



DEVELOP 10 TIMES FASTER



PCSOFT

Remember to visit the download section of [www.windev.com](http://www.windev.com) on a regular basis to check whether upgraded versions are available.

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

This documentation is not contractually binding. Some modifications may have been made to the software since this guide was published. Please check the **online help**

All product names or other trademarks mentioned in this publication are registered trademarks of their respective owners.  
© PC SOFT 2012: This publication may not be reproduced in part or in whole in any form without the express permission of PC SOFT.

## In which order should these guides be read?

WebDev is a powerful web development tool, which comes with all the tools needed to design and build web sites.

**To quickly and efficiently learn how to use WebDev, we recommend that you proceed in the following order:**

- 1** Reading the "Concepts".  
This guide presents the main concepts required to create a high-powered WebDev site. Some concepts are followed by a "Practical" section that presents some features of the editor.
- 2** "Tutorial" (guide + exercises)  
The tutorial provides a first "hands-on" approach to WebDev. It enables you to familiarize yourself with the main editors of WebDev.
- 3** Test of the examples  
Run the test of the examples supplied with WebDev in the fields you are interested in (e-business, directory, scheduling, ...)

The "WLanguage" book presents the programming in WLanguage. For each programming theme, you will find a description of the associated feature and the list of corresponding WLanguage functions.

The online help, accessible from <http://doc.windev.com> or installed with WebDev, allows you to easily find the syntax of a WLanguage function, get help about an interface ...

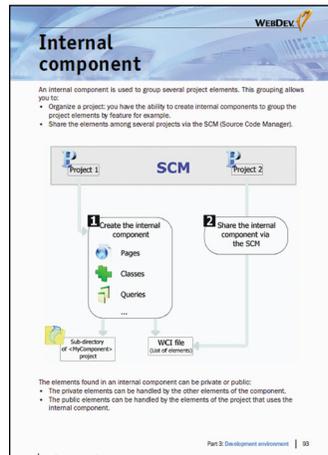
Note: If there is a difference between the guide and the online help, follow the instructions given in the online help.

We hope you enjoy getting started with WEBDEV.

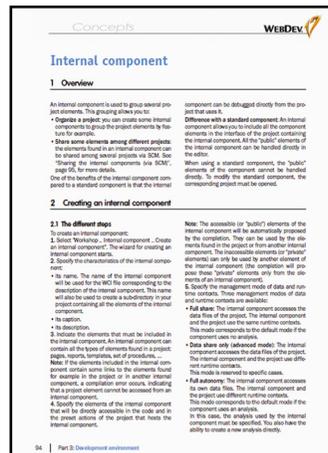
# Organization of the guide

This guide presents the main concepts required to create a high-powered WebDev site. Some concepts are followed by a "Practical" section that presents some features of the editor.

Therefore, two types of pages are found in this guide:



Concept page



Implementation page

# Summary

## PART 1: INTERNET: MAIN CONCEPTS

- What is a Web site? ..... 11
- Static, semi-dynamic or dynamic site? ..... 12
- The different types of dynamic sites..... 13
- Hardware and software requirements..... 14
- The different types of pages..... 15
- Dynamic pages and AWP pages..... 19
- AJAX technology ..... 21
- If you are already familiar with WinDev ..... 24

## PART 2: DEVELOPING A SITE

- Project and Analysis..... 27
- Development cycle of a site ..... 32
- I want to create a page in the editor ..... 34
- The following files are automatically created: ..... 35
- Editing a page: zoning mode ..... 36
- Internal page ..... 37
- Page templates ..... 38
- The different types of standard controls ..... 45
- Repeating a group of controls: the loopers..... 55
- The two types of code ..... 56
- WLanguage: a powerful language ..... 59
- Sequence for running the code of buttons/links..... 60
- Upload: Sending files to the server ..... 62
- Control templates..... 63
- Reports ..... 64
- Report templates ..... 65
- Different print modes ..... 66
- Displaying and sequencing pages ..... 67
- Performing a process on several successive pages..... 73
- Style sheets: for simplifying the layout ..... 74
- Centered or adapted site? ..... 75
- Referencing a site ..... 76

## PART 3: DEVELOPMENT ENVIRONMENT

- WebDev editors ..... 83
- Project dashboard..... 87
- WebDev, WinDev, WinDev Mobile: 100% compatible ..... 88
- Project configuration..... 89
- Multiple generation..... 90
- Source Code Manager (SCM) ..... 91
- Internal component ..... 98

External component.....101  
 Generation modes .....109  
 The Control Centers .....113  
 Managing the requirements.....114  
 Project Monitoring Center.....115  
 Managing the business rules.....116

**PART 4: DEVELOPING WEB SITES: ADVANCED CONCEPTS**

RAD RID .....119  
 The UML model ..... 120  
 User groupware .....127  
 Multilingual sites.....136  
 10 tips for better interfaces ... 140  
 Overlaying the controls .....144  
 Type of image: dynamic, static, generated, ... ? .....145  
 Customizing a site's look: the skin elements.....147  
 Choosing a type of button/link.....149  
 Preventing going back to a page.....150  
 Security management: managing the "Back" button .....151  
 Cookies: information stored on the computer of the user .....157  
 Protecting the access to the site: passwords.....158  
 TLS/SSL: encrypting information on the web .....159  
 Secure payment with provider .....160  
 Sending emails.....161

**PART 5: DATABASES**

Analysis: Structure of the database .....165  
 The different types of accessible files .....176  
 HyperFileSQL Classic .....178  
 HyperFileSQL Client/Server.....179  
 HyperFileSQL Client/Server clusters .....180  
 HyperFileSQL: the files physically created.....181  
 Associating the controls with the data .....182  
 The queries.....184  
 Embedded queries.....185  
 The Table/Looper control .....186  
 Retrieving data from a site .....188  
 Universal replication .....193  
 3-tier architecture .....194

