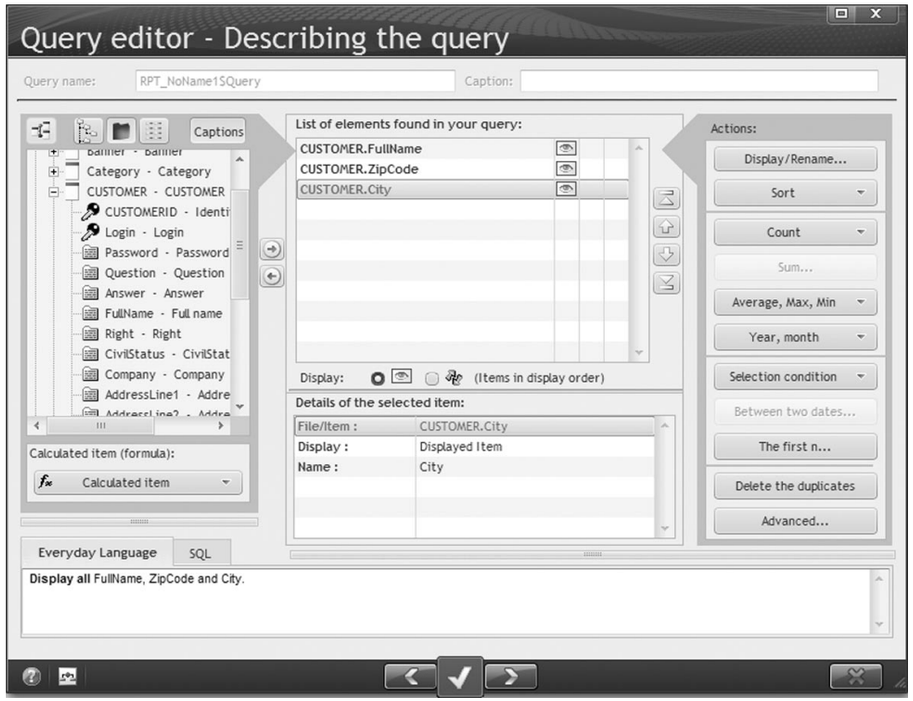


- Select "Table". Go to the next screen.

- 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 description is displayed. This screen enables you 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



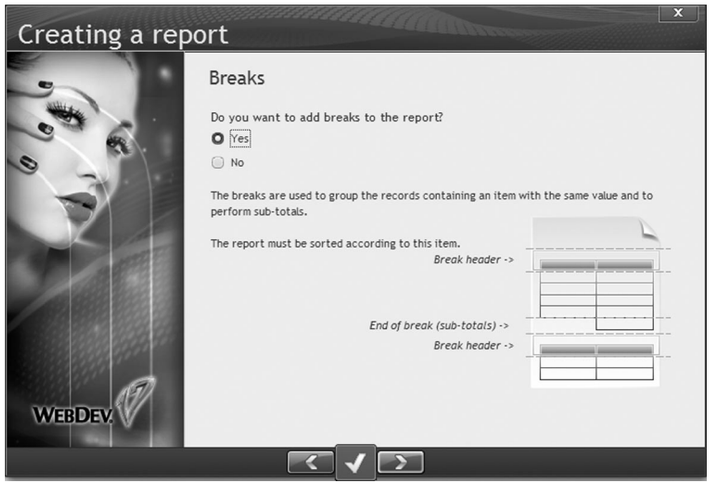
- To sort by city, a sort criterion must be chosen:
  1. Select the sort item: click "CUSTOMER.City" in the middle section of the screen.
  2. In the right section, click the "Sort" button and choose "Sort on the selected item".



3. Choose "Ascending" and validate. A red arrow is displayed to the right of the "City" item. This arrow indicates that this item will be sorted in ascending order.

- Go to the next screen to validate the creation of the query.

The next screen asks you to specify whether there is a break.



Notes

What is a break?

A **Break** is used to group records (or rows) according to one or more criteria. Caution, the records (or the rows) will be printed.

This break is **NECESSARILY** linked to a sort.

You will notice the presence of a break header and a break footer. The information common to the different rows will be grouped in the break header.

The totals and the sums will be displayed in the break footer.

In our example, the break is done on the city and is used to group all the customer from a same city.

- Answer "Yes" and go to the next screen.



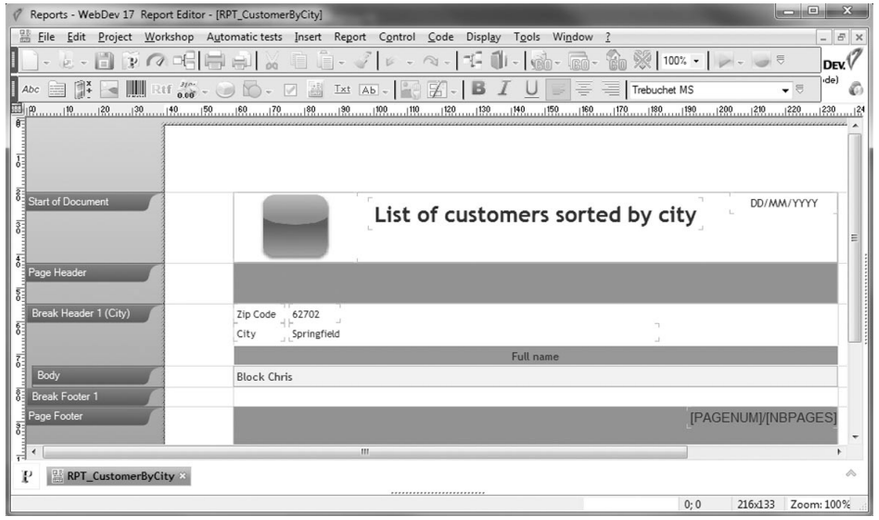
- The break is performed on the city. Go to the next screen.
- You will now specify the order in which the items will be printed and how they will be arranged in the blocks.
  In our example, the zip code and the city will be printed in the "Break header 1" block and the name in the Body block.
  In the column on the right, select "Break header 1" for "ZipCode".



The arrows to the right are used to modify the order of items. Go to the next screen.

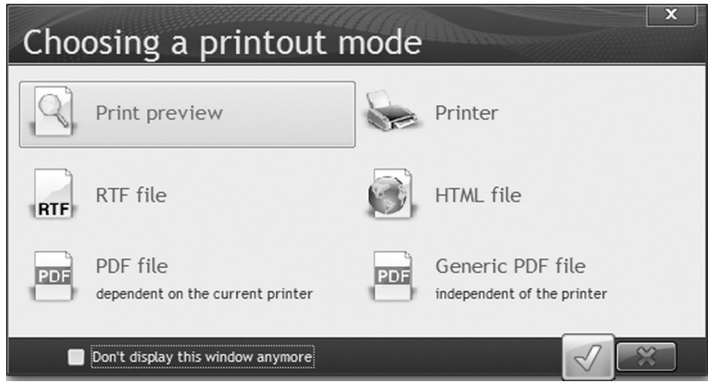
- This screen enables you to define the layout of the report. We will keep the default values as well as the "Portrait" orientation.
  When choosing the print margins, don't forget to take into account the physical margins of the printer. The physical margins of the printer are margins in which no print is allowed. Furthermore, the physical margins differ according to the type of printer. Go to the next screen.

- Select the skin template named "ActivPhone3". Go to the next screen.
- Let's now give a name and a caption to the report and save it.
  1. Enter the "RPT_CustomerByCity" name.
  2. Enter the title: "List of customers sorted by city".
  3. Validate. The report is displayed in edit mode. Resize the title if necessary so that it is entirely displayed.

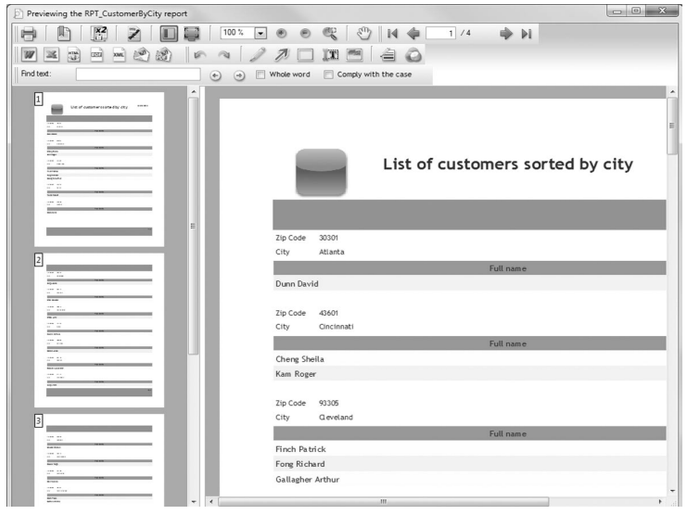


4. Save the report ( ). The report is completed. Let's now run the test of the report.

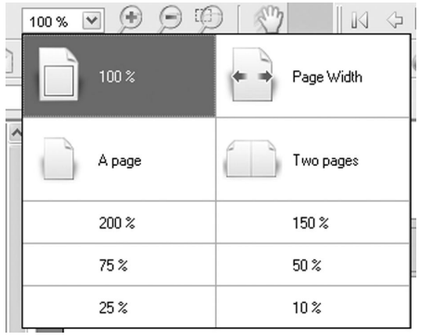
- To run this report, click the "GO" icon found in the icon bar. The print destination can be:



- 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

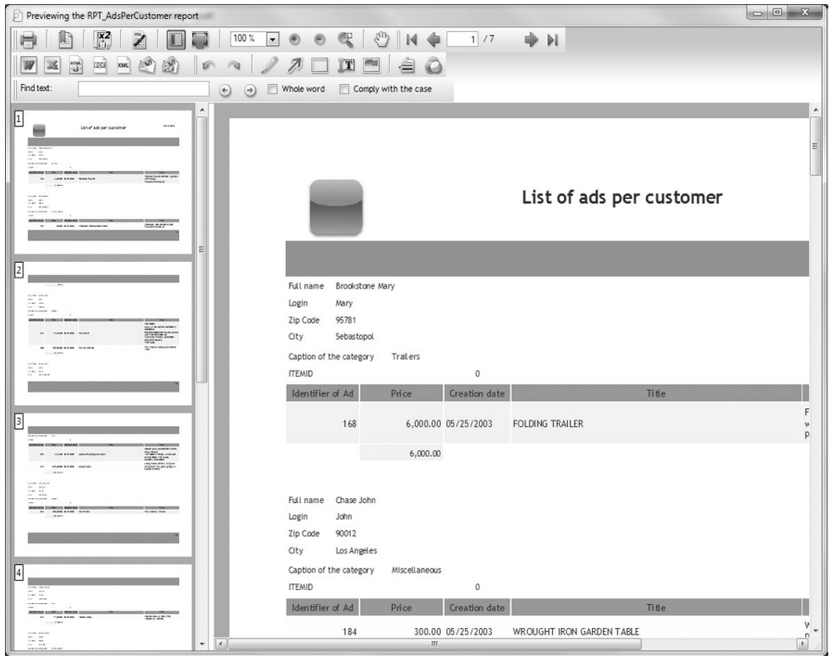
This lesson will teach you the following concepts ...

- 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 uses several data files as data source.

Reminder: A sample project containing the necessary data was created beforehand.

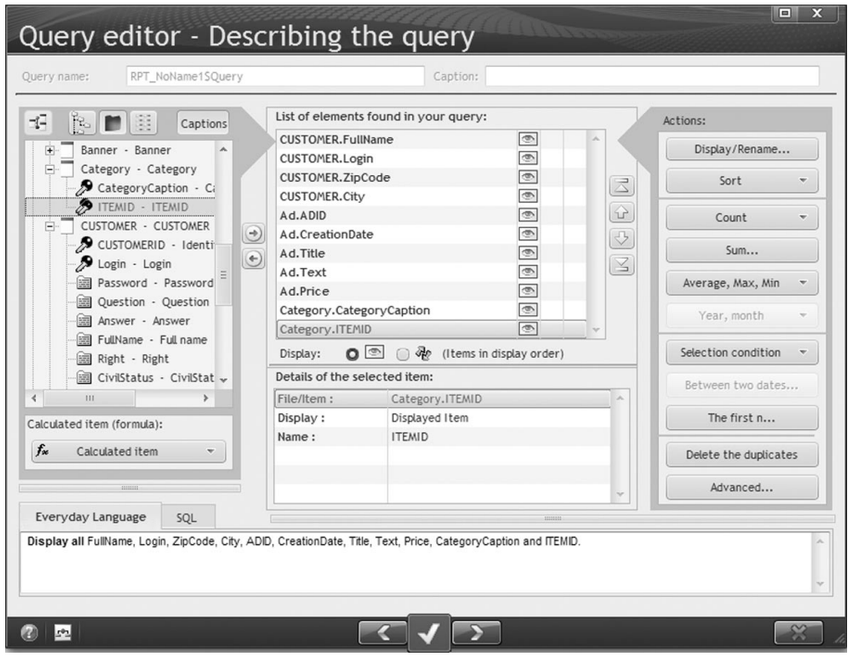
- Open (if necessary) the "Reports.WWP" project via "? .. Tutorial .. Printing (Exercise)".

Notes

If you do not want to perform the operations in this lesson, the corrected project is available from the following menu option: "? .. Tutorial .. Printing (Answers)"

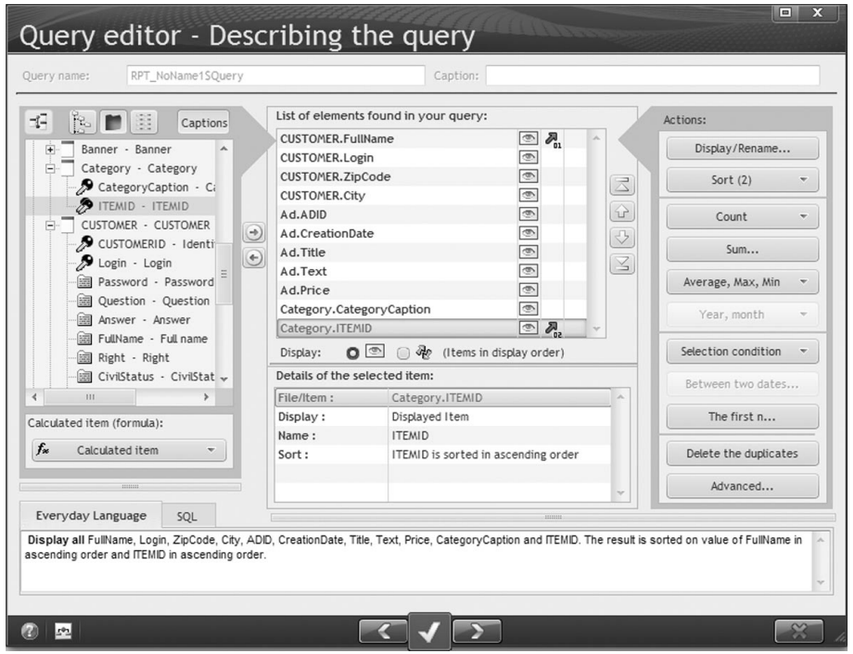
Step by step

- To create a new report:
  1. Select "File .. New .. Report".
  2. Select "Table". Go to the next screen.
  3. 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.
  4. 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 is displayed to 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.

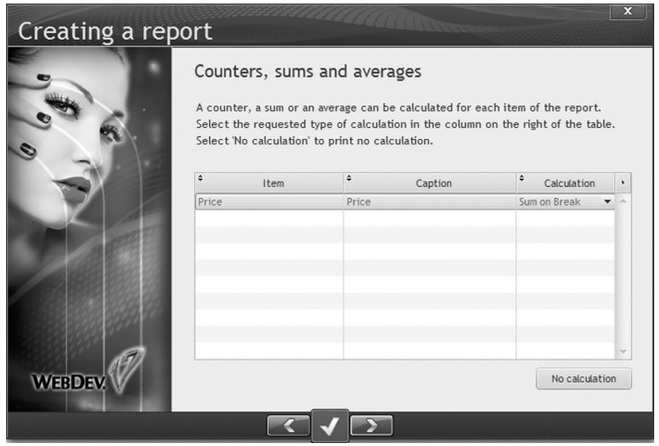
- We are now going to define the repartition of the items.
  - The information regarding the customer will be displayed in the break header 1: FullName, Login, ZipCode, City.
  - The information regarding the category of the ad will be printed in the break header 2: CategoryCaption, ITEMID.
  - The body block contains the details of each ad: ADID, CreationDate, Title, Text, Price.