**PART 6 : RUNNING THE TEST OF A WEB SITE**

Running the test of a site: The elements to test.....197  
 How do I run the test of a site? .....198  
 The debugger ..... 200  
 When can I use the debugger? .....201

**PART 7: DEPLOYING A WEB SITE**

Deploying a static site .....211  
 Deploying a dynamic WebDev site.....213  
 On-lining a dynamic WebDev site.....217

**PART 8: HOSTING THE WEBDEV SITES**

What is the WebDev administrator intended for? .....227  
 Configuring the server ..... 233  
 Dedicated server or shared server? ..... 234  
 What type of server should I choose? ..... 236  
 Hosting Control Center .....237  
 Dynamic site on specific configurations..... 238  
 Traffic statistics of dynamic sites.....241  
 Monitor your sites, servers, .....242

**PART 9: APPENDICES**

WLanguage functions specific to WebDev 17 .....245  
 Components supplied with WebDev ..... 250  
 Examples supplied with WebDev .....251



**PART 1**

**Internet: Main concepts**

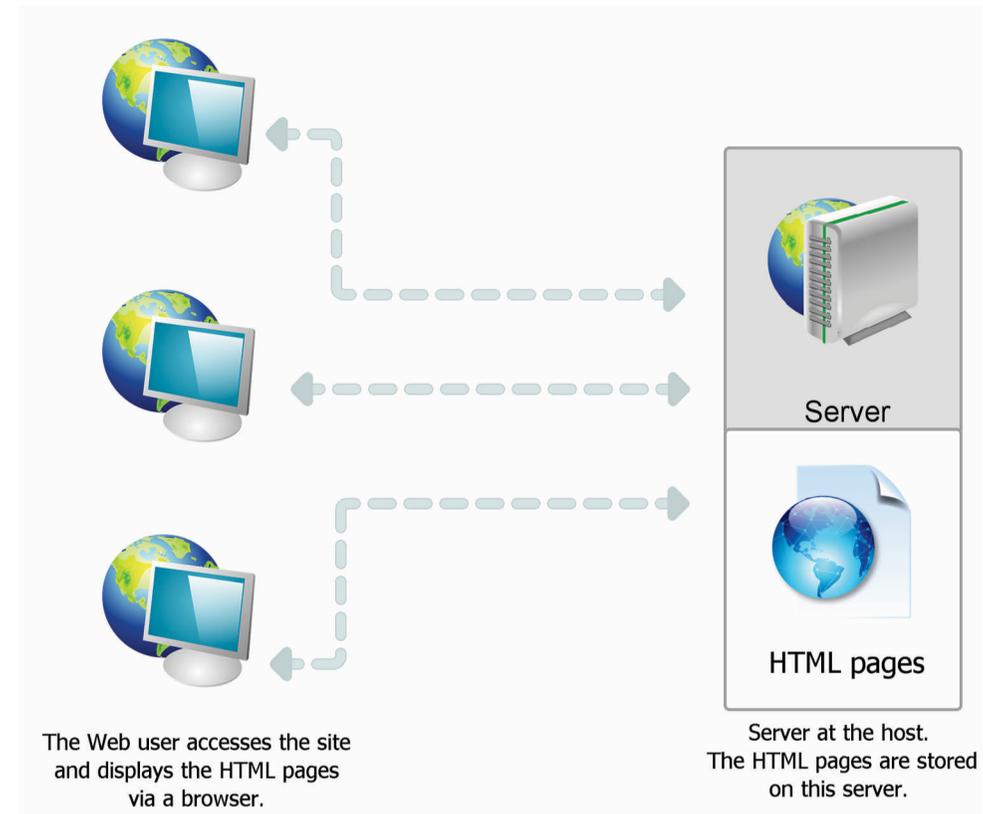


# What is a Web site?

A Web site is a set of HTML (HyperText Markup Language) pages stored on a Web server. These HTML pages are organized for a specific purpose (present a company, sell products ...).

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

WebDev enables you to easily create Web sites that manage (or not) the data.



# Static, semi-dynamic or dynamic site?

Several types of sites can be created:

- static sites,
- semi-dynamic sites,
- dynamic sites.

The table below presents the differences between these types of sites:

Static site	Semi-dynamic site	Dynamic site
<p>The content of the pages is fixed, it is defined once and for all. A static site cannot interact with the data.</p> <p>Static WebDev site: the WebDev Application Server is not required.</p>	<p>The content of the pages is built from the data found in the data files. However, the content of the pages is fixed.</p> <p>Semi-dynamic WebDev site: the WebDev Application Server is not required.</p>	<p>The data displayed in the pages changes. In most cases, the pages are used to:</p> <ul style="list-style-type: none"> <li>- perform processes and/or calculations by programming.</li> <li>- display the data stored in a database.</li> <li>- display images and interactive text.</li> </ul> <p>Dynamic WebDev site: the WebDev Application Server or the PHP language are required.</p>

**Note:** A dynamic site can contain a static section (description of the company, ...)

WebDev enables you to create:

- static pages,
- semi-dynamic pages,
- dynamic pages.

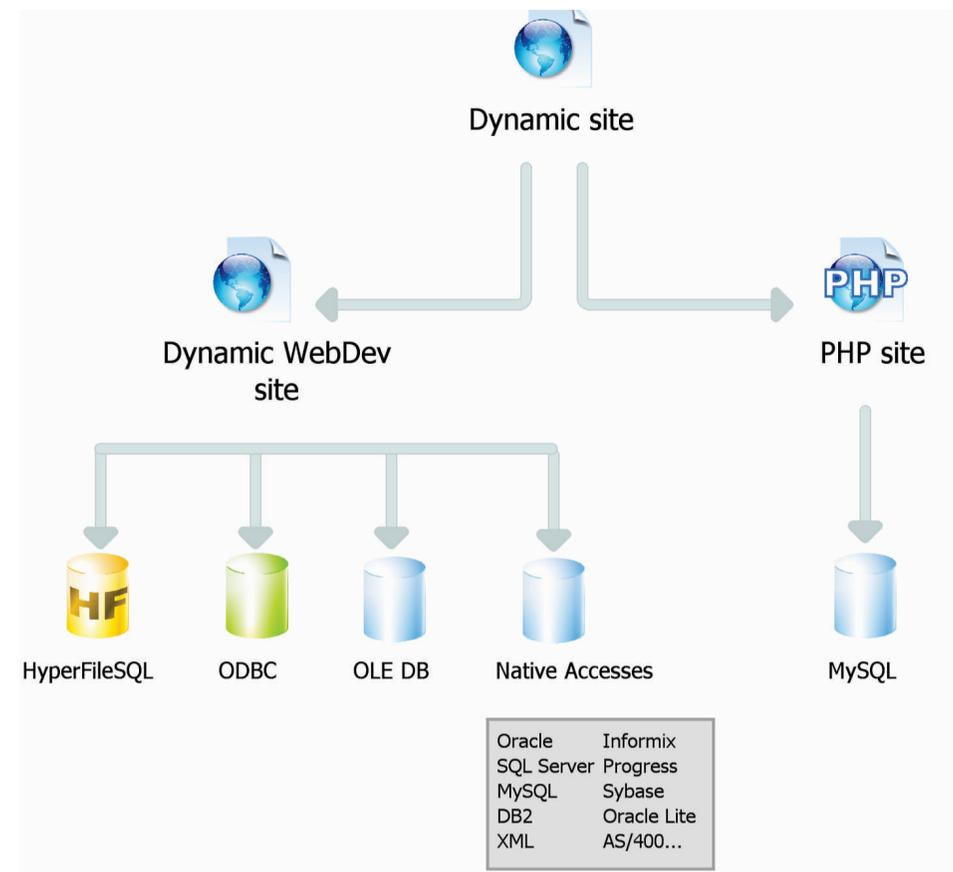
# The different types of dynamic sites

A dynamic site is used to display pages whose content changes. In most cases, the content of these pages is linked to a database.

A dynamic site can be used to manage real-time travel booking for example. These bookings are saved in a data file and they can be displayed at any time.

WebDev proposes two types of dynamic sites:

- **dynamic WebDev site.** This type of site requires a WebDev Application Server on the server. It can be used to handle several databases.
- **dynamic PHP site.** This type of site requires a PHP engine on the server. This type of site is recommended for the dynamic sites hosted by a public hosting company.



# Hardware and software requirements

## The Web user

- **a machine:** PC, Mac, Unix, Smartphone, ...
- **a browser:** Internet Explorer, Firefox, Chrome, ...
- **an Internet** (or Intranet) access

No module to download. This operating mode is fast and immediate.

### The server

- **1 server:** PC
- **1 server operating system**  
(Windows or Linux): Windows 2000 or later.
- **1 Web server software:**  
IIS, Apache, ...
- **1 Application Server WebDev**  
or
- **1 PHP engine**  
(version 4.3.2 or later)

### The development computer

- **1 computer:** PC
- **1 operating system:**  
Windows 2000 or later.
- **1 Web server software:**  
IIS, Apache, ...
- **1 WebDev "Development"**
- **the site currently developed**
- **the data** (optional) \*
- **at least one browser:**  
Internet Explorer, Firefox, Chrome, ...

\* The data can be found on another computer linked via network.