The image below presents the repartition to perform:



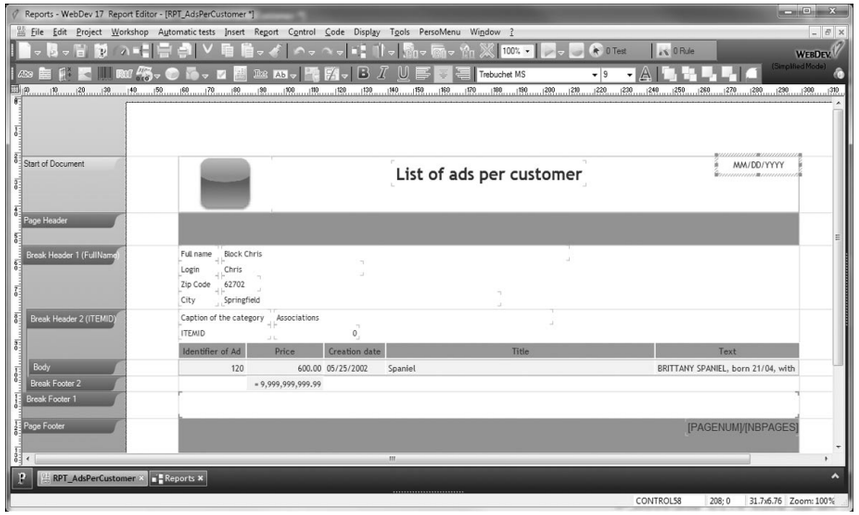
Go to the next screen.

- We must now specify whether totals or counts are found in the report. We are going to add up the prices proposed by the customers for their ads. This sum will be in included in a break. We are going to keep the sum on the price.



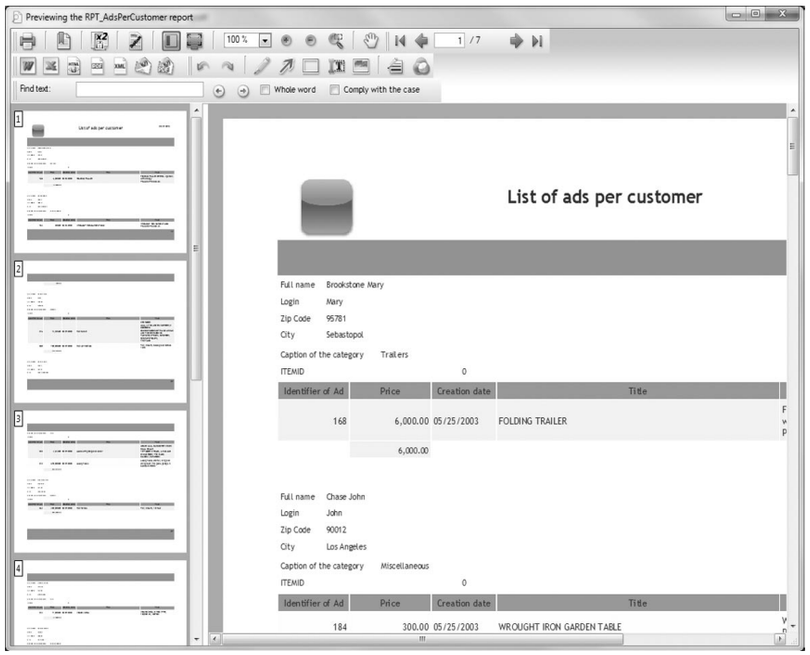
Go to the next screen.

- Keep the parameters proposed by default for the page format. Go to the next screen.
- Choose a preset template. Go to the next screen.
- Give a name and a caption to the report: "RPT_AdsPerCustomer" and "List of ads per customer" for example.
- Validate. Since the report is too large, accept the switch to landscape mode.
  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 ( ). Click "GO" to run the test.

The print preview displays the following report:



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. CROSSTAB REPORTS

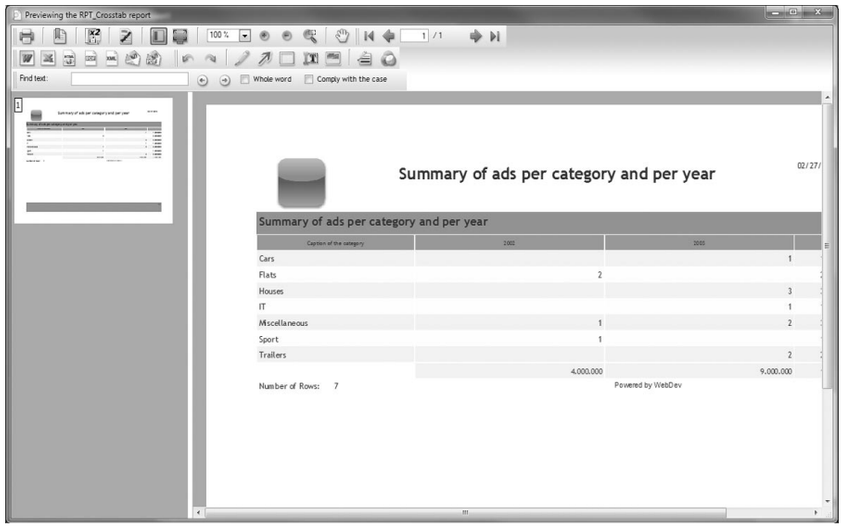
This lesson will teach you the following concepts ...

- **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.

- Open the "Reports.WWP" project via "? .. Tutorial .. Reports (Exercise)".

Notes

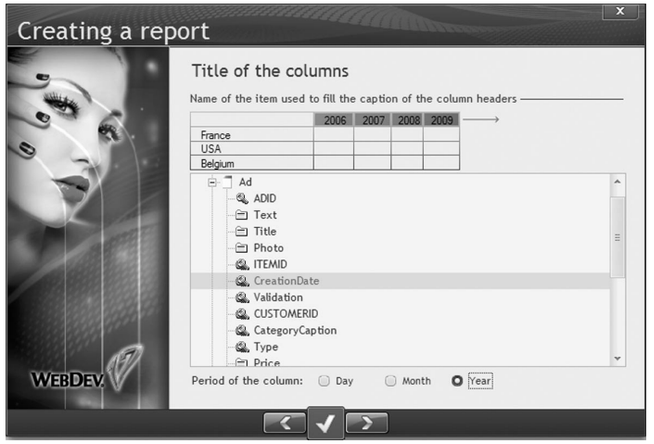
If you do not want to perform the operations in this lesson, the corrected project is available from the following menu option: "? .. Tutorial .. Reports (Answers)"

Step by step

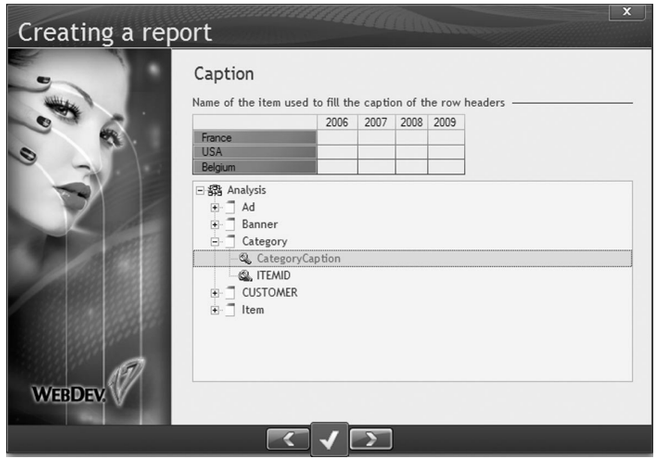
We are going to create a report used to list the number of ads per year and per category. The results will be sorted by year (horizontally) and by category (vertically).

- To create a new report:
  1. Select "File .. New .. Report".
  2. Select "Crosstab". Go to the next screen.
  3. 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):

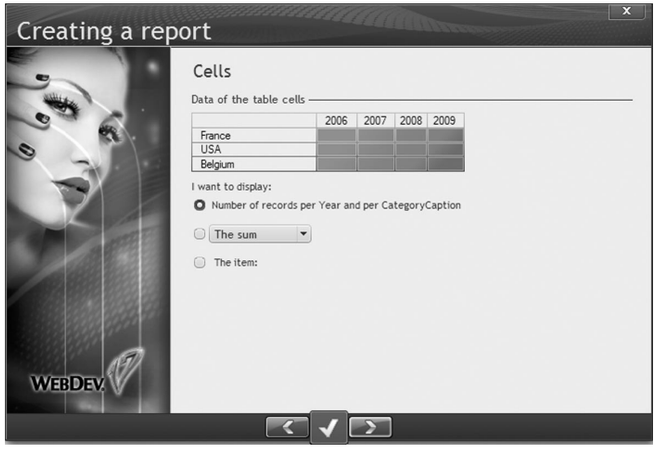


4. Choose "Year". Go to the next screen.
5. The wizard asks you whether the dates must be "bounded". Don't check anything as all the ads will be selected. Go to the next screen.
6. The wizard asks you to choose the item that will be used as data source for the rows. This enables you to define the second entry (column) of your crosstab.
Select the "CategoryCaption" item found in the "CATEGORY" file.



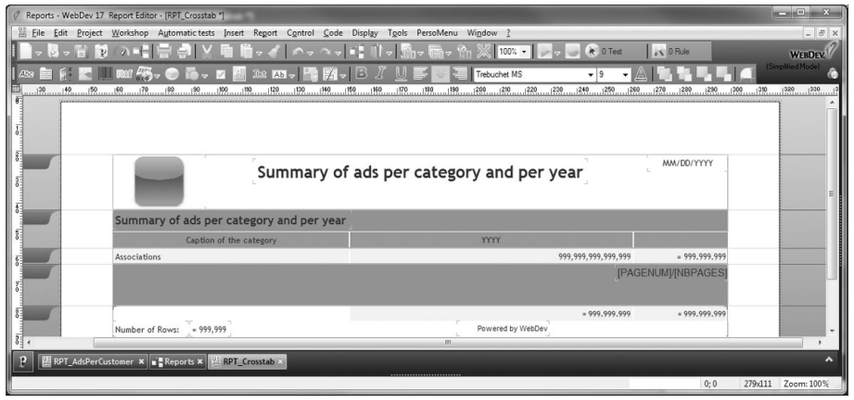
Go to the next screen.
7. The wizard asks you to select the type of information that will be displayed in the cells of the crosstab.

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".



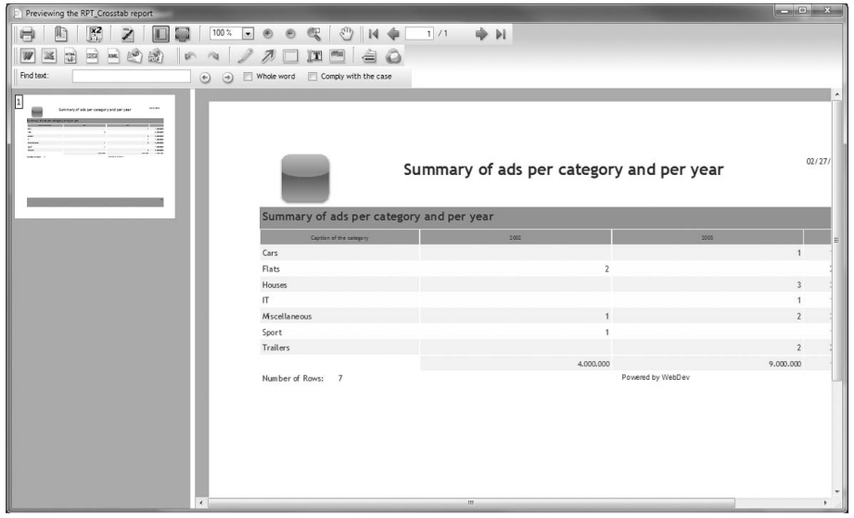
Go to the next screen.
8. The wizard proposes to calculate a sum per row and a sum per column. Accept this calculation. Go to the next screen.
9. Don't change the page layout. Go to the next screen.
10. Select a preset template. Go to the next screen.

▶ 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. Confirm the switch to landscape mode.
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 ( ). Click "GO" to run the test.

Caution, the execution of a CrossTab report may 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, 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 on 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 the server function named ***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 named Report_Table_Population
// 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:

- the function named ***fDataDir*** is used to create the file corresponding to the report in the directory of HyperFileSQL data files (this directory has write rights).
- the functions named ***Today*** and ***Now*** are used to create files with different names on the server. This enables you to distinguish the reports created by each Web user on the server.
- the function named ***fDelete*** is used to delete the created file as soon as it is displayed.

iDestination function

iDestination is used to select the output mode for 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.

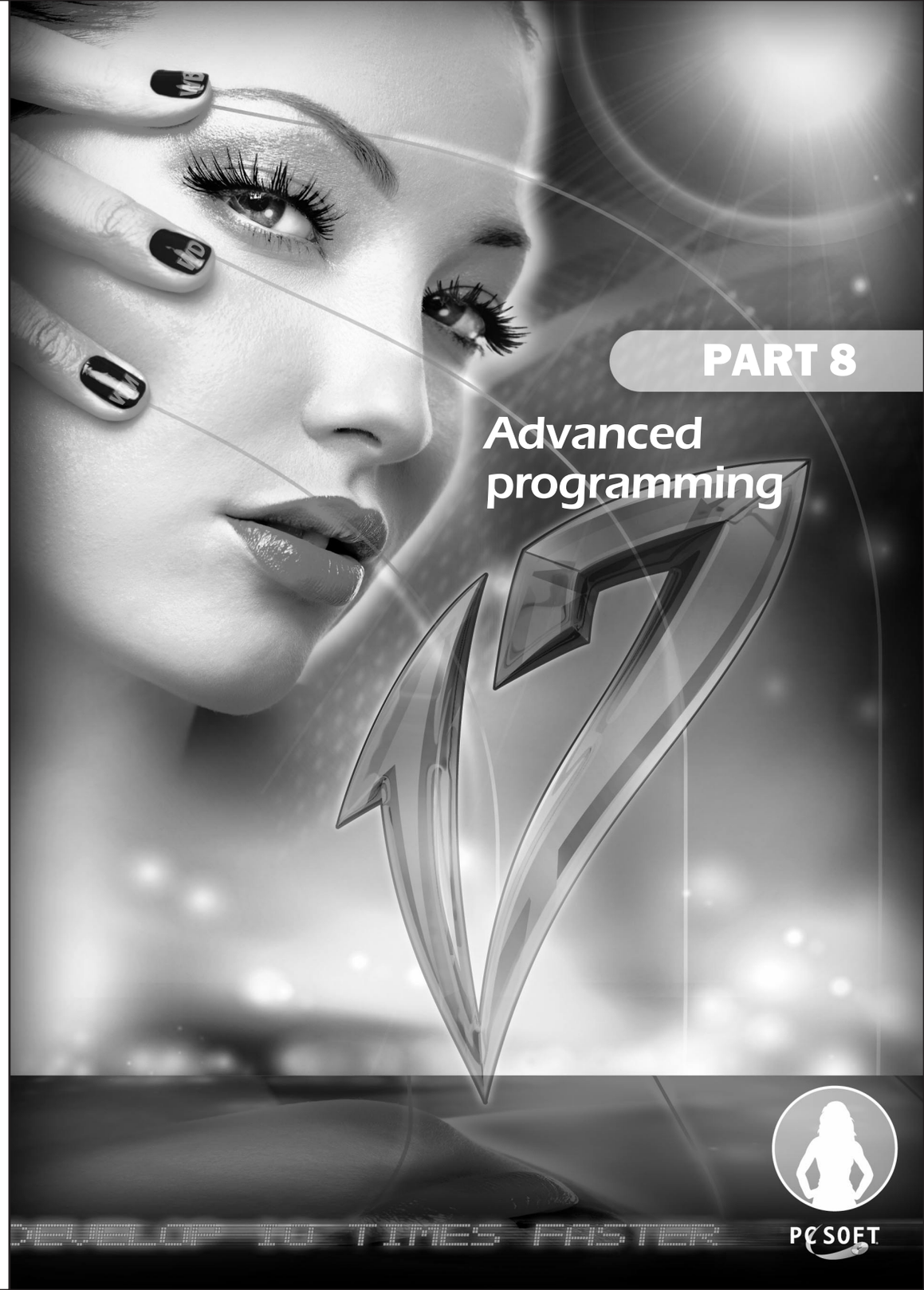
See the online help (keyword: "iDestination") for more details.

iPrintReport function

iPrintReport is used 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.



LESSON 8.1. WEB FEATURES

This lesson will teach you the following concepts ...

- The cookies.
- Upload and Download: transferring files.
- Securing the pages.
- Secured payment.



Estimated time: 30 min

Overview

This lesson presents several features specific to the WEB that can be included in a WebDev site.

For this lesson, we have created a project containing the data required to test the different features.

- Close the current project if necessary. The home window is displayed.
- In the home window, click "Tutorial" and select the project named "Web features". The project is loaded.
  Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. 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 afterwards.

When the Web user visits the same site several times, the site "identifies" the user from his cookies. This enables you to avoid requesting details already provided during a previous visit and to display 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 over.

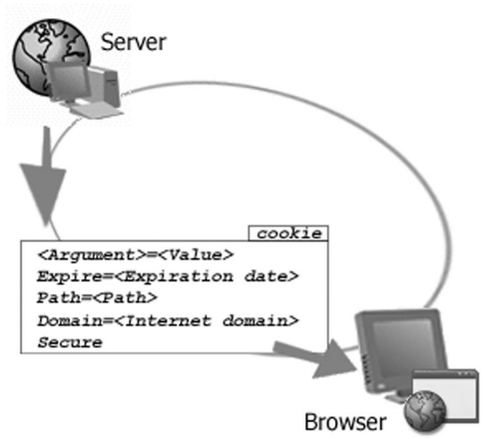
A cookie is used to store on the computer of the Web user various information such as the user name, the pages displayed by the user, the date of his last connection, the backup of his options, ...