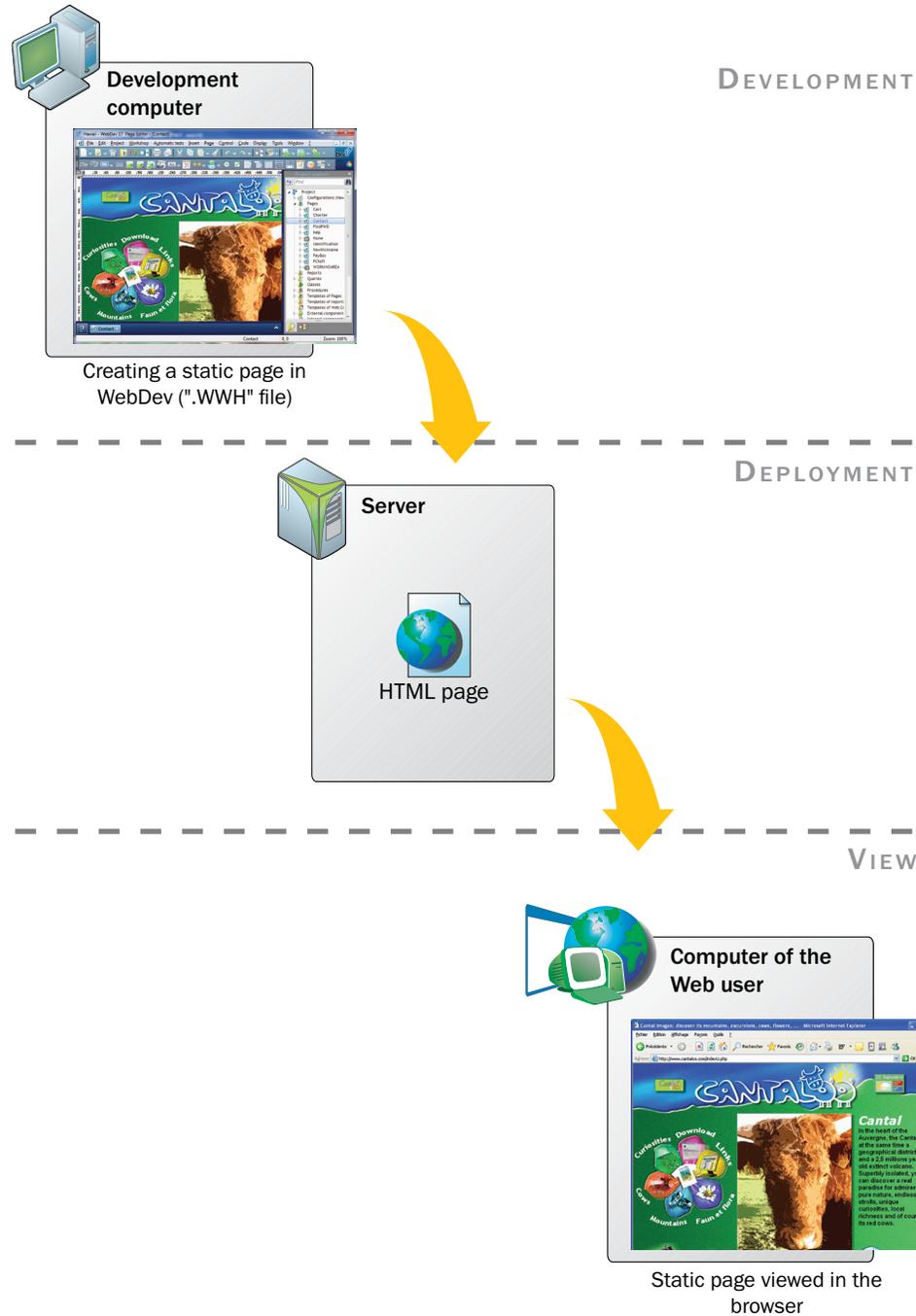
# The different types of pages

Different types of pages can be used in the sites developed by WebDev:

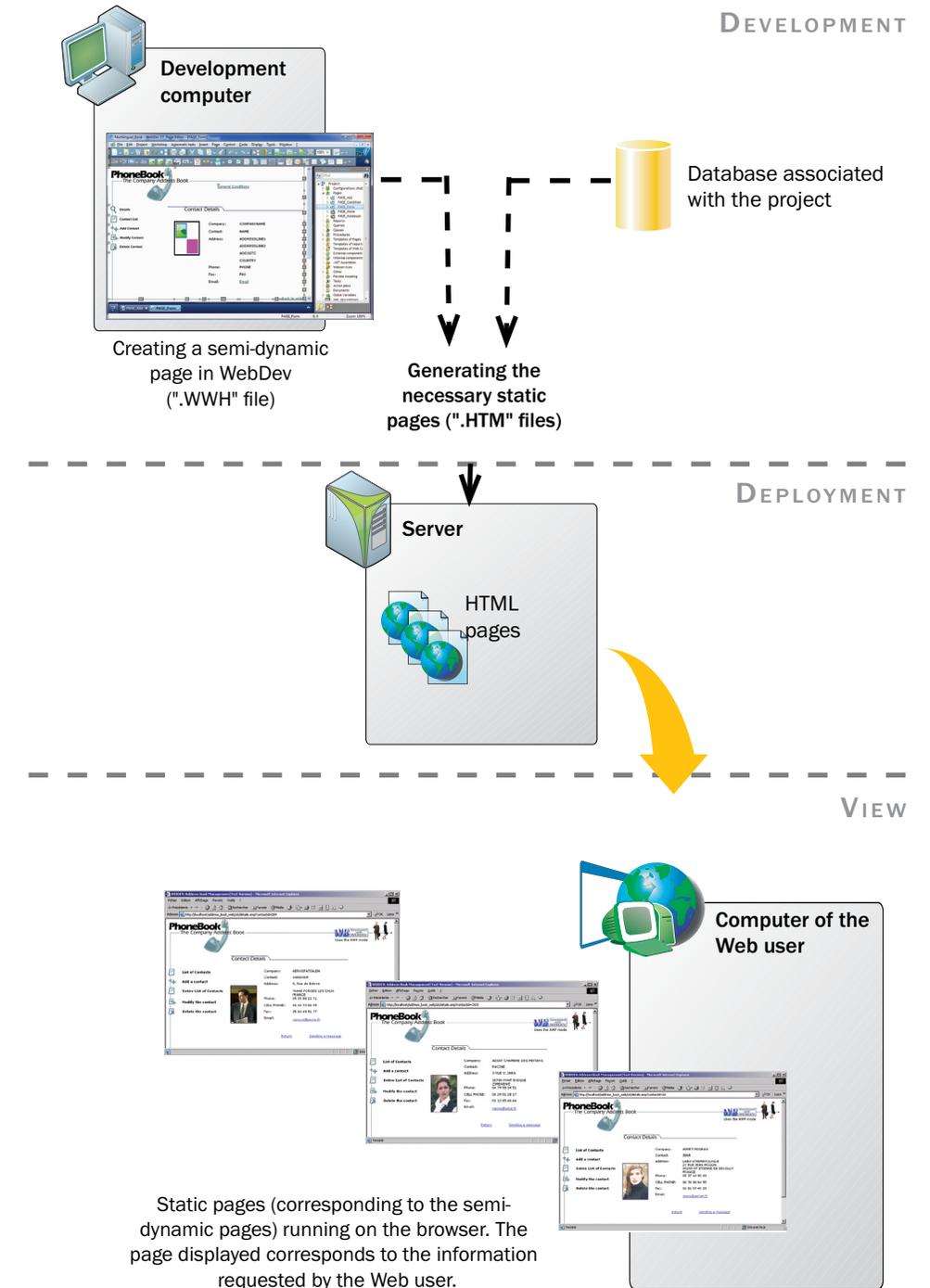
- **The static pages.** This type of page are used to display fixed data.
- **The semi-dynamic pages.** These pages are used to display data available in a database. These pages and their content are generated on the development computer when the site is created. When the HTML pages corresponding to the dynamic pages are created, the content of the page is defined.
- **The dynamic WebDev pages.** These pages are used to dynamically display the information found in a database. The content of the page depends on the record displayed. When a dynamic page is displayed, the corresponding context is automatically saved and updated on the server. The context of the application mainly contains the global variables, the variables for positioning in the data files, ...
- **The dynamic WebDev AWP pages.** These pages are used to dynamically display the data found in a database. The content of the page depends on the record displayed. The AWP pages may (or may not) use the context of the application. This context can be stored as a cookie or on disk. If the AWP pages do not use the context of the application, they are stand-alone pages.
- **The dynamic PHP pages.** These pages are used to dynamically display the data found in a database. These pages can only be used in a PHP site.

The following pages present theses different types of pages.

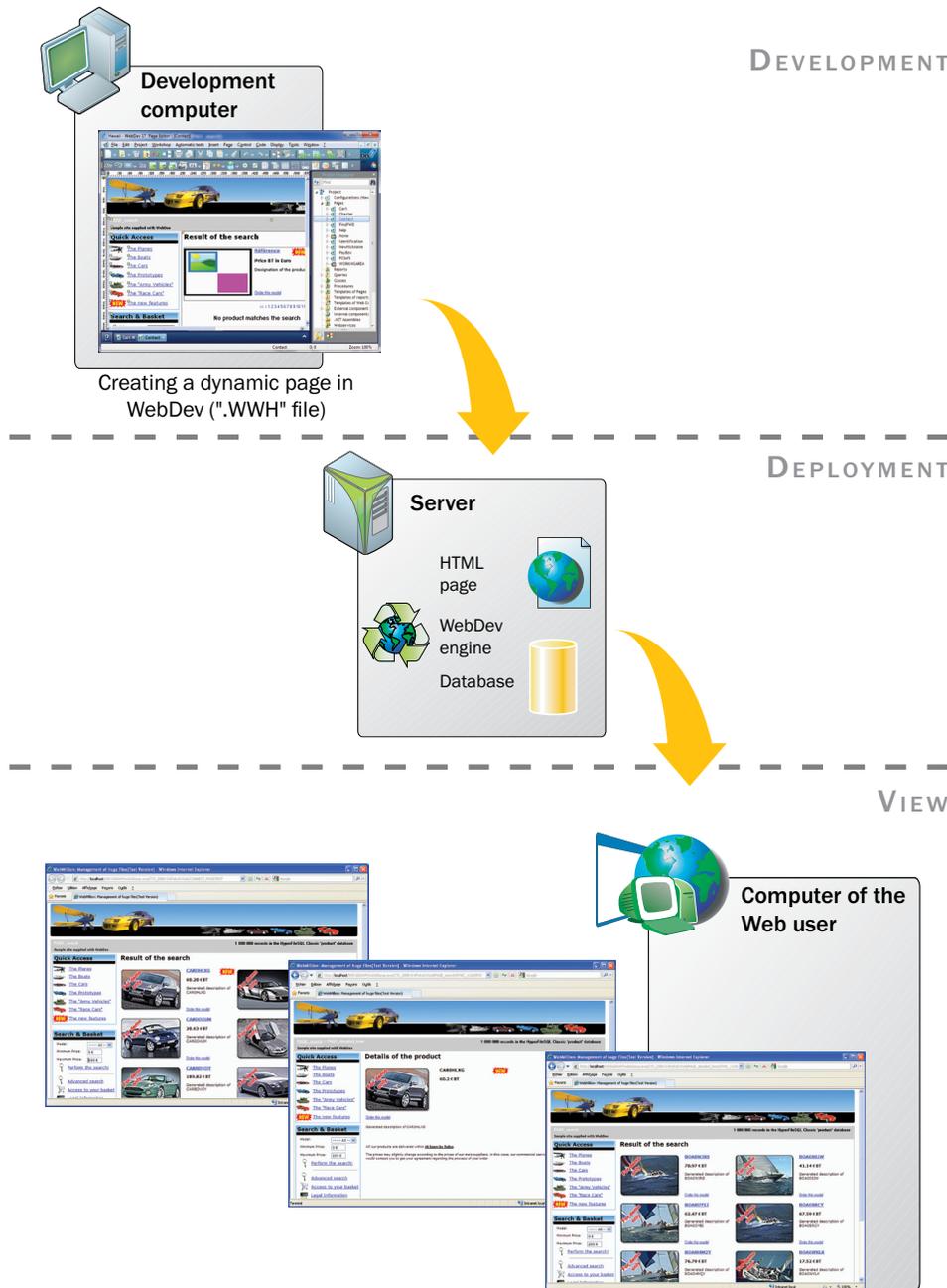
### Principle of a static page



### Principle of a semi-dynamic page



### Principle of a dynamic page



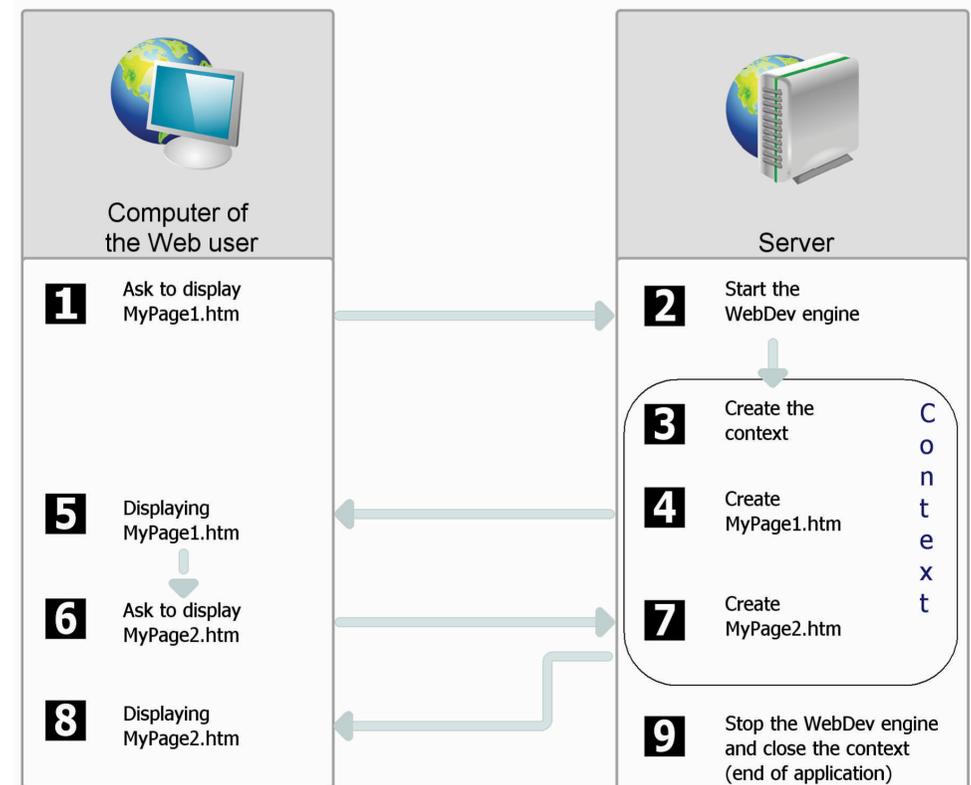
Dynamic page run in the browser.  
The data displayed in the pages changes according to the actions performed by the Web user.