This information, saved as cookies, will be read during the next connection to the site. The site will then 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 the 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

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 the next page is displayed. 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:
  1. Run the test of the "Web_features" project.
  2. Click the link named "The management of cookies".
  3. Enter your name and click "Save".
  This name will be automatically displayed the next time this page is loaded.
- Go back to WebDev, open the "PAGE_Cookies" page and see the various pieces of code associated with the button and the page.

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 where the user will have to select a file. Furthermore, the default file type cannot be specified.

The file upload can be performed in the following cases:

- 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 on the server.
- in a "Document management" site, each Web user can propose all types of documents: text, ... These documents are selected on the computer of the Web user and uploaded to the server in order to be made available to all the Web users.
- ...

Managing the upload with a single-file upload control

To manage the file upload in your WebDev site, an Upload control (single-file for example) 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 "Web_features" project, open the "PAGE_UploadDownload" page and run its test.
2. See the code of the different buttons.

The file download

The download is the reverse operation of the upload. This is the most frequently used method on Internet.

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. The file download can be canceled by the Web user at any time.

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 directly viewed in the browser (text or PDF files for example).

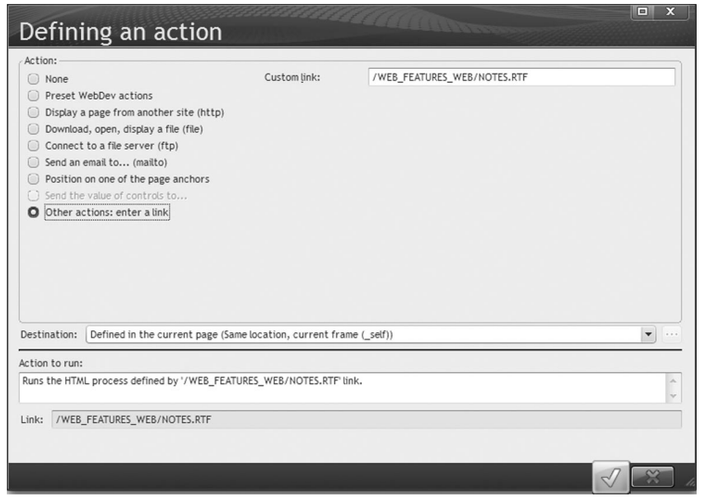
The file download can be performed in the following cases:

- in a site for selling or renting DVDs. For example, the trailer of a movie can be downloaded to the computer of the Web user.
- in a "Document management" site. Each Web user can download different types of documents. These documents are selected on the server and downloaded to the computer of the Web user.
- ...

Managing the download

Several methods can be used to manage the file download:

- using a "Link" control (Other type), with the address of the file to download:



- using the function named ***FileDisplay*** in server code.

Practical example

- To understand the operating mode of the download:
  1. In the "Web_features" project, open the "PAGE_UploadDownload" page and run its test.
  2. See the code of the different buttons.

Securing the pages

A site may 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 the site administrator), ...

In most cases, an identification will be requested (login and password). For example, in a business site, an identified Web user will be able to view the history of his orders ... The names of the users and their passwords can be stored in a HyperFileSQL data file or directly entered in the server code of the WebDev project.

- To see an example used to identify a Web user in a WebDev page:
  1. In the "Web_features" project, open the "PAGE_Security" page and run its test.



This page is used to access a secured area intended for an administrator. The following pages (if the identification is successful) will be protected by the TSL/SSL protocol (see "Securing data and pages via the TSL/SSL protocol" later in this lesson).



Caution !

A TSL/SSL certificate must have been previously installed on your Web server. See the online help (keyword: "SSL") for more details.

2. Enter a user name and a password, and see what happens.
3. Then, enter the "ADMIN" login and the "ADMIN" password (caution: the identification is case sensitive), and see what happens.

4. Go back to WebDev and study the different 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 HyperFileSQL 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(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(Attempt, ...
             (4-Attempts))
           Error("Wrong user name or password.", ...
                                     "Please retry.")
       ELSE
           // The user has entered the proper user name
           // and the correct password
            PageDisplay(PAGE_SecureOK, EDT_USERNAME)
     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 there is concordance, the "secured" area is displayed in the browser of the Web user (***PageDisplay***). Otherwise, an error occurs and the Web user must resume his input.

Securing the data and the pages via TSL/SSL

By default, the data exchanged between the computer of the Web user and the Web server is not protected. The data confidentiality is not guaranteed.

Several systems enable you 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/wd170awp/wd170awp.exe/connect/customers).

Implementing secured transactions via the TLS/SSL protocol

To implement secured 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 should be secured: credit card numbers for example. Indeed, the secured mode is slower.

The secured mode is implemented when opening a new page via a button. All you have to do is use the function named ***SSLActive*** in a browser code.

As soon as the secured page is opened, all the actions will be performed in secured mode (which means encrypted), regardless of the objects used (link, table, looper, clickable image, ...).

▶ To understand the use of ***SSLActive***:

1. In the "Web_features" project, open the "PAGE_SecureOK" page.

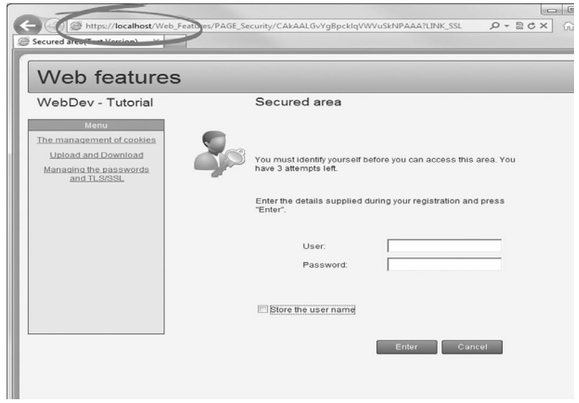
2. Study the code of the "Enter" button.



Caution !

To run the test of "PAGE_SecureOK", an SSL certificate must be installed on your Web server.

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



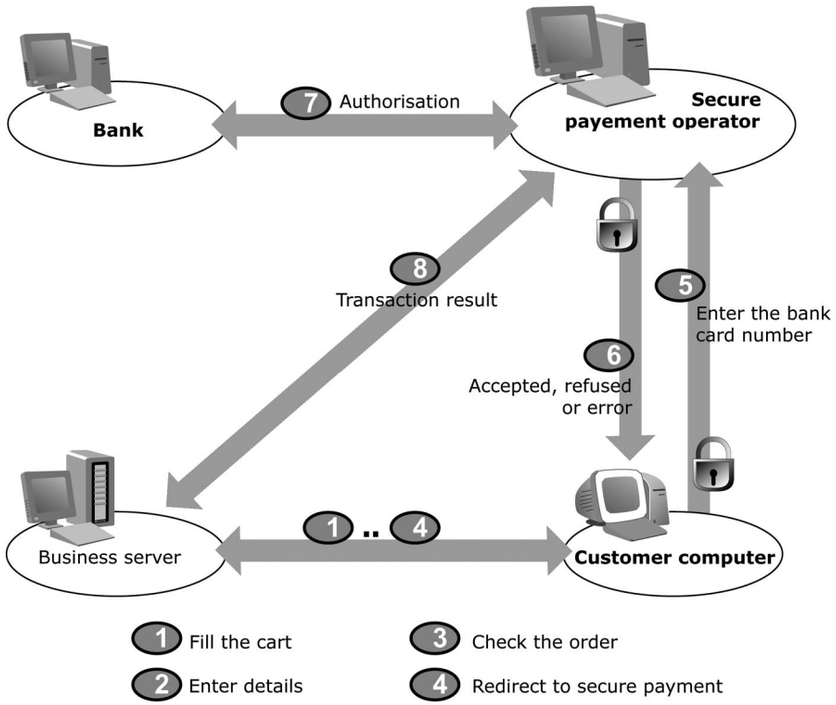
See the online help (keyword: "SSLActive") for more details.

Secured payment

Most business sites allowing Web users to perform online purchase use a system for online payment by credit card.

Secured 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 secured 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 secured payment. The transmission of data is protected via SSL to ensure confidentiality. 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 providers !

System for secured payment in a WebDev site

General procedure

In most cases, the following operations must be performed when implementing a solution for secured 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 an example.

The "PayBox component" component will be used in this Tutorial. The component has already been imported into our example. We won't go into step by step details about integrating this component, this topic will be presented in the "The external components", page 380.

▶ To understand the operating mode of online payment via PayBox.
1. In the "Web_features" project, open the "PAGE_SecuredPayment" 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 shall not 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 recommendation of PC SOFT.
WebDev can be used with any provider of secured payment.

LESSON 8.2. THE TIMERS

This lesson will teach you the following concepts ...

- Benefit of timers.
- programming the timers.



Estimated time: 5 min

Introduction

A timer is used to periodically and automatically call a process performed by a browser procedure. This enables you to refresh a page on a regular basis or to display the time in the status bar of the browser.

The timers can be used in browser code only and they are automatically "stopped" when the page that started them is closed or refreshed.

Programming the timers

Practical example

▶ To understand the operating mode of the timers:

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

2. In the home window, click "Tutorial" and select the "Managing the timers" project. The project is loaded.
Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. Managing the timers".

3. Open the "PAGE_Timer.WWH" page.

4. Run the test of this page.

Creating a timer

To create a timer, you must:

1. Write a browser procedure (local to the current page or global to the project).

2. Then, call it with the WLanguage function named ***Timer***. By specifying the call frequency, the browser procedure will be run every ***N*** hundredths of a second.

▶ In our example, the "PAGE_Timer" page contains the local browser procedure named **DisplayTime**. This procedure is used to display the system time:

```
PROCEDURE DisplayTime()
  EDT_TIME = Left(TimeToString(TimeSys()), 8)
```

The code of the "Start" button (beside the clock) is used to create the timer in order to display the time in real time. The syntax is as follows:

```
// Runs the DisplayTime procedure every second
// The time will be displayed every second
MyTimer = Timer("DisplayTime", 100)
```

The procedure called can perform modifications in the page, check the input ...

Stopping a timer

To stop the execution of a timer, use the WLanguage function named ***EndTimer***.

▶ In our example, the "Stop" button (beside the clock) uses the following syntax:

```
// Stop the time display
EndTimer(MyTimer)
```

See the online help (keywords: "Timer", "EndTimer") for more details.

LESSON 8.3. 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.

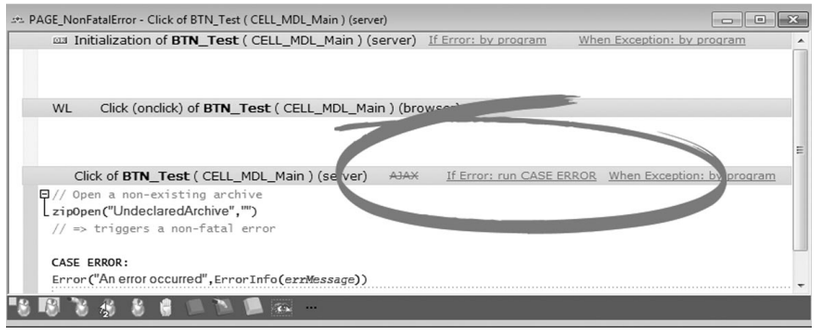
Operating mode

Two operations are performed when an error is detected by a WLanguage function:
- an error value is returned by the function (***fOpen*** returns "-1" if the selected file cannot be opened for example).
- the error is detected by the WLanguage (the ***ErrorDetected*** variable corresponds to *True*); the details of this error can be retrieved by ***ErrorInfo***.

This second operation can be automatically managed by WebDev via the feature for error management.

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 the function named ***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 an 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

- Close the current project if necessary. The home window is displayed.
- In the home window, click "Tutorial" and select the project named "Discover the management of errors". The project is loaded.
  Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. Discover the 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 a non-existing archive).
    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.

- To run the test of the project, click  .

LESSON 8.4. THE INDIRECTIONS

This lesson will teach you the following concepts ...

- **What are indirections?**
- **Use example.**

 Estimated time: 5 min

Overview

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.

Example

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 to a procedure used to make a control invisible
INVISIBLE(ControlName)

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

To optimize the process time, the indirections can accept a parameter specifying the nature of the object used:

```
ControlName is string = "MYCONTROL"
{ControlName, indControl}..Value = "New value"

FileName is string = "MYFILE"
ItemName is string = "MYITEM"

HReadFirst(FileName, ItemName)
Value = {FileName + "." + ItemName, indItem}
```

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

LESSON 8.5. THE MANAGEMENT OF 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 allow you to manage the incoming and outgoing emails in server code. Furthermore, WebDev enables 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 enables 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 managing the emails is used by most of the Microsoft applications, especially by Microsoft Exchange.

In this lesson, we shall only present the management method via the POP3/SMTP protocols. Indeed, this mode is frequently used all over the world.

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

All the functions for email management start with "Email".

The POP3 and SMTP protocols

The POP3 protocol is used to receive the emails.

The SMTP protocol is used to send the 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***.

Starting an email session

- To start an email session, use ***EmailStartSession***:

```
USER is string = "User"
Password is string = "Password"
POP_Serv is string = "pop.wanadoo.eu"
SMTP_Serv is string = "smtp.wanadoo.eu"
UserName is string
IF EmailStartSession(USER,Password,POP_Serv,SMTP_Serv)THEN
   UserName = USER
ELSE
   UserName = ""
   Error("Unable to establish the connection")
END
```

The function named ***EmailStartSession*** enables you to read and send emails. See the online help (keyword: "Email, WLanguage functions") for more details.

Sending an 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 the advanced Email variable.

See the online help for more details.

Reading an email

Now that we are connected to the server and that we have sent a message, let's see how we can read an email.

Like a data file can be browsed by ***HReadFirst***, ***HReadNext*** ... the incoming emails can be read by ***EmailReadFirst***, ***EmailReadNext*** ... See the online help (keyword: "Email, Reading an email") for more details.

The ***Email*** structure is initialized for each email read. The variables of this structure correspond to all the characteristics of the current email.

Example of code:

```
// Connection to the server established by EmailStartSession
// During the connection, we retrieved
// the name of the user in the UserName variable
// Read the first email
Message is string
EmailReadFirst(UserName)
IF NOT Email.Out THEN
   IF Email.HTML= " " THEN
      Message = Email.Message
   ELSE
      Message = "Email in HTML format:" + CR + Email.HTML
   END
   LooperAddLine(LOOP_EMAIL, Email.Sender, ...
         Email.ReceiveDate, Email.Subject, Message)
END
```

Disconnection

To disconnect, all you have to do is use ***EmailCloseSession***:

```
IF UserName <> "" THEN
   EmailCloseSession(UserName)
   UserName = ""
END
```

Other possibilities

You can also:

- send attachments 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 439).

See the online help (keyword: "Email, Managing the emails") for more details.

LESSON 8.6. OOP

This lesson will teach you the following concepts ...

- **Object-Oriented Programming concept.**
- **Examples of object declaration.**



Estimated time: 30 min

Concepts

Object-Oriented Programming (OOP) is designed for better code reusability. The programs developed in OOP are structured: they are made of 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 structure of data where the procedures will act.

To use "object" 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.

Once 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 enables 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 the methods of the parent classes.

Constructor and Destructor

The notion of **Constructor** and **Destructor** is important because it allows 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 in order to handle the classes.

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. Display (if necessary) the project explorer ("Display .. Toolbars .. Panes .. Project explorer").
2. Select "Classes".
3. In the popup menu, select "New class".
4. In the window that opens, enter the name of the class and validate. The code of the class can be directly entered in the code editor.

A class has the following format:

```
MyPager is class
  NbButton is int
            // Number of buttons in the pager
  ButtonName is string
            // Name of the buttons in the pager
  Next is string
            // Name of the next button
  Previous is string
            // Name of the previous button
END
```

"Pager" is the name of the class.

"NbButton", "ButtonName"... 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 opens, enter the name of the method and validate. The code of the class can be directly entered in the code editor.

A method has the following format:

```
PROCEDURE PreviousButton()
// Display the previous button
PreviousButton is int = 0
// Increment the subscript of the active button
PreviousButton = ActiveButton - 1
// Call the ButtonClick method
ButtonClick(PreviousButton)
```

The name of the method is "PreviousButton".

Declaring and handling objects

In the processes of the window, an object is declared at the same time as the other variables:

```
GLOBAL
    Pager is MyPager
```

To refer to a member of the "File" object, use the following syntax

```
<ObjectName>.<name of member>
```

The object is handled as follows:

```
// Display the previous buttons
Pager.PreviousButton()
Pager.DisplayPage("DisplaySearch")
```

We won't go into details about OOP in this tutorial.



LESSON 9.1. REQUIRED CONFIGURATION

This lesson will teach you the following concepts ...