# Dynamic pages and AWP pages

## Dynamic page

The following operations are performed when a dynamic WebDev page is displayed:

1. Ask to display the page.
2. Start the WebDev engine. This engine will remain on the server until the end of the application.
3. Create the context of the application. This context will remain on the server until the end of the application.
4. The WebDev engine runs the server code and builds the HTML page (from the data found in the database for instance).
5. Once the HTML page is entirely built, the server sends the result to the client (the browser).

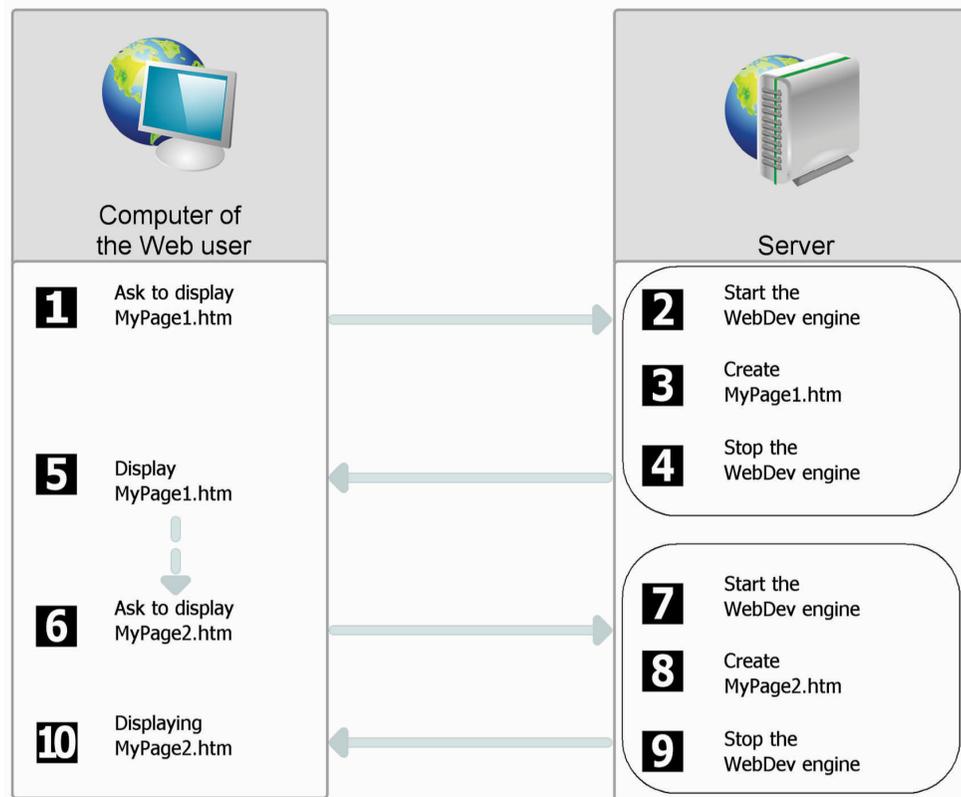


### AWP pages

An AWP page is a binary file containing the HTML code of the page and the server code (entered by the developer in the code editor). This file must have the ".awp" extension in order to be interpreted by the server.

When a browser (the client) wants to access a dynamic AWP page created by WebDev:

1. The server recognizes the file as an AWP file.
2. The server calls the WebDev engine to read the AWP file.
3. The WebDev engine runs the server code and builds the HTML page (from the data found in the database for instance).
4. The engine is closed once the HTML page is entirely built.
5. The server sends the result (the HTML page) to the client (the browser).



**Note:** a context can be used with the AWP pages by defining persistent variables via `DeclareAWPContext`.

# AJAX technology

The AJAX technology is available in WebDev.

### What does AJAX mean and what are its benefits?

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.

This technology presents several benefits:

- the server is less used. Therefore, it can support an important number of simultaneous connections.
- the information that circulates is less bulky.
- the transmission time is shortened.
- the display is immediate and without visual effect for the web user.

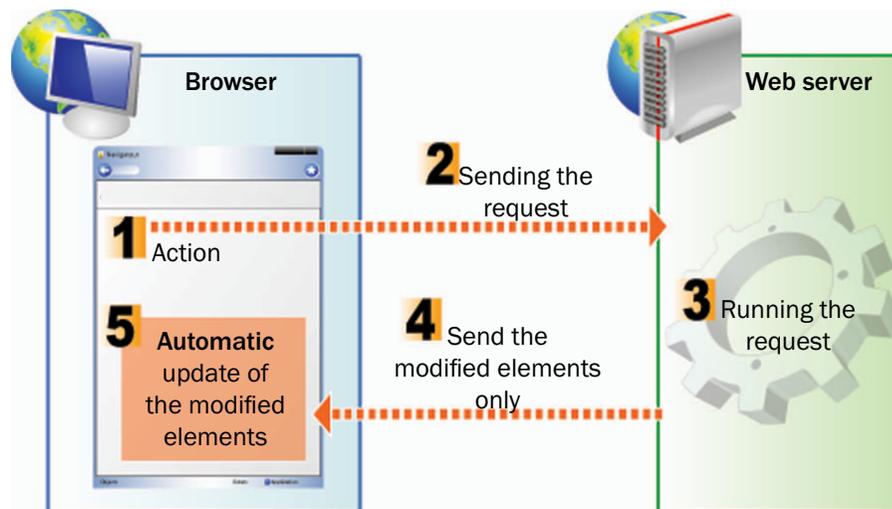
AJAX can be used at two different levels in a WebDev site:

- Automatic and instant AJAX: a simple click allows you to benefit from the advantages of AJAX. The code remains the same.
- Programmed AJAX: the functions for AJAX management are used to perform the complex processes.