- Configuring WebDev to follow part 9 of the Tutorial



Estimated time: 5 min

Overview

Since the beginning of this tutorial, you have been using WebDev in a simplified environment. This enabled you to discover the main features of WebDev.

In this part, we are going to study the advanced concepts of project management.

To perform the operations presented in this part, WebDev should be configured to use the standard environment. This gives you the ability to access all the features of the product.

Reminder: WebDev gives you the ability to configure the environment. Several modes are available:

- **Retrieve the configuration of your environment xx**: This mode uses the features available in version xx (if version xx of WebDev is installed on your computer).
- **Simplified environment**: This mode enables you to discover the main features of WebDev.
- **Full environment**: This mode proposes all the features of WebDev.

At any time, regardless of the type of environment used, you have the ability to add or delete the access to some unused features.

Implementation

▶ To use the full environment of WebDev:

1. Start WebDev 17.
2. Select "Tools .. Options .. Options of the environment".
3. Click "Restart the wizard for configuring the environment".
4. Select "Full environment".
5. Go to the next screen to validate your choice.
6. Validate the options of the environment.

That's it, WebDev is configured to follow this part of the Tutorial.

LESSON 9.2. DASHBOARD

This lesson will teach you the following concepts ...

- **Overview**
- **View mode**
- **Options of the dashboard**



Estimated time: 20 min

Overview

The project dashboard is a main 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 "My_library". If this application was not previously created, a corrected version is available via the following option: "? .. Tutorial .. Site with data (Library)".



Note

Required configuration

To follow this lesson, you must use the environment in full mode. See “Required configuration”, page 351 for more details.

The different elements of the dashboard



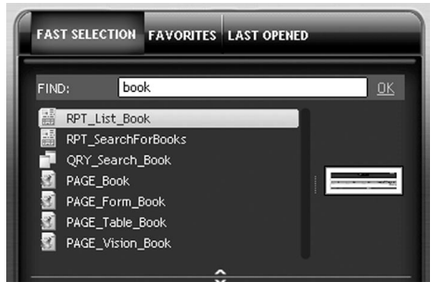
The project dashboard includes:

- buttons
- lists of elements
- counters

The lists

The dashboard includes different lists. These lists are used to optimize the access to the project elements.

For example, **the "Fast selection" list** is used to quickly find a project element. All you have to do is enter some letters found in the name of the sought element. The selection is automatically performed and a preview enables you to choose the requested element.



To find the PAGE_Book page: type "Book" and make your choice. A double click performed on the name of the element enables 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 be available.

Another interesting list: the favorites. Who does not have a window or a source code that is frequently used in an application? A window that groups the main features for instance? To avoid losing time searching for this object, all you have to do is include it in the favorites.

For example, the "PAGE_book" page found in our site will now be included in the favorites:

1. In the dashboard, select "Favorites" and click the link for displaying the favorites pane.

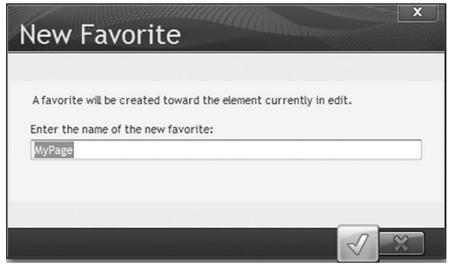


Note

To display the Favorites pane, you also have the ability to select "Display .. Toolbars .. Panes .. Favorites".

2. Display the "PAGE_Book" page in the editor (double-click its name in the project explorer for example).

3. In the "Favorites" pane, click the "+" button. You can now enter the name of the favorite (MyPage for instance).



Validate. The favorite is displayed.



Note

If several developers work on the same project, you have the ability to view "All the favorites".

4. In the dashboard, you also have the ability to find your favorite by pressing F5.

The buttons

In part 2, we took a look at a button used when testing. Let's take a look at the other buttons.

The information regarding the **global quality of the project** is grouped in a specific counter.



In the counter, you will find:

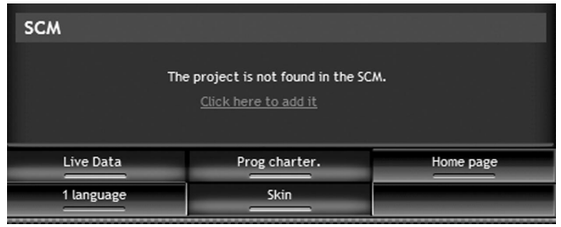
- A progress bar indicating the overall quality of the project,
- A progress bar for backup. All you have to do is click the progress bar to save the project.
- A progress bar for cleaning. All you have to do is click the progress bar to clean the project.
- Warning buttons: components not updated, compilation errors, information, warnings.

The dashboard also includes several optimization buttons:

- **Static audit**: If this button is red, it means that the static audit has not been run since a long time on your project. The static audit is used to get the 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 list the project elements that are not used (page used for test during the development of the site for instance).
  - **detect whether an optimization of queries is required**: the static audit informs you if your project uses queries that can be optimized by a simple modification of the analysis (adding a composite key for example).

  Don't hesitate to run the static audit: it can improve the performance of your site.
- **the infrastructure audit**: If this button is red, it means that the infrastructure audit has not been run since a long time on your project.
- **the dynamic audit**: If this button is red, it means that the dynamic audit has not been run since a long time on your 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
  - ...

The dashboard also enables you to find out and modify some characteristics of the project. These elements are represented as lights: the light is green? It means that the feature is enabled in your project. Otherwise, a click performed on the light is used to enable or configure the feature. Let's take a look at the available elements:



- **SCM**: Used to find out whether the project is shared via the Source Code Manager. This feature will be presented in the next lesson ("SCM", page 360).
- **Live data**: Enables you to use the content of the data files found on the development computer in the pages, the reports, ... handled in the editor. This feature is very useful to define the size of the controls found in a page.
- **Programming charter**: As already seen in part 1, the programming charter is used to standardize the names of the elements used in a project. As a programming charter is used by our project, this light is green.
- **Home page**: This option indicates whether a home page was included in your site. This element is important to improve the referencing of your site.
- **Languages**: enables you to find out the number of languages supported by your project. A single language is supported in our case. A click performed on this light enables you to add new languages to your project. The management of multilingual projects will be presented later in this tutorial ("Multilingual", page 413)
- **Skin**: As you already know it, the skins, the styles are used to easily define the style book of your site. A click performed on this light enables you to access the screen for configuring the skin and the style of your application.

The counters 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 HyperFileSQL Control Center, that is mainly used to control the data files found in the HyperFileSQL 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.

We will only present the elements in relation with the dashboard. See "Control Centers", page 371 for more details.

The task counter

The task counter represents the number of tasks currently in progress in the project. These tasks are referenced in the Project Monitoring Center.

These tasks correspond to:

- features that must be added to the project
- the forthcoming development, ...

The request counter

The request counter represents the number of requests currently found in the project. These requests are referenced in the Quality Control Center.

The message counter

The message counter represents the number of messages associated with the project. These messages are visible in the messaging of WebDev. These messages are automatically sent when working with the Source Code Manager for example. This feature enables you to be automatically notified when an element is updated in the SCM.

The rule counter

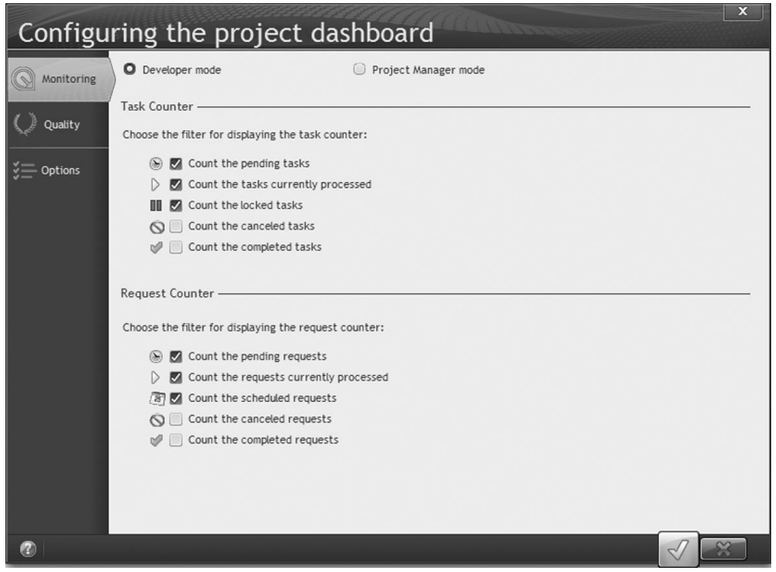
The rule counter is used to find out the number of business rules defined for the project. The business rules will be presented later in this tutorial. You, the project developer, can enter the business rules yourself.

Configuring the dashboard

We have presented the main categories of information displayed in the dashboard. WebDev proposes two modes for viewing this information: Developer mode and Project Manager mode. To change mode, all you have to do is click "Mode" at the top of the dashboard.

The levels of alert for the dashboard can be configured for each mode.

To configure the dashboard, select "Dashboard options" from the popup menu of the dashboard.



The following elements can be configured for the two modes:

- the buttons/lights regarding the tests
- the buttons/lights regarding the optimization
- the counter of tasks and the counter of requests
- the cleaning
- the backup
- the quality of the project
- the position of the dashboard, ...

LESSON 9.3. 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 among developers and to find out the full history of the modifications performed (in the code, in the interface, ...).

Note

Required configuration
To follow this lesson, you must use the environment in full mode. See “Required configuration”, page 351 for more details.

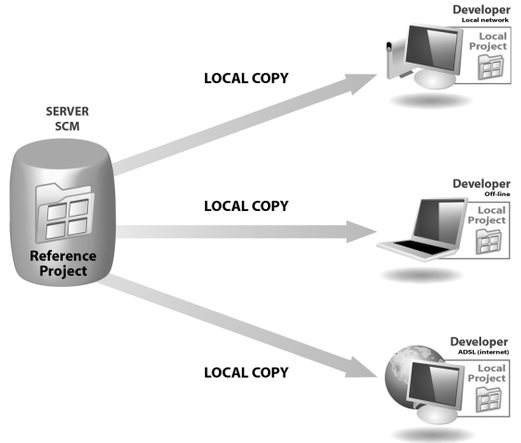
SCM (Source Code Manager)

Principle of SCM

The Source Code Manager is used to store and share the projects and their elements.

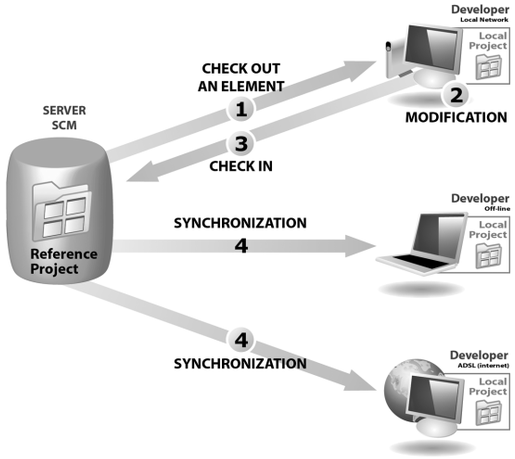
The principle is as follows:

- A reference version of each one of your projects is found on a server. All these versions are called a "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.
- Each developer gets exclusive rights on this element: all the requested modifications can be made to 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 in 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 enables you to find out the history of all the modifications.
The SCM can also be used to manage and control the elements shared among 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 recommend you 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

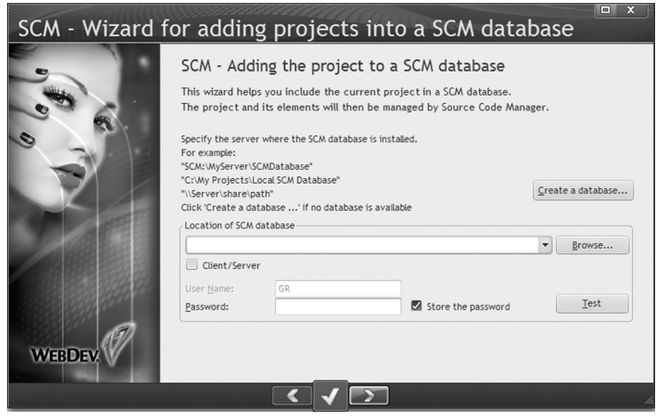
To use an existing project with the Source Code Manager, all you have to do is include this project in the SCM database.

▶ We are now going to include the "Pages_and_controls.WWP" project in the SCM database:

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

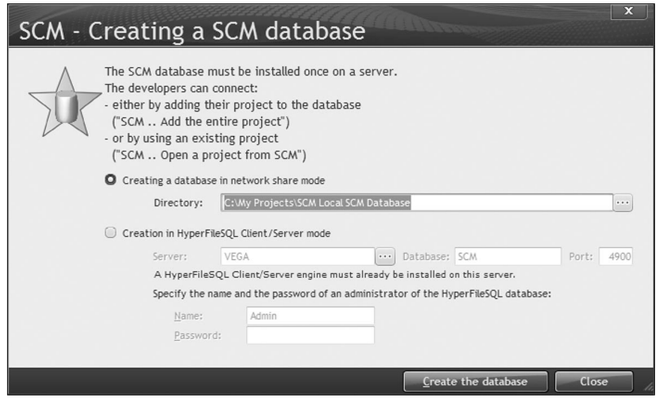
2. In the home window, click "Tutorial" and select the project named "Pages and controls". The project is loaded.
Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. Pages and controls".

3. Select "SCM .. Add the entire 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 HyperFileSQL Classic format (local or network) or in HyperFileSQL Client/Server format. We are going to create a SCM database in HyperFileSQL Classic format.

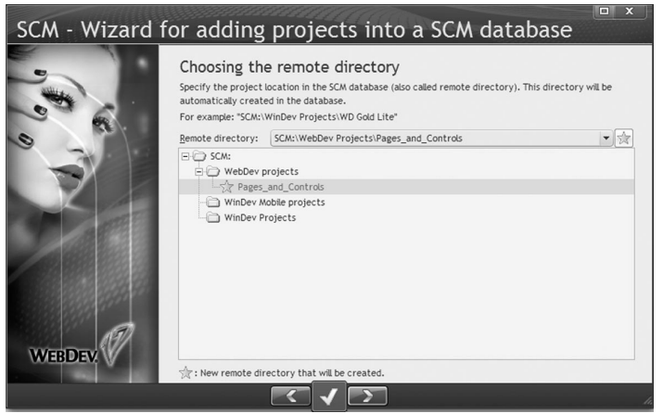


Notes

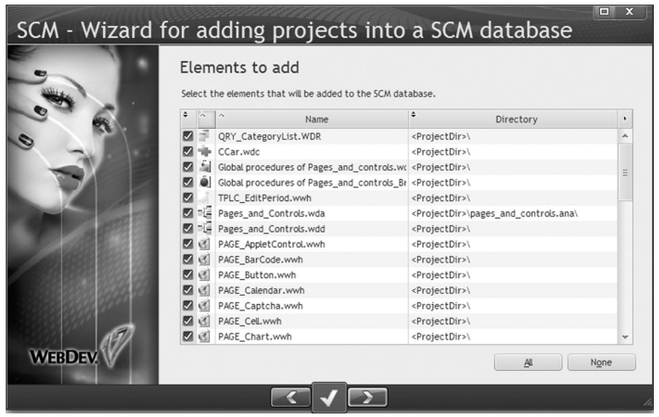
If the SCM database is in HyperFileSQL Client/Server format, this SCM database can be used remotely.

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. Our project can be included 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.
A help window is displayed. Read and validate this window.

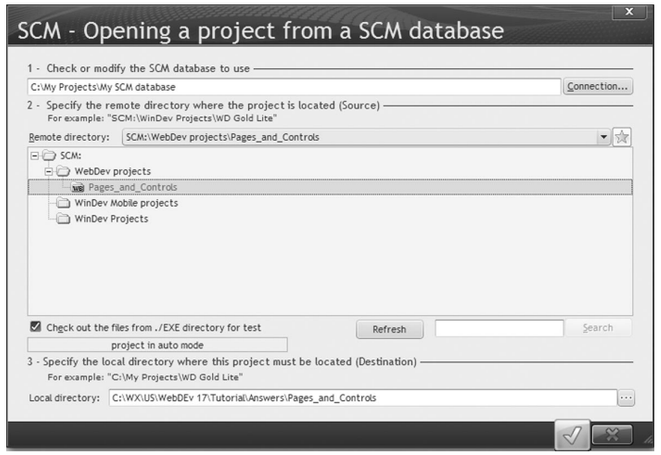


Notes

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 among the projects. Therefore, the same element is checked in once only into the SCM.

Opening a project from the SCM

To work on a project found in the Source Code Manager, you must retrieve a local copy of this project. To do so, open the project from the Source Code Manager ("SCM .. Open a project from SCM").



This operation must be performed once only by each developer who uses the project.
The developer who added the project to 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 we 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. The modified element is 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 attempt 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 automatic check-out.

▶ To make sure that the automatic check-out is enabled, select "Tools .. Options .. General options of WebDev". In the "General" tab, check whether "Checking out elements during the first modification" is selected. If not, enable this option.

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 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 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 has been 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.

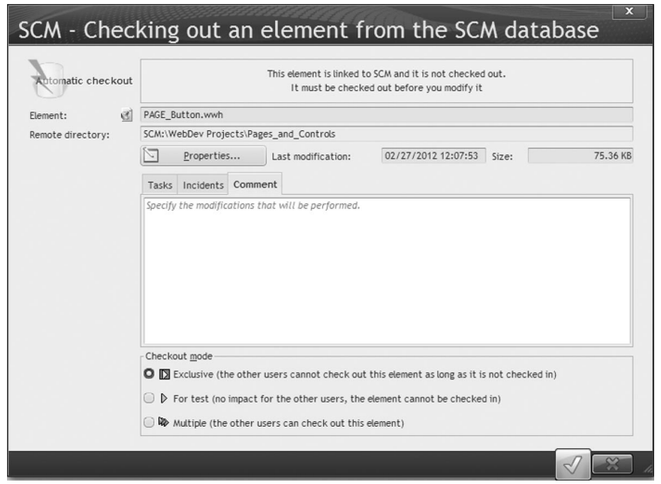
▶ To enable the automatic check-out of the project, select "SCM .. Project management .. Manage the project check-out automatically".

Checking out an element

We are now going to start working 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 "Send to server - (Enter)" button ("Code" from the popup menu).
4. We are going to enter a comment before the line "Info ...". Position the cursor and press the ENTER key.
5. The window for automatic check-out is displayed:



6. Three check-out modes are available:
   - **Exclusive** (recommended mode): nobody can check out this element until it is checked back in. The element can be checked out for test only.
   - **For test**: the element can be modified but the modifications will not be checked back in.
   - **Multiple**: the element can be checked out by other users. In this case, the differences between the different versions of the element can be displayed 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 checkout.

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 made to a checked-out element are not visible to the other developers.

If another developer runs the checked-out element, the element that is currently found in the SCM database will be used.

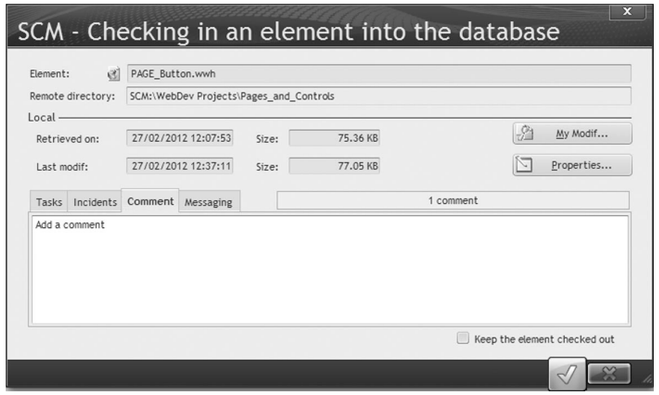
This enables you to modify an application while keeping a steady version in the SCM database.

- Modify the checked-out page.
  1. Enter the following comment: "// Display a message". Close the code window.
  2. Change the location of the controls, modify the captions, add controls or code, ...
  3. Save your page (CTRL + S).
- Run the test of your modifications.

Checking in the checked-out element

Now that the modifications are over, we are going to check in the page back into SCM. Then, your modifications will be accessible to the other developers.

- Select "SCM .. Check in the element". 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).
- 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 in 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 ("Project .. Project description").

The main 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 in when the project is closed.
- *Propose to check in and to get the latest version of the elements when generating the executables, libraries, components, ...*
  When creating a library, this option is used to display the list of checked-out elements in order to check in and to get the latest version of the elements.
  Therefore, the component or the library can contain the most updated elements.
  By default, the library is generated with the project elements currently found on the local computer.



Notes

Merging code

An element can be compared to one of its earlier versions. This enables you to compare the code in order to retrieve a section of code that was "lost" or accidentally deleted by another developer.

Off-line mode (or mobile mode)

The SCM allows you to work in off-line mode.

This mode allows a developer who uses 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, select "SCM .. Remote work .. Disconnect for a mobile use".
- during the reconnection, select "SCM .. Remote work .. Reconnect and synchronize". Then, all you have to do is check in the modified elements.

In mobile mode, two solutions are available for checking out elements:

- No element is checked out from the SCM. 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 the 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, ...).
- view the history of an element.
- view the status of the elements.
- perform backups.
- grant rights to the different users of SCM.
- list the projects in which you are involved in order to dissociate from them (if necessary).

▶ Start the SCM administrator ("SCM .. SCM Administrator"). All the project elements are listed.

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

LESSON 9.4. 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 HyperFileSQL Control Center was presented in a previous chapter. This Control Center (redistributable) is mainly used to manage the HyperFileSQL 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 uses a specific database. This database can be:

- a HyperFileSQL Classic database: the path of this database was specified the first time WebDev was started. By default, this database is installed in a sub-directory of WebDev 17.
- a HyperFileSQL Client/Server database.

This information can be modified at any time from the options of WebDev ("Tools .. Options .. General options of WebDev", "Parameters of Control Centers" button).



Note

Required configuration

To follow this lesson, you must use the environment in full mode. See “Required configuration”, page 351 for more details.

The Project Monitoring Center

The Project Monitoring Center is the heart of the organization. The Project Monitoring Center helps you organize and schedule a project, from design to delivery. It enables you to define the tasks that must be performed in a project, the developers, the durations, the progress status of the project, ...

To test the Control Centers, we will be using our application named "My_Library".

▶ Start the Project Monitoring Center ("Tools .. Project Monitoring Center"). 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.



Notes

The Project Monitoring Center helps you manage and schedule any project or task (development, business or other).

1. Create a new project ("Management .. Manage the projects", "+" button).

2. Enter:

- the name of the project ("Tutorial" for example)
- its description ("My_Library" for example)
- the type of application: Internet/Intranet
- the path of the project.
  In our case, specify the path of a project outside SCM (for example, path of "My_Library.wwp" project in the "Tutorial\Answers\ My_Library" sub-directory of WebDev).



Note

If the project is available in 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 spool (a new one can be created).
- the color associated with the project (used to easily identify the different projects).

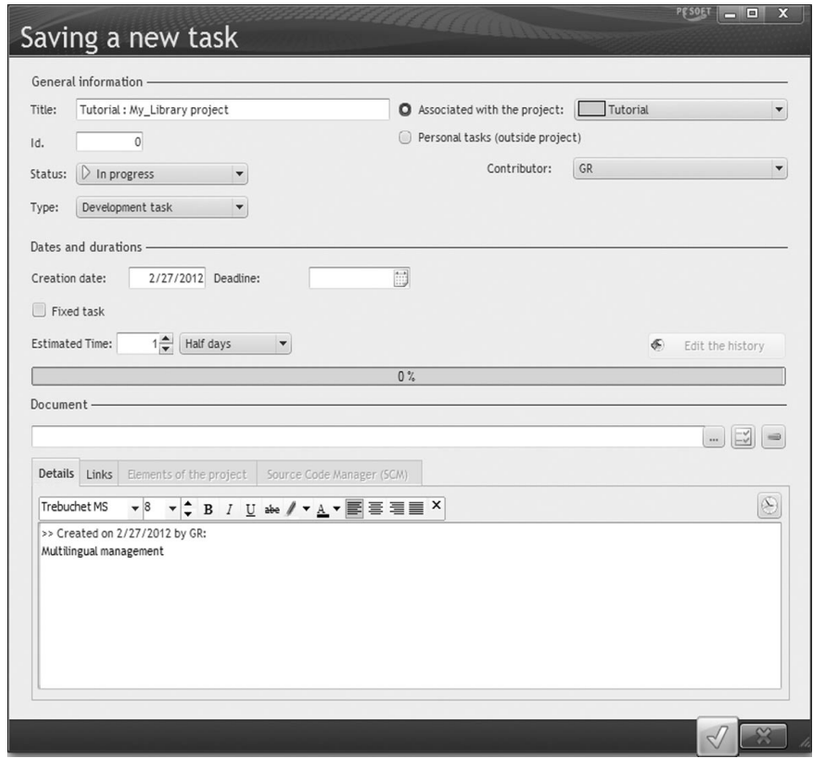
3. The screen used to select the contributors of your project is automatically displayed. Keep yourself and validate.

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 to make our project multilingual:

1. Select "Tasks .. Create a new task".

2. Specify the following characteristics:



3. Validate. The task appears in the task list. The task is "InProgress", 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, 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

The time management is extremely important. Where is time spent in a day? How to find out without adding constraints, without requiring some fastidious time keeping from team members and without making people feel like somebody is watching over their shoulder?

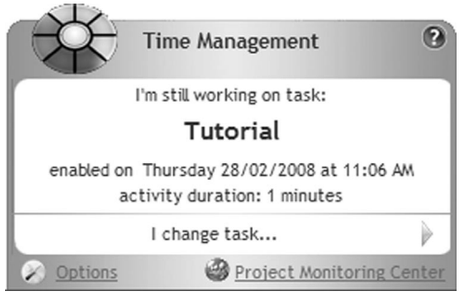
Entering the time spent on the tasks is an interesting feature of WebDev. This feature is linked to the use of the Project Monitoring Center.

The principle is straightforward: the task list is entered in the Project Monitoring Center.

▶ To enable the help for time management:

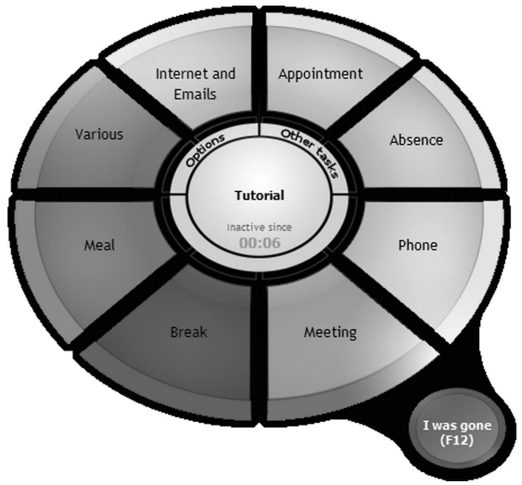
1. In the Project Monitoring Center, select "Parameters .. Options of Project Monitoring Center".
2. In the "Help for time management" tab, select "Enable the help for time management".
3. Validate.

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: "Project .. Time management .. Options for time management".

The other Control Centers

WebDev proposes other Control Centers, reserved to the WebDev developers:

- Quality Control Center: This center is used to enter the different problems that occurred on a project and to follow their process.
- Reusability Center: This center is used to centralize the resources (components, libraries, classes, ...) of your different projects in order to make them available to your development team.
- Document Management Center: This center is used to centralize the help systems and the documents generated for your projects.
- Topological Center: This center is used to draw a topological diagram of a site. It can be used to represent the servers, the client computers, the mobile devices, ...

LESSON 9.5. 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 a specific 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 costs, ...

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 "My_Library" project created in part 2 of this tutorial.

If this application was not previously created, a corrected version is available via the following option: "? .. Tutorial .. Site with data (Library)".



Note

Required configuration

To follow this lesson, you must use the environment in full mode. See “Required configuration”, page 351 for more details.

Application on a real example

Creating a business rule

- Open the "My_Library" 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. This image is named "IMG_NoName1".

  2. Display the description of the image. Via the "Catalog" button, select a GIF animation ("New" for example). Validate the selected image and its inclusion in the site directory. In the "GUI" tab of the control, indicate that the control is invisible.