**Note:** Only recent browsers support the AJAX technology; Internet Explorer 5.5 and higher, Firefox 1.0 and higher, Netscape 7 and higher, Opera 8 and higher, Safari 1.2 and higher, ... The function named `AJAXAvailable` is used to find out whether the current browser supports AJAX. If a process that uses the AJAX technology is run on a browser that does not support this technology, the process is run "as if" it did not use the AJAX technology (the entire page is refreshed for example).

### Automatic and immediate AJAX

The diagram below presents the automatic and immediate use of AJAX in a WebDev site:

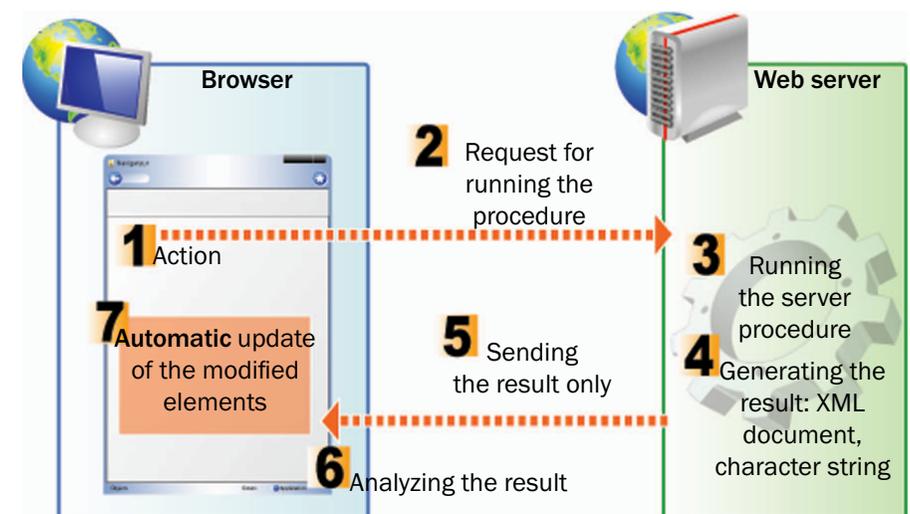


For example, a site page is used to find out the different characteristics of a country (capital city, currency, flag, location, ...). The corresponding information is displayed according to the country selected by the Web user.

- 1** Action performed by the Web user. In our example, the Web user selects the country in the combo box named "Select a country".
- 2** Sending the request to the server.
- 3** Running the query: find the characteristics of the selected country.
- 4** Sending the result of the request:
  - without AJAX: the entire page is sent.
  - with AJAX: only the characteristics of the country are sent.
- 5** Displaying the characteristics of the country:
  - without AJAX: the entire page is redisplayed.
  - with AJAX: only the controls containing the characteristics of the country are refreshed.

### Programmed AJAX

The diagram below presents the use of "programmed AJAX" in a WebDev site:



- 1** Running a browser process (*AJAXExecute* or *AJAXExecuteAsynchronous*).
- 2** Request for running a server procedure.
- 3** Running the server procedure.
- 4** Generating the result. The result of the procedure will be contained in a character string or in an XML document.
- 5** Sending the result of the procedure (*RESULT* keyword).
- 6** Analyzing the result of the procedure.
- 7** Displaying the modified information. Only the necessary controls will be refreshed.

# If you are already familiar with WinDev

The main differences between WebDev and WinDev are as follows:

- WebDev is used to create pages while WinDev is used to create windows.
- In WebDev, different types of code can be entered: some WLanguage code run on the server, some WLanguage code run on the browser and JavaScript code run on the browser. When you create a PHP page, an additional code is displayed: a PHP code run on the server.
- New WLanguage functions specific to the Web are available.
- Some WLanguage functions not relevant for the Web do not exist anymore.
- Some WLanguage functions can only be used in an executable process in server code.
- Some WLanguage functions can only be used in an executable process in browser code.
- New types of controls designed for Web applications are available:
  - Formatted display control
  - Java Applet control
  - Captcha control
  - Cell control
  - Site Map Path control
  - Flash control
  - Flex control
  - Clickable Image control (Map Area)
  - iFrame control
  - HTML Static control
  - Link control
  - Horizontal Rule control
  - Internal Page control
  - Peeling Corner control
  - Site Map control
  - Pager control
  - SilverLight control
  - HTML Table control
  - Upload control
  - Thumbnail control
- Some types of controls that serve no purpose for Internet are no longer available: scrollbar, slider, ActiveX, OLE Object, spin, ...

## PART 2

### Developing a site



# Project and Analysis

The development of a **Web site** through WebDev relies on two main elements: the Project and the Analysis.