  3. Enter the following code in the exit code with modification of the "Entry Date" control:

```
IF DateDifference(EDT_EntryDate, Today)<365 THEN
   IMG_NoName1..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 a specific 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 costs, ...

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.

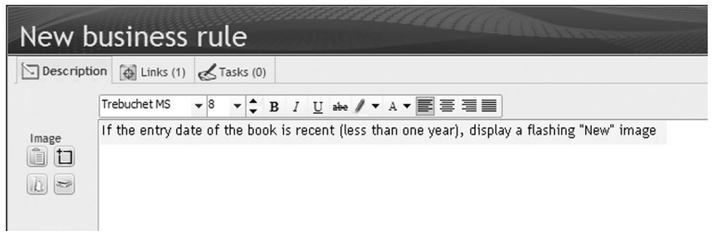
1. Display the pane of business rules (click 

).

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 on 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 en element that respect the conditions of the rule 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 "Automatically find the elements in: ".

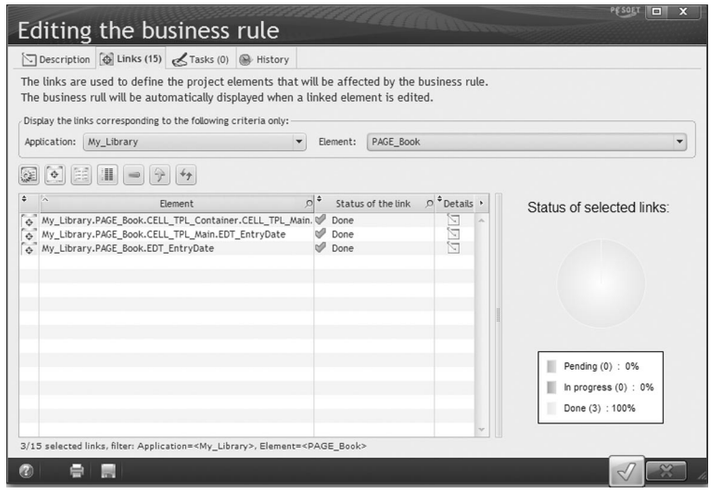
8. We are going to define the selection conditions of the business rule. Click "+" and specify the following information:



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. 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 "Performed" in the "Status of the link" combo box.



11. Validate the creation of the business rule.

12. Save your page.

LESSON 9.6. THE EXTERNAL COMPONENTS

This lesson will teach you the following concepts ...

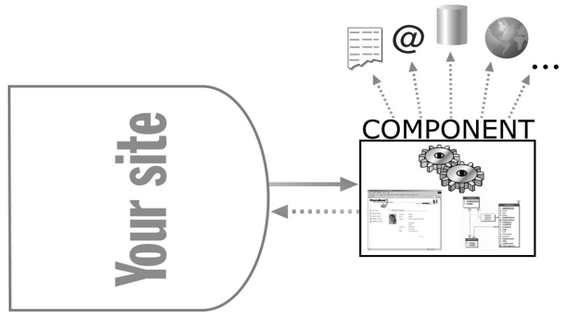
- What is an external component?
- Creating an external component, step by step.
- Distributing an external component.

Estimated time: 30 min

Overview

An external component is a brick that can be re-used in an unlimited number of projects.

An external component enables you to have an identical process with a unique source code to perform a given operation even if this process must be performed by several projects.



The projects that use an external component have access in the WinDev editor to the name of the objects, procedures or methods made visible by the creator of the component. The projects cannot view or modify the source code. The confidentiality of the source code is guaranteed.

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!

An external component can contain some code, some pages, an analysis, some 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. The external components
5. The internal components

Let's present some cases where components may be useful.

Teamwork

A developer creates and maintains a component that is made available to the other developers. No risk of modifications "made by mistake"!

Large projects

Components allow you to have several small projects as well as a central project in which the elements of the other projects are visible. Using components is a lot more convenient than using libraries (WDL files).

The databases accessed by several projects

When the same database is accessed by several projects, you often find inconsistencies in the database caused by modified or old source code. By grouping the operations for accessing the database (at least in write mode) in a component, a single source code must be checked and maintained: the risk of database inconsistency is 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 facilitates the common modifications.

Furthermore, the PRIVATE keyword enables you to make sure that your source code remains confidential at all the 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 instance) will be displayed!

The ability to distribute a feature or set of features

Components allow you to develop a feature or a set of features. Other WebDev developers will be able to include these features in their own projects. Developers who use a component can see the component elements that are made visible. However, the source code cannot be viewed or modified.

Your components can be distributed (free of charge or not)!

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.

In this last case, WebDev enables you to:

- include the elements coming from different products (WinDev and/or WinDev Mobile) in the same component.
- specify the corresponding WLanguage code for each runtime platform (for example, a window is displayed by ***Open*** in WinDev and a page is displayed by ***PageDisplay*** in WebDev).

Step by step

Step 1: Creating an external component

We are going to create an external component used to manage the logins and passwords for connecting to a site.

To avoid having to develop the code required for the component to operate, all the necessary elements have been grouped in a project named "Component_Login-Source". This project will be used to create our component. A new project will be created afterwards to use this component.



Notes

WebDev is supplied with the "Login" component. This component is a full version of the component used in this tutorial.

▶ 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 "External component (Exercise)". The project is loaded.

Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. External component (Exercise)".

This project contains all the elements required for the component to operate.
The component includes pages, classes, internal procedures and an analysis.

We are now going to create our component.



Notes

If you do not want to follow the different steps for creating the component, you have the ability to open the corrected projects via the following options:

- "? .. Tutorial .. External component (Answers)": Project that contains the generated component.
- "? .. Tutorial .. Using the external component (Answers)": Project that uses the external component.

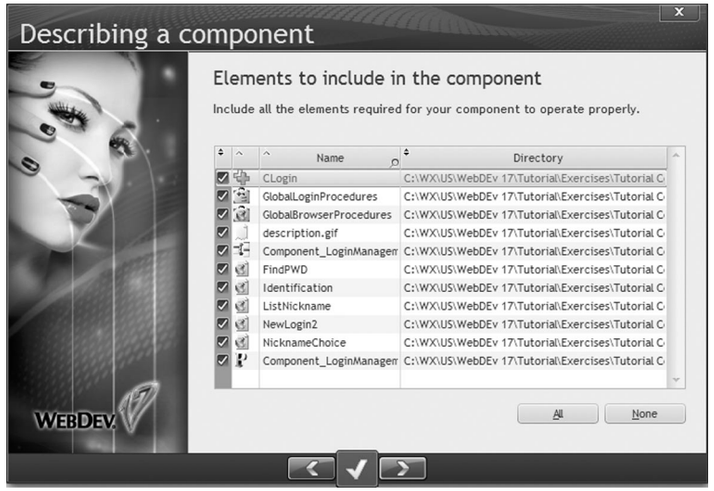
▶ To create a component, in the WebDev menu, select "Workshop .. External component .. Define a new component from this project". The wizard for creating a component starts.

▶ Identify your component:

1. Enter the name: "Component_Login".

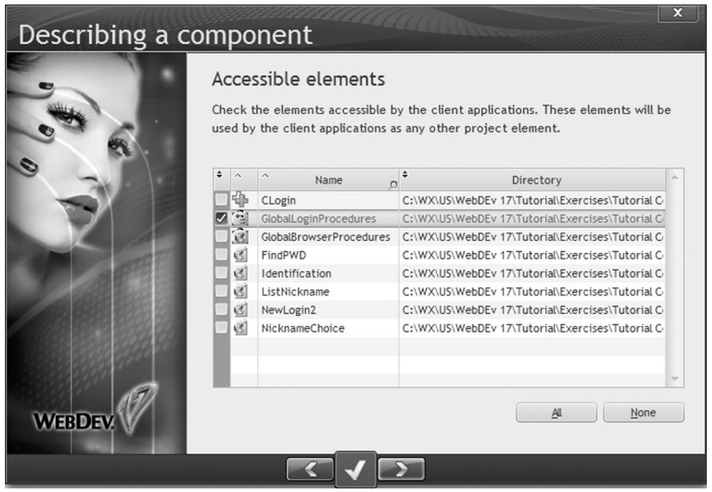
2. Enter the caption: "Login component created by the tutorial".

▶ Go to the next screen.
Select the elements that belong to the component. Check all the elements.



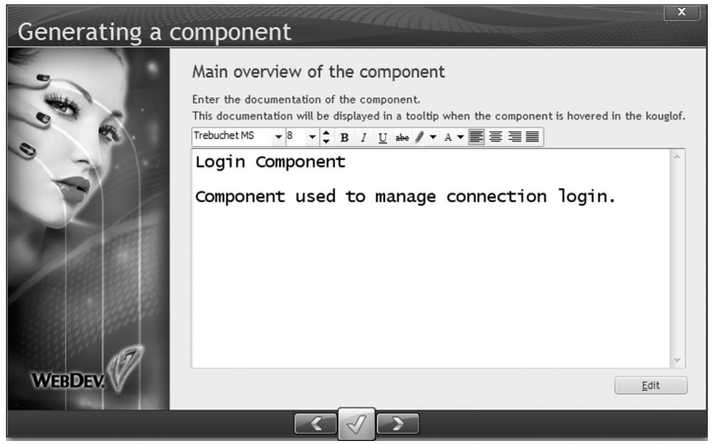
▶ Go to the next screen.

The wizard asks you to select the component elements that will be accessible from the client project. In our example, only the global procedures will be accessible:

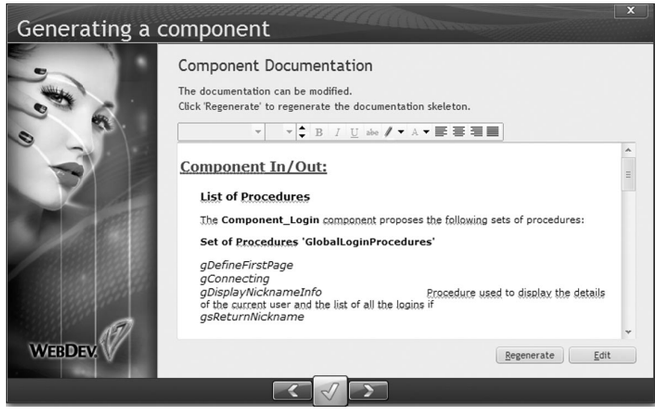


- Go to the next screen. WebDev creates the component then it proposes to generate the component.
- In the wizard for component generation, WebDev proposes to allow the component to be translated. This feature is interesting if your component must be used by some multilingual applications or sites. If "Allow the component to be translated" is checked, the specified component elements can be translated from the application that uses the component.
  This translation will be performed via WDMSG, independent module used to check out and check in the project messages to translate.
  In this example, don't check this option. Go to the next screen.
- Choose the languages of the component. Our example will contain the English language only. Go to the next screen.
- The wizard proposes to manage the different versions of the component. In our example, the component was just created.
  Keep the default options. Go to the next screen.
- In this window, enter the information regarding the component:
  - Owner,
  - Caption, Copyright, ...
    Go to the next screen.
- An image can be associated with your component. Users will be able to easily identify the component in the "Catalog of examples" pane.
- Go to the next screen. This screen is used to specify the general overview of the component.

You can specify:

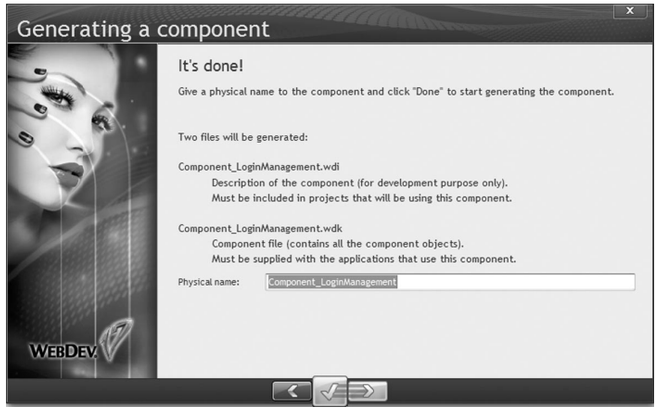


- Go to the next screen. The wizard will now automatically generate the documentation for the component. This documentation can be based on the comments inserted into your source code.



You have the ability to modify the generated documentation.
Don't do anything. Go to the next screen.

- You have the ability to include the component in the Reusability Center: the component can be easily re-used in other projects. In this example, the component will not be included in the Reusability Center.
  Go to the next screen.
- Now let's give a physical name to our component.



- Validate. Well done, you have just created your first ready-to-use component!
  The component is displayed in the "Wizards, Examples and Components" pane ("Components" button, "Custom components" item).

Step 2: Using the component

Once created, your component can be used in any other WebDev project. Let's see how this component can be re-used.

- Open the "Component_Login-Use.WWP" component ("? .. Tutorial .. Using an external component (Exercise)").



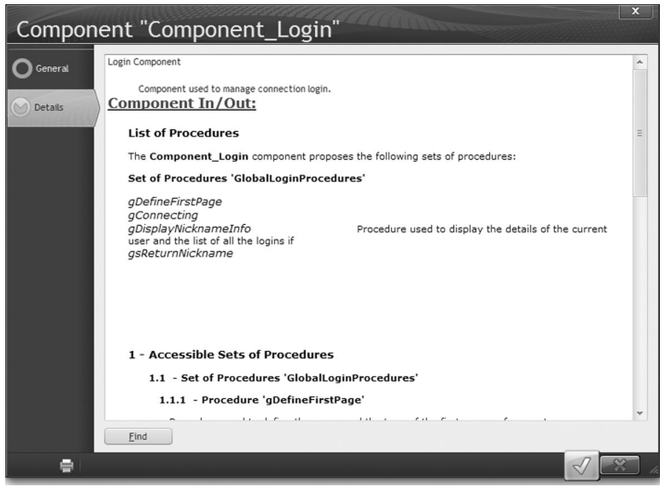
Notes

If you do not want to follow the different steps for creating the component, you have the ability to open the corrected projects via the following options:

- "? .. Tutorial .. External component (Answers)": Project that contains the generated component.
- "? .. Tutorial .. Using the external component (Answers)": Project that uses the external component.

- In the project editor, select "Workshop .. External component .. Import a component into the project .. from a file".
- In the directory of the "Login component-Source" project, select the EXE sub-directory and the "Component_Login.WDI" file.

- Click "Open", the description of our component is displayed:



- Validate.
- We are now going to modify the code of the example to take the component into account:
  - In the "ConnectionPage" page, display the code of the connection button. Remove the comments to call the "gConnecting" function of the component.
  - In the "ConnectionPage" page, display the code of the page ("Code .. Page code"). In the "Declaration of global variables" process, delete the comments found before the "gDefineFirstPage" function.
- Run the test of the project.

That's it! Child's play isn't it?

You now know how to create a component and how to re-use it in your WebDev projects. You also have the ability to manage the setup procedures of your components, to distribute them separately from your sites for instance.

Distributing a component

Two methods can be used to distribute a component:

1. Provide the necessary files "manually", this is a "standard" distribution.
2. Create a more "professional" distribution via the setup editor of WebDev.

Standard distribution

In this case, you must supply all the files required for your component to operate. These files are created when generating the component (WDI, WDK and WDO files, images, other documents, ...). These files will be manually copied from their source directory to the destination directory. The WebDev projects that will be using this component will find the dependent files in this destination directory.

List of files that must be supplied for a standard distribution:

- the files automatically generated by WebDev (WDK, WDI, ...).
- the dependency files.
- the WDO file must be supplied if the component uses dependency files. This file contains the references to the different external files used in the component.

Professional distribution

The distribution of components via a setup procedure consists in supplying a setup program to the users of the WebDev component. This program installs all the files required for using the component in the directory specified by the user.

This setup mode is used to automatically manage:

- the WDO file and the setup of additional files used by the component.
- the automatic setup of the necessary tools (MDAC, ODBC driver for HyperFileSQL, ...).
- the automatic update of the data files used by the component (if necessary).
- the uninstall program of the component.

▶ Go back to the "Component_LoginManagement-Source" project. To create the setup, select "Workshop .. External component .. Create the setup procedure of a component".
The wizard for creating the component setup starts.

We won't go into details about the different setup modes of a component. Follow the instructions given by the wizard. See the online help (keyword: "Component, Distributing a component") for more details.

Some components supplied with WebDev 17:

- Managing the logins
- PayBox
- Gantt chart
- Managing the basket
- Search engine
- Request for documentation

LESSON 9.7. 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 configurations of WebDev project.

The project configurations are used to create several different "targets" from the same project. You have the ability to choose the requested configuration at any time 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 project named "Site with data (Library)". The project is loaded.
Tip: if the home window is not displayed, you also have the ability to select "? .. Tutorial .. Site with data (Library)".
3. In the "Project Explorer" pane, select "Configurations (My_Library)".
4. Select "New configuration" from the popup menu. The wizard for creating a project configuration starts.
5. In the general information of the project configuration, specify the name, the description and the type of generation (test component, test, component for our example).



Go to the next screen.
6. Specify the products in which the component will be used (WebDev).
Go to the next screen.

7. Specify the elements that will be included in the configuration. In our example, two pages will be included. Go to the next screen.
8. Validate the creation of the configuration.
9. The wizard for generating the object of the configuration (the component in our case) starts automatically.
10. Follow the wizard to create the component.
11. At the end of the wizard, the "Test components" configuration is automatically selected in the "Explorer" pane. Select the "My_Library" configuration then "Open this configuration" from the popup menu.

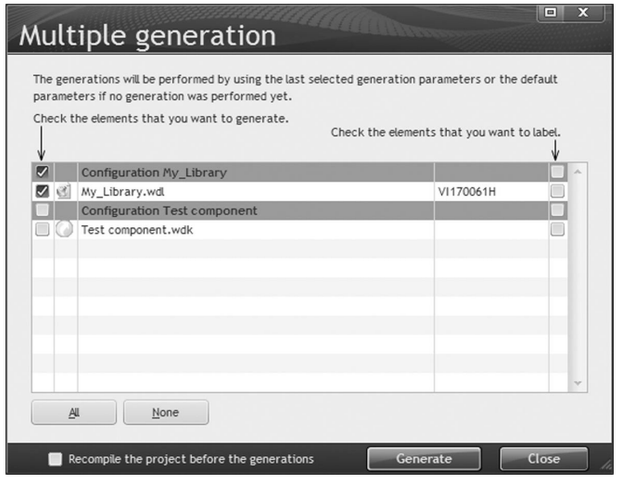
Managing the project configurations

"Project .. Project configurations .. Configuration management" is used to define all the options for project configurations. You can:

- 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 that will be performed before and after the generation (copy the element to a specific directory for example).

Multiple generation

We now 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, select "Workshop .. Multiple generation".
All you have to do is select the elements to generate. This generation will be performed in a single operation.



LESSON 9.8. 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 the full site (My Library 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 the code lines during the debug operation.

The debugger can be used in 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

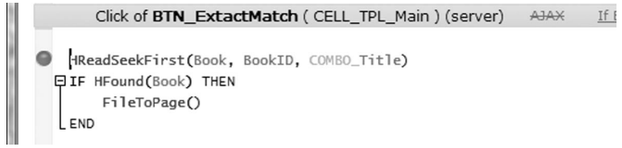
Starting the debugger

To run the test of a page, click the GO icon (or press [F9] or select "Code .. Run the page test").

To debug a page, all you have to do is start the debugger. If you have identified the code that triggers the error in your page, the debugger can be started 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 "My_Library" project. If the project was not developed, it can be opened from the following option: "? .. Tutorial .. Site with data (Library)".
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's a breakpoint.



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 ( ).
6. Select an element in the combo box.
7. Click the "Exact-match search" button.

8. The code editor is displayed in debug mode.

We are now going to present the information displayed in the debugger.

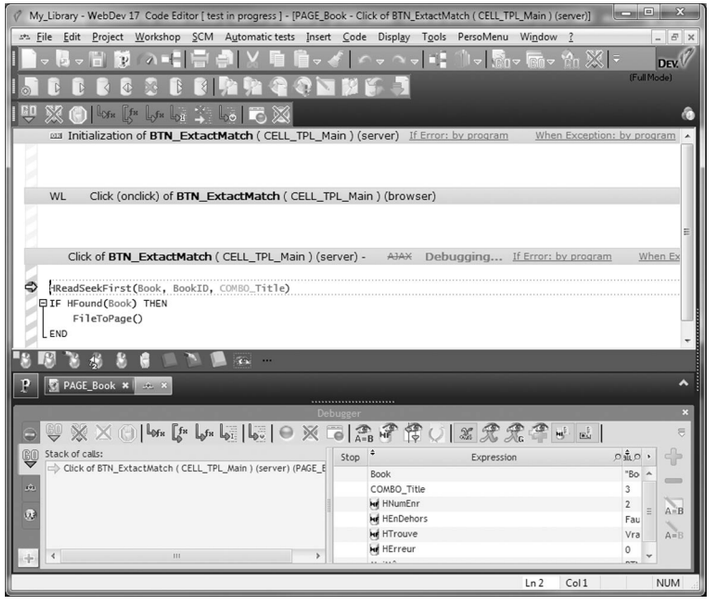
Viewing the information in the debugger

The code editor is a little bit different in debug mode. A specific toolbar is displayed and the "Debugger" pane is automatically displayed.



The code editor will allow you to follow the execution of the code line by line.

The "Debugger" pane will allow you to follow the values taken by the different variables handled by the code.



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 over the "END" line.
4. Press [F6]. All the lines up to the cursor are run.

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 were 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

The WLanguage function named ***Trace*** can also be used to debug a project. This function is used to display the requested information in a "trace window". You can for instance display the value of a variable, a message to find out the way taken by the application, ...

When running in GO mode (for the project, page, and so on), the trace window is automatically displayed when ***Trace*** is reached. This window disappears when the test is over. However, the information displayed in the trace window can be retrieved via the "Trace of debugger" pane.

This pane displays the various information displayed during the test with "Trace". A click performed on one of the lines found in the trace pane enables you to automatically access the corresponding code line.



Notes

An error occurs when running 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 enables you to display the detailed information.

Debugging a project

You also have the ability to debug the entire project. In this case, the parameters specified in "Project .. Test mode .. Configure the test mode" are automatically taken into account.

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 instance.
- 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 for more details (keyword:"Debugger") for more details.

LESSON 9.9. 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 instance).

Its 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 WebDev editor** ("Project .. Performance profiler .. 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 your site.
  The performance profiler displays the result of the analysis. This result is saved as a WPF file.
- **from one of your processes in WLanguage**, via the following functions:

ProfilerStart	Starts "collecting data" for the performance profiler.
ProfilerEnd	Stops "collecting data" for the performance profiler.

In this case, only the code found between ***ProfilerStart*** and ***ProfilerEnd*** is analyzed. The corresponding WPF file can be opened in the performance profiler ("Project .. Performance profiler .. Performance report .. 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. Select "Project .. Performance profiler .. Performance report .. Open a report of performance".
2. Specify the path and the name of the statistical file.

The list of the last statistical files opened can be found in "Project .. Performance profiler .. Performance report".

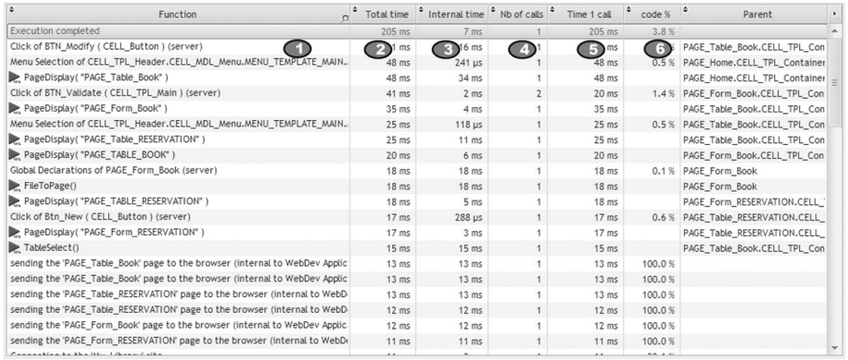
Optimizing a process with the performance profiler

Reading the result of the performance profiler

The performance profiler presents the result of the analysis in two tabs:

- the "Summary" tab presents the 20 longest processes.
- the "Details" tab presents all the processes started while running the test of the application (from the slowest one to the fastest one).

The following information is displayed for each process:



The elements displayed are as follows:

1. Function: Function, process or procedure run.
2. Total Time: Execution time of the function.
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 processing the function or the procedure (developer code that could 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 be optimized.
- the percentage of time spent processing the function or procedure. The higher this percentage is, the greater the number of processes that can be optimized in the code.

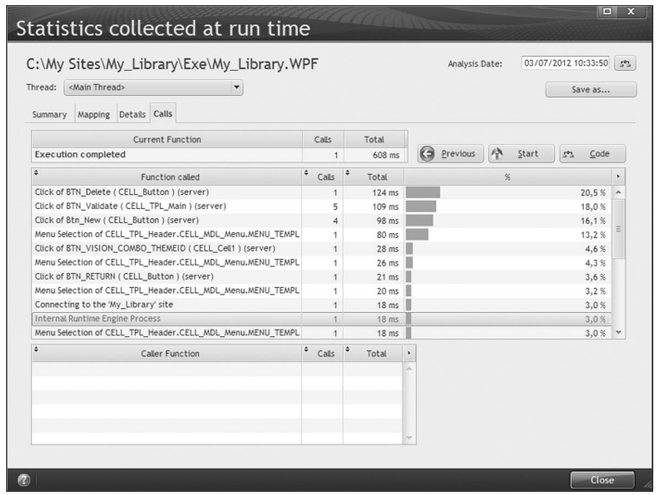
Optimizing a process

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 that call the selected process.

The list of functions called enables you 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 that contain WLanguage code (named "Developer code") can be optimized.

A double click performed on one of the functions found in this list ("Click Validate" for instance) enables 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 for the same function or procedure (excluding 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 enables you to go back to the calling function.
- The "Start" button enables you to go back to the process at the beginning of the application.

LESSON 9.10. 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 some existing WebDev elements into your project. All types of WebDev elements can be imported:

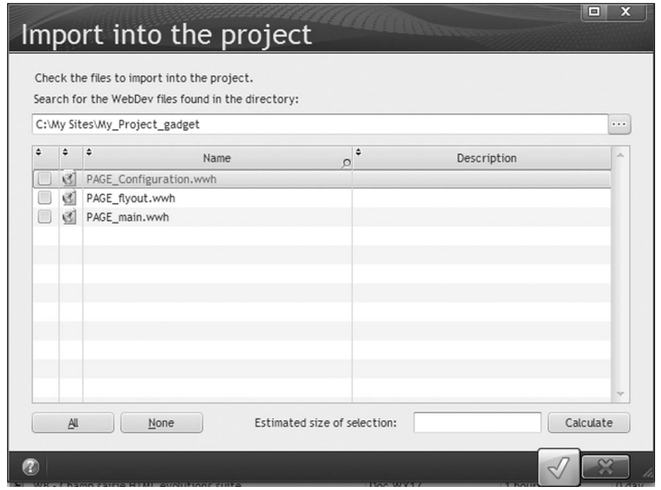
- pages, reports
- classes, components
- procedures, ...

▶ To import existing elements into the current project:

1. Select "File .. Import .. 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 part of the project.

Notes

The "Calculate" button (found in the import window) is used to calculate the size of the selected elements along with all 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.



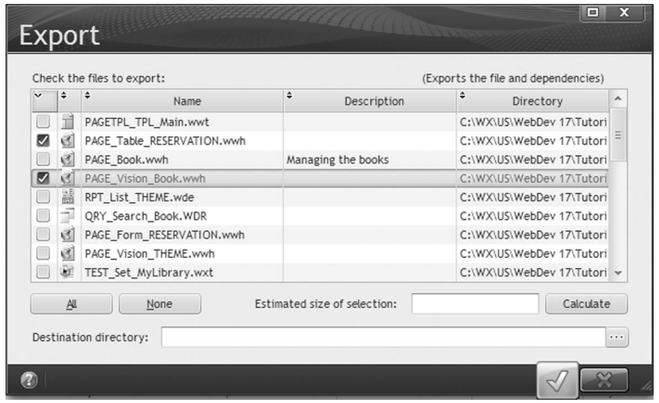
Caution !

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 all their dependencies by email for instance.

▶ To export elements from your project:

1. Click "File .. Export .. To a directory...".

In the window that opens, 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 all 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. Select "File .. Import .. 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 converted into WebDev project.

To import some WinDev elements into a WebDev project:

1. Open the WebDev project where the element must be imported.

2. Select "File .. Import .. WinDev or WinDev Mobile elements". The import wizard starts. Select the WinDev project containing the elements to import or directly select the WinDev elements to import. Go to the next screen.

3. Specify the elements that will be shared among the WinDev project and the WebDev project. 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

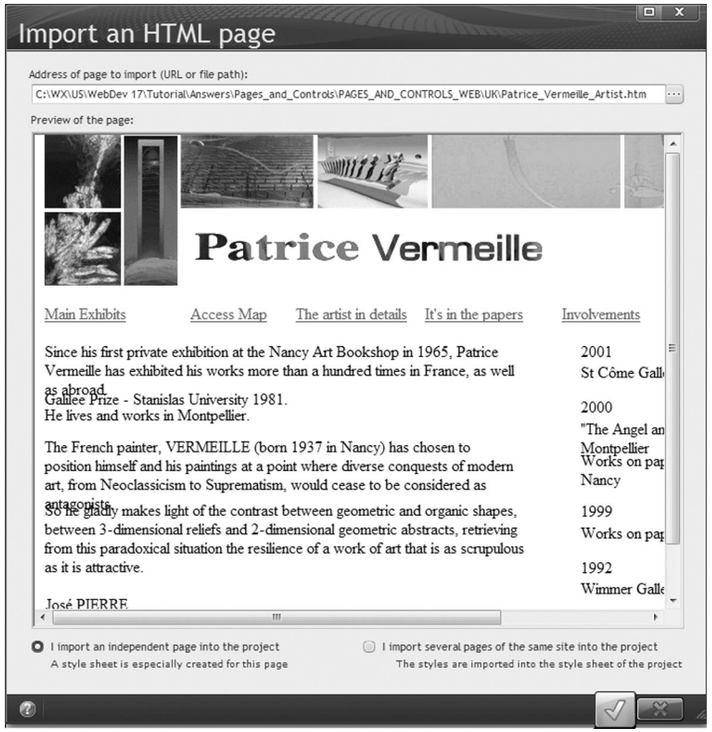
The feature for importing HTML pages is a great help given to the developers of WebDev sites. This feature is not designed to "suck up" the Internet sites.

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. Select "File .. Import .. An HTML page...".
The wizard for importing the 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 instance). 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 9.11. 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.



Note

Required configuration
To use the UML diagrams proposed by WebDev, the UML feature must be enabled in your configuration. We advise you to use the environment in full mode. See “Required configuration”, page 351 for more details.

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 represents 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 usual layers of an application: GUI, processes and data.

An application will include 3 independent layers:

- a presentation tier
- a process tier
- a data access tier.

Specific functions (APIs in standard languages, advanced WLanguage functions) allow these 3 tiers to communicate between themselves.

The reason for separating the layers is to facilitate 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 process tier 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).

In the data model editor and in the query editor, you create the different data files and the necessary queries.

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, select "Analysis .. 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 ("File .. New .. Architecture .. UML .. Class diagram .. Build the class diagram corresponding to the database and to the queries (3-tier architecture)").

Choose the type of code to generate (procedural or object-oriented).

3. The classes and structures are generated in WLanguage.

Close the UML model and select "Workshop .. UML modeling .. 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 classes and the diagram are synchronized in real-time.

You can also modify the analysis and 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 class members 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 9.12. FLEXIBLE MODELING

This lesson will teach you the following concepts ...

- **Principle**
- **Operation**



Estimated time: 20 min

Principle

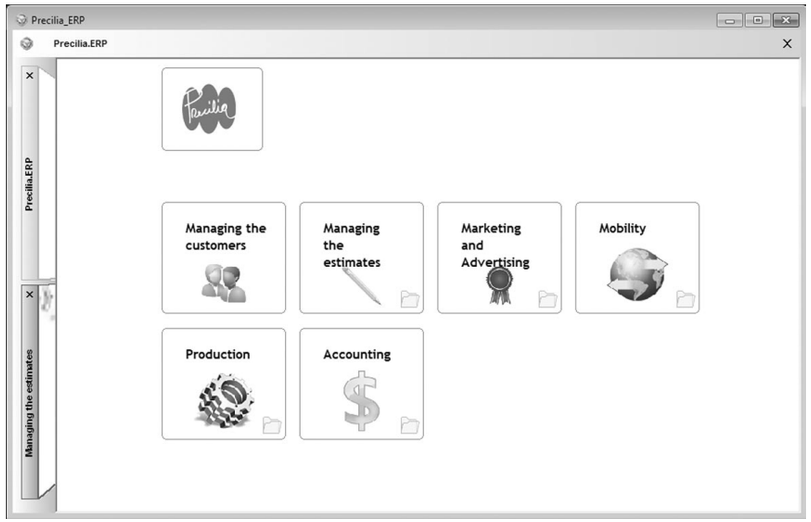
The modeling editor is used to represent all the organizations (existing ones or forthcoming ones). The modeling editor enables you to represent the organization of email management in your company, the organization of contacts with the customers, ...

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.

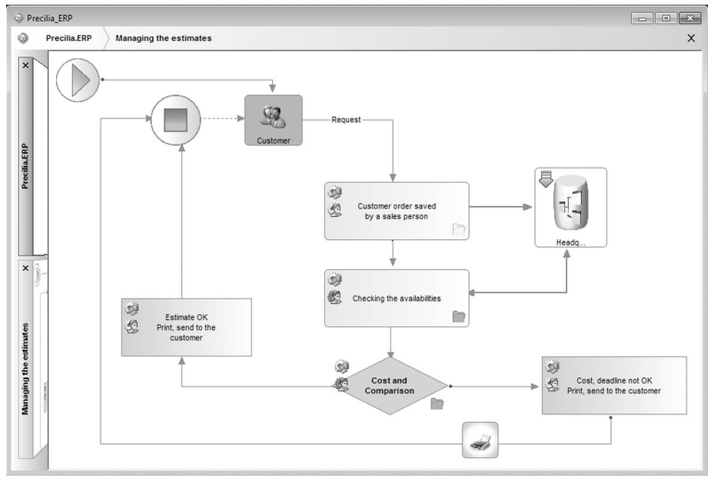
Operations

- 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, you also have the ability to select "? .. Tutorial .. Flexible modeling".
- Open the file named "Precilia_ERP.wem" ("File .. Open" or 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 that a sub-modeling is found. 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, and so on).
- 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 indicate:

- 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 may come from the contributors described in the Project Monitoring Center.

See the online help for more details.

LESSON 9.13. 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, depending on 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 "Multilingual_Book" project. This project, available in French, will be translated in English and in Arabic.

Choosing the project languages

▶ 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, you also have the ability to select "? .. Tutorial .. Multilingual site".

The associated analysis is "Book". This information can be checked in the project description ("Project .. Project description", "Analysis" tab).

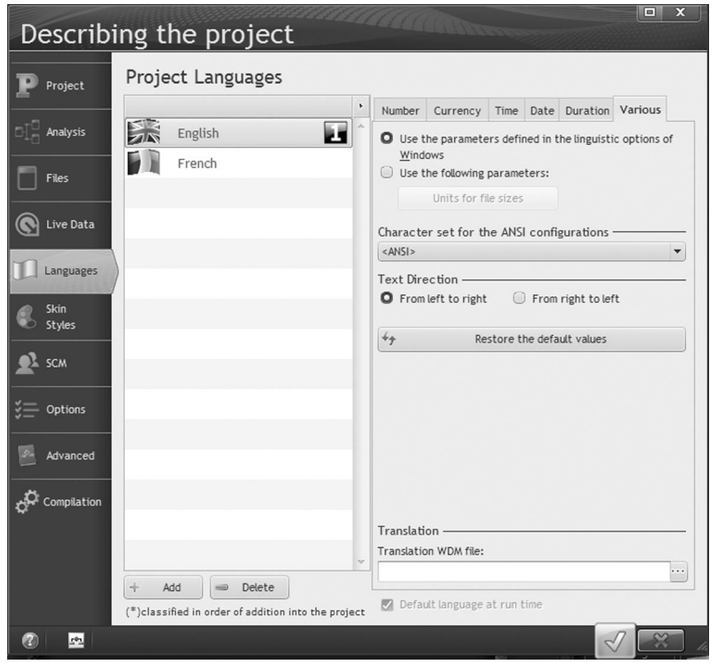
The first operation consists in choosing the project languages.

▶ Display the project description ("Project .. Project description").

▶ Click the "Languages" tab. Our application will support French and English. Check whether these 2 languages are displayed.

▶ This tab is also used to configure the linguistic options regarding the numbers, the currencies, the dates, ... for the selected language. Let's see an example:

- Click the "English" language.
- Click "Date".
- The linguistic options of Windows are used by default. Select "Use the following parameters": you now have the ability to define the date format used as well as the translation for the days and for the months.
- Keep "Use the parameters defined in the linguistic options of Windows".



▶ Validate.

Notes

In the linguistic options, you have the ability to choose the text direction of the language ("Various" tab, "Text direction" option). This enables 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 ("Project .. Load the analysis").

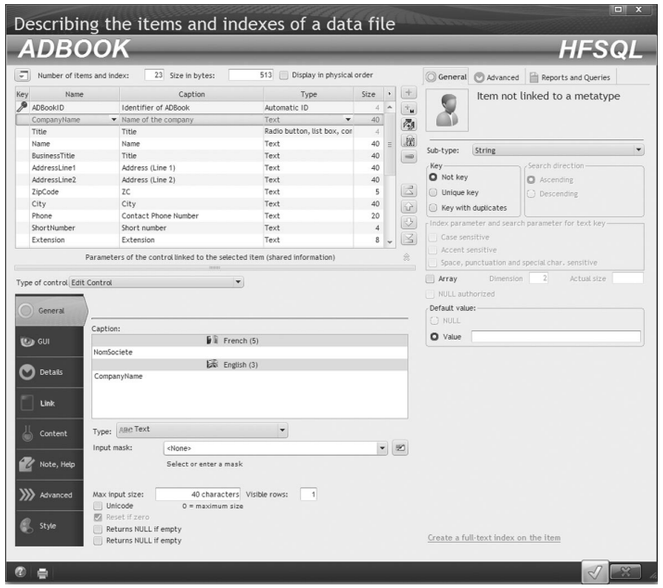
2. In the analysis description ("Analysis .. Analysis description"), select the "International" tab.

3. Specify the languages supported by the analysis and validate.



To enter the shared information in several languages:

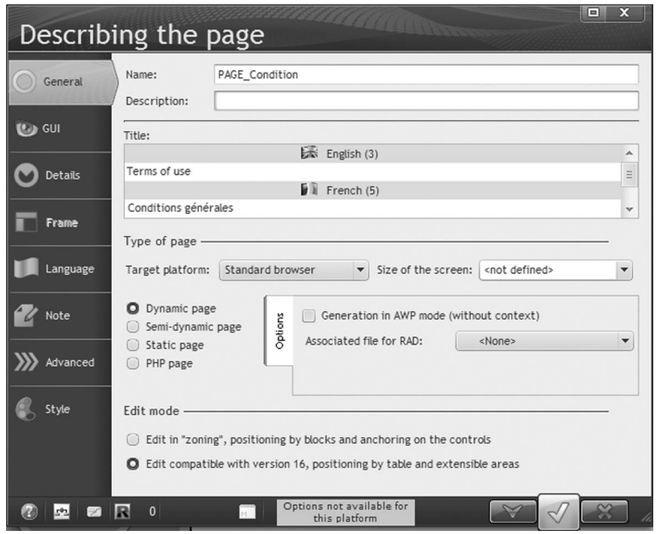
1. Display the description of the items found in a data file (select the data file named "ADBook" then "Structure of files .. Items...").
2. For each item, display the parameters of the control linked to the selected item (shared information). To do so, click the double arrow at the bottom of the screen.



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_Condition.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.

- Display the description of the "Back" link. The "General" tab also presents the translation of the link caption.
- 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 some 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 translation performed in the different editors.
- a translation performed via an external tool (WDMSG and WDTRAD)

Direct input of the 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 directly performed in the editor.

To translate the elements of a page (including the menu) :

1. Open the page containing the popup menu.
2. Select "Display .. 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 caption 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. Select "Tools .. Options .. 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 have been defined, click  to use the software defined for the translation.

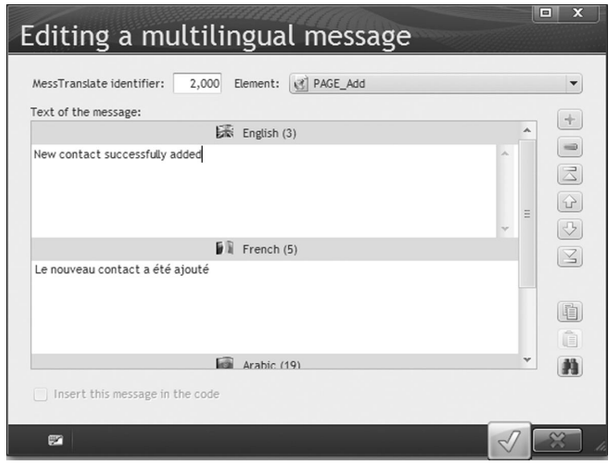
Special cases

Direct translation of the programming messages

All the messages found in your program can also be entered in several languages.

```
Info("New contact successfully added") // Contact successfully added
```

To translate this type of message, select "Code .. Multilingual messages .. Translate the messages". The following window is displayed:



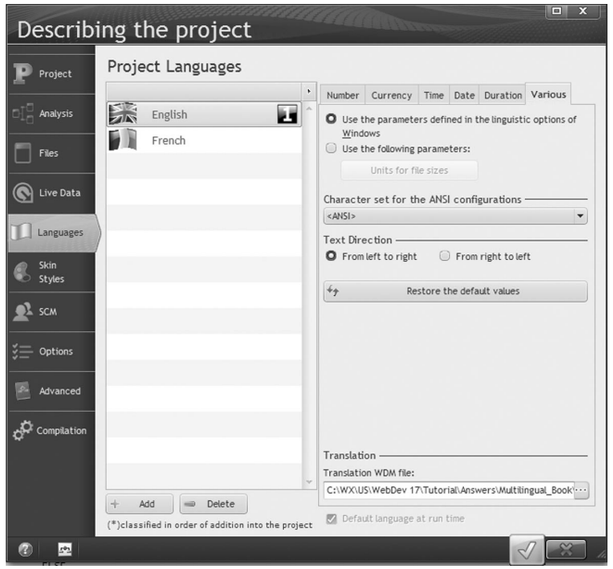
This window enables you to translate all the messages found in your program into all the project languages.
A number is assigned to each message.
When a translation is entered for the message, the icon  is displayed in the code editor.

```
Info("New contact successfully added") //New contact successfully added
```

Translating the result of the 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 can use the function named ***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 WDMSG and WDTRAD

A tool named **WDMSG** (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 a text format, which can be configured to be used by most translation software.

WDMSG is also supplied with a tool for computer-aided translation, WDTRAD. WDTRAD is used to enter all the translations for the multilingual information found in a project.

Contact our Sales Department for more details about **WDMSG and WDTRAD**.

Programming the change of language

By default, the project is run in the first language defined for the project ("Project .. Project description", "Languages" tab).

The function named ***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 by clicking a button. As soon as a language is selected by the Web user, the site is loaded in the selected language.

- In the "PAGE_Notebook" page, a combo box ("COMBO_LANGUAGE") is used to change the language. The code associated with the combo box is as follows:

```
SWITCH COMBO_LANGUAGE
   CASE 1 //"French" is selected
      Nation(nationFrench)
   CASE 2 //"English" is selected
      Nation(nationEnglish)
END

PageUse(CurrentPage())
```

The function named ***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.

- Running the test of the site.
  In the "Language" combo box, select a language to modify the current language.

Managing the Unicode and the 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 HyperFileSQL 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 some 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
DEVELOP 10 TIMES FASTER
PC SOFT

LESSON 10.1. DEPLOYING A DYNAMIC SITE

This lesson will teach you the following concepts ...

- Installing WebDev Application Server - 10 connections.
- Creating the setup.
- Running the 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 be made 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-ROM, ...). 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. Deploying the site via the test hosting service of PC SOFT.

When installing WebDev, you have the ability to install the test version of WebDev Deployment. This enables 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 application server for test.

The following elements must be installed on the computer:

- a Web server (IIS, 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
- on the setup DVD of WebDev: start "Menu.exe" and choose "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 17" by default). Validate the creation of the directory.
4. Choose a full setup. Go to the next plane.
5. Validate the next plane without modifying it.
6. The active Web server found on the setup computer is selected by default. Go to the next plane.
7. Keep "Host the sites in earlier versions" and go to the next screen.
8. Validate the setup.
9. Select the virtual Web servers on which the WebDev application server must be installed. The default Web site is sufficient. Validate.
10. Keep the selected options and check "Start the administrator of WebDev 17". Validate.
11. 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 to the server.

Configuring the WebDev account for deployment

In the WebDev administrator :

1. Select the "Management" 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 advise you to create some 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 sites of ADMIN: 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 pages of the site. 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 pages of the site. 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 previously created. 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 17\Account" by default.

To modify the rights granted to this user, we are going to modify the registry. This modification is performed via "regedit.exe" (use regedt32.exe if your operating system is earlier than Windows XP).

1. Start the registry editor.
2. Find the following key: "HKEY_LOCAL_MACHINE\SOFTWARE\PC SOFT\WEBDEV\17.0\Applications".
3. Select "Authorizations" from the popup menu of the key (with regedt32, select "Authorization" in the "Security" menu).
4. Add the "Test" user and give him total control.
5. Validate.

Deploying your dynamic site

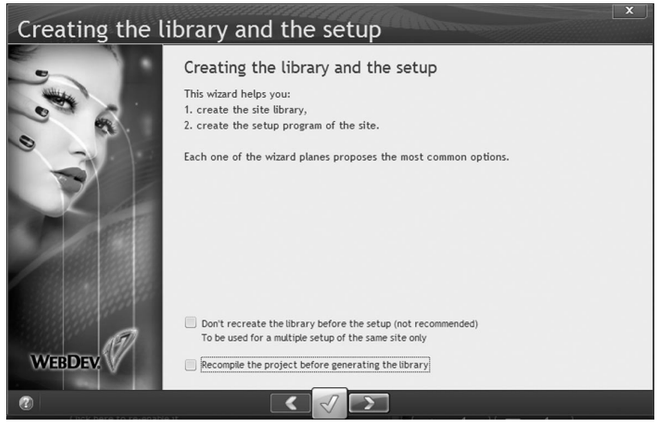
We are going to deploy the "My_Library" project that was created in part 2 of this Tutorial. If this project was not previously created, a corrected version is available.

▶ To open this project in WebDev:
1. Select "File .. Open a project".
2. Select the "My_Library" project.

If this application was not previously created, select "? .. Tutorial .. Site with data (Library)".

Preparing the setup

A setup wizard is supplied with WebDev. This wizard enables you to easily install your site on the hosting company (Internet or Extranet site for example) or on one of your servers dedicated to WebDev (Intranet site for example). To access it, select "Workshop .. Deploy the site .. Deploy the site".



▶ Display the next screen.
Before you continue, 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, description of the 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.



Several setup modes are available:

- Setup on a remote WebDev application server (by FTP),
- Setup on a PC SOFT test server,
- Preparing a deployment package,
- Setup by physical media (for a manual setup),
- Stand-alone setup by physical media containing the Web server and the WebDev application server.

▶ In our case, we will 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 parameters of the server, the following information must be supplied by your hosting company. We are going to enter the information corresponding to the setup that was previously performed:

- 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 host 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 host 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 the files that will be transmitted to the server.

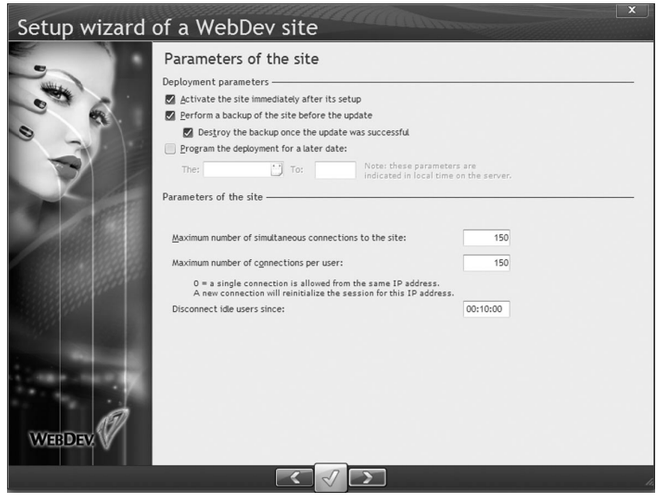
.



Go to the next screen.

5. The wizard proposes to include the automatic modification of the data files in the setup. You also have the ability to configure the elements required to use a HyperFileSQL 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, the site is enabled as soon as the setup is over. 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. 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.

8. Display the next screen and validate the setup. The setup wizard transfers the files.

During the file transfer, the wizard compresses and encrypts the data that was transferred. Your data is transferred with a high-security level.
At the end of setup, a link allows you to start the site immediately.

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 made available in less than two minutes.

LESSON 10.2. MANAGEMENT REMOTE

This lesson will teach you the following concepts ...

- **Managing a dynamic WebDev site remotely.**
- **Using WDAdminWeb170 in test deployment mode.**



Estimated time: 10 min

Overview

The WebDev application server proposes several tools for managing the dynamic sites, among which **WDAdminWeb170**. This tool is in fact a WebDev site included in the engine of WebDev. It is used to manage a dynamic WebDev site remotely.

You will be able to use this tool to update the parameters of your site when your dynamic site is deployed on the host system (if the remote management was allowed by the hosting company).



Notes

To deploy a dynamic WebDev site, the WebDev application server must be installed on the servers.

▶ To start WDAdminWeb170:

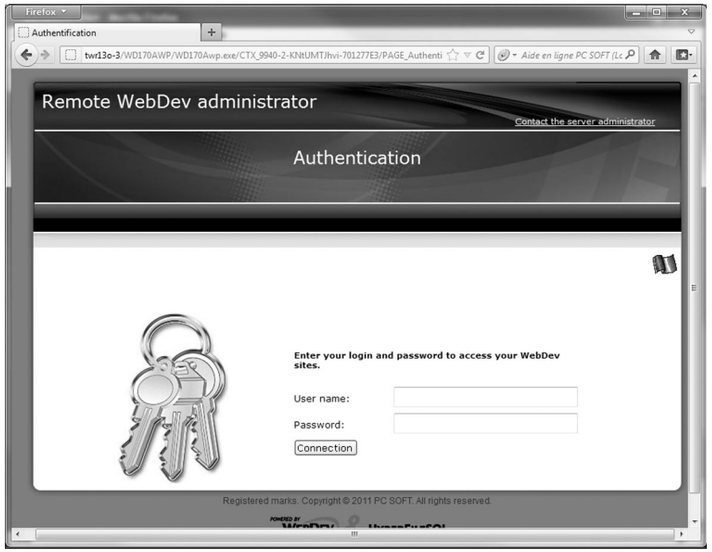
1. Start the WebDev administrator (WD170ADMIN) on the deployment computer.
2. Open your favorite browser (Internet Explorer for example) on your computer.
3. Enter the following URL in the address bar of the browser (while respecting the case) :

```
http://server/WD170AWP/WD170Awp.exe/CONNECT/WDAdminWeb170
```

or

```
http://server/WDAdminWeb170
```

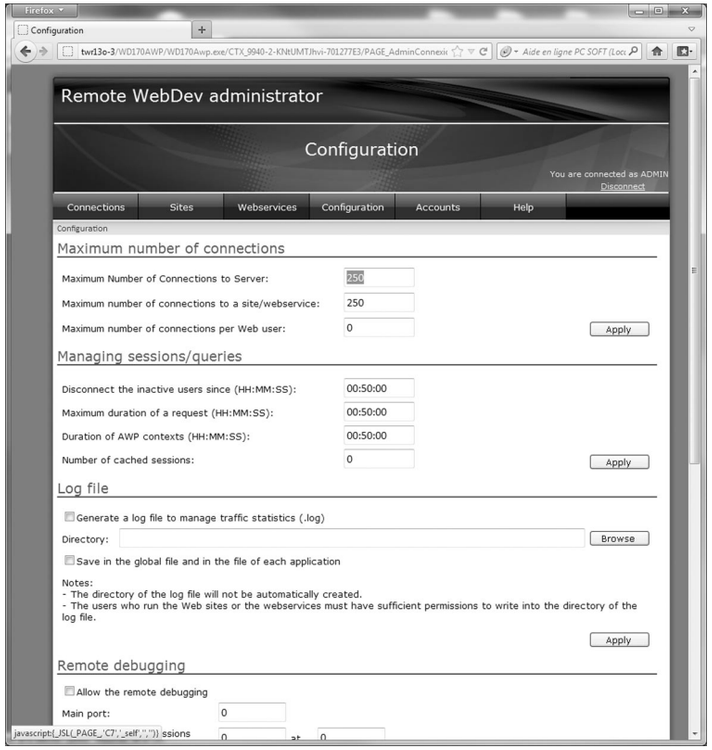
where "computer" is the name of the deployment server for test (your computer for example).



4. Enter the login and password supplied by the site administrator.
5. Click "Connection". The remote administrator is now available.

Using WDAdminWeb170

WDAdminWeb170 includes most of the options available in the WebDev administrator.



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 WDStatistic.

Estimated time: 10 min

Introduction

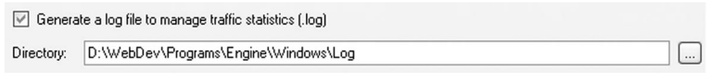
The traffic statistics are very important for a business site. They can be used to analyze the behavior of the Web users and to improve the navigation in the site.
WebDev includes an advanced tool for calculating statistics about your dynamic sites.

Configuring the server

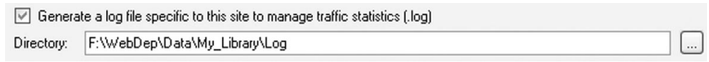
On your deployment server (if your hosting company proposes to manage the statistics), WDStatistic is used to analyze the different connection logs created by the WebDev administrator.

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:



- 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":



Note: you also have the ability to configure a site to create a specific log file from the remote administrator.

▶ We are now going to create a log file for one of the dynamic sites found in this course (the "My_Library" project for example).
1. Start the WebDev administrator ("Tools .. WDAdmin - WebDev administrator" from WebDev development version).
2. In the "Sites" tab, select the project for which the log file must be generated ("My_Library" 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.
6. In the "Advanced" tab, start the site from the "Test page".
7. Handle the different pages of the site.

Using WDStatistic

Starting WDStatistic

On your computer (development computer for example), WDStatistic is available in the "\Programs" directory of the setup directory of WebDev ("WDStatistic.exe"). To start WDStatistic, select "Tools .. WDStatistic - Statistics of WebDev sites" from WebDev.

▶ Start WDStatistic.

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 WDStatistic.



Notes

In our example, a manual import will be performed (the log files are found on the development computer).
To systematically use WDStatistic, we advise you to perform an automatic import. Therefore, the next logs will be automatically retrieved by WDStatistic.
See the online help about WDStatistic 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 (previously specified in the WebDev administrator).
4. Validate.

▶ To import the log files:
1. Click the "Import" button found in the main screen of WDStatistic. 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 the import, close the import window ("Close" button).
3. Validate the window for importing logs.
4. The statistical dashboard si displayed in the main window of WDStatistic. You can 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 be designed for placing orders, making bookings, buying products, requesting information about the company, ...

All the details entered by the Web user are saved in a HyperFileSQL 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 by 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 enables 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 ...

The WLanguage (which is fully compatible between WinDev and WebDev) enables 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 by 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 buying (WebDev),
- monitoring of industrial equipment (WinDev), sales report via Extranet or Intranet (WebDev),
- banking application (WinDev), account and investment lookup (WebDev),
- ...



Notes

If you do not own WinDev, contact the sales department of PC SOFT to get the free technical brochure that presents WinDev 17 or connect to the Internet site of PC SOFT: http://www.windev.com!

CONCLUSION

The tutorial is over now!

This course 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 only address one 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**!