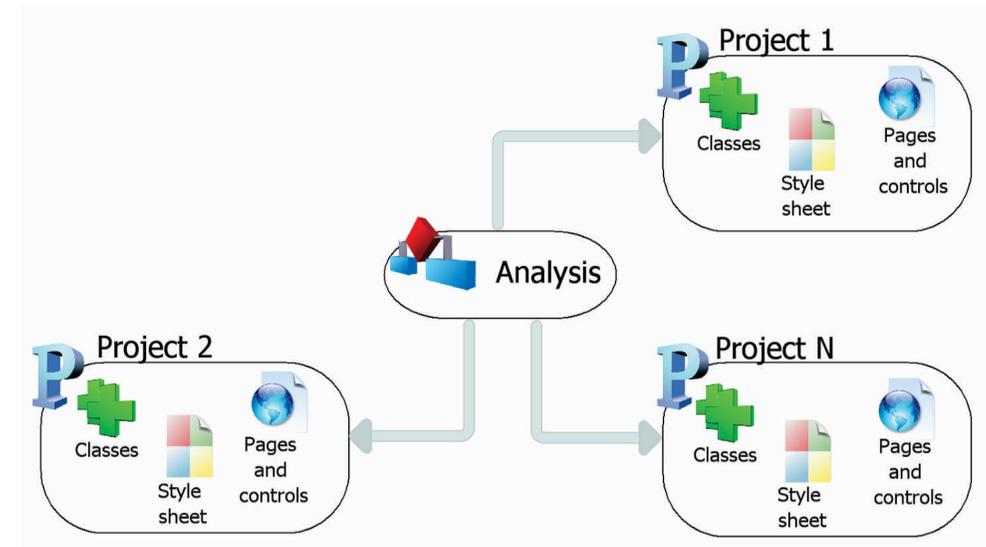
A **WebDev project** is made of a set of elements: pages, reports, controls, classes, components, ... whose combination allows you to build a Web site.

A **WebDev Analysis** groups the description of the data files of the site.

A site is built from a project.

In most cases, a project is associated with an analysis.

An analysis can be associated with one or more projects.



# The project in practice

## 1 Overview

The first step when describing a site consists in creating a project.

When creating a project, several questions will be asked to you in order for the project to correspond

to your expectations. All the characteristics specified when creating the project can be modified later.

## 2 Creating the project

To create a project:

1. Select "File .. New .. Project". The wizard for project creation starts.

2. Specify the different options of the project:

- **the name and location.** These options cannot be modified. The project corresponds to a ".WWP" file. All the objects associated with the project will be created in the specified directory.
- the **generation type of the project**, that is the type of site that will be generated by the project: static site, dynamic WebDev site, semi-dynamic site, dynamic PHP site, ...
- the **creation mode of the project**: you have the ability to create a blank project or a project based on the Applicative RAD.
- **whether the project will be handled by several developers.** You can use the developer groupware or the source code manager (SCM) to share the project.

• **the programming charter.** This charter is used to automatically prefix the variables, the names of the controls, the names of the pages, ...

• the style book.

• the supported languages. These languages will be proposed by default as soon as an option found in a control, in a page, in a report, ... can be translated.

3. Specify **whether the project is associated with a database or not.** If yes, the database can already exist or it can be created.

4. Validate the wizard. The created project becomes the current project.

**Notes:**

- If you have requested the creation of an analysis, the corresponding wizard is automatically started.
- The specified information can be modified in the project description ("Project .. Project description").

## 3 Project dashboard and project graph

### 3.1 Dashboard

Any project manager would like to have a global and synthetic view of the status of his projects.

Any quality manager would like to know the number of critical bugs and follow their status.

The product manager would like to know what new features are requested by the users.

The developer would like to directly and quickly

start the project elements, the most often used sections of code, ...

The dashboard gives an answer to all these wishes. The dashboard includes several lights that provide a global view of the different indicators. The lights switch from green to red whenever a section of the project requires special attention.

To display your project's dashboard, all you have to do is select "Project .. Project Dashboard".

## 3.2 Project graph

The project editor enables you to graphically view the project elements (pages, reports, queries, ...) as well as their sequence.

To display the project graph, all you have to do is select "Project .. Project graph".

## 3.3 The project elements

The project includes some pages, reports, queries, sets of procedures, ...

To find out the list of project elements, select "Project .. List of project elements".

This option is used to:

- add to your project the elements belonging to the projects accessible from your computer. The corresponding files will not be moved into the directory of your project.
- delete elements from your project. The corresponding files are not physically deleted.

To quickly find an element in your project, press "CTRL + E" from any editor.

## 4 Operations performed on a project

The main operations that can be performed on a project are as follows:

- Archiving a project
- Restoring a project

- Duplicating a project
- Copying or deleting a project
- Renaming a project

See the online help for more details.

## 5 Project documentation

WebDev allows you to print several documents that present all the elements (page, report, data file, item, ...) found in the project. To print these documents select:

- "Project .. Print the project documentation". The documentation can contain all the characteristics of the project.
- "File .. Print the documentation". In this case, the documentation contains the characteristics of the current element (page, report, query ...).

### Edit mode of the documentation

The documentation can be:

- printed,
- exported to a RTF document, to an XML file,
- exported to an HTML file (an Internet browser must be installed on the current computer),
- exported to a text file.

### 5.2 Print areas

To print graphic representations (project graph, LDM, UML diagrams, ...), the print areas must be configured before printing the documentation.

To specify the print areas of the current graphic representation:

1. Select "Display .. Print areas". Borders representing the areas that can be printed in the documentation are displayed in the current editor.
2. Reduce the display of the graphic representation ("Display .. Zoom" or [Ctrl] + Mouse wheel) to view the entire graph.
3. With the mouse:

- moves these borders to the requested location by keeping the mouse button down (the mouse cursor is black).
- choose the number of pages on which the graphic representation must be printed (the mouse cursor turns into a North-West/South-East double

### 5.1 Type of documentation

Several types of documentation are proposed:

- Analysis/Data files/Queries documentation: Contains the information about the analysis, the data files and the queries of the project.
- GUI documentation (Graphical User Interface): Contains the presentation of pages and reports as well as their sequence.
- Code documentation: Contains all the processes of all the elements found in the project.
- Technical documentation: Contains the processes and the technical description of all the elements found in the project.
- Full documentation: Contains all the information about the project. This documentation combines all the other types of documentation.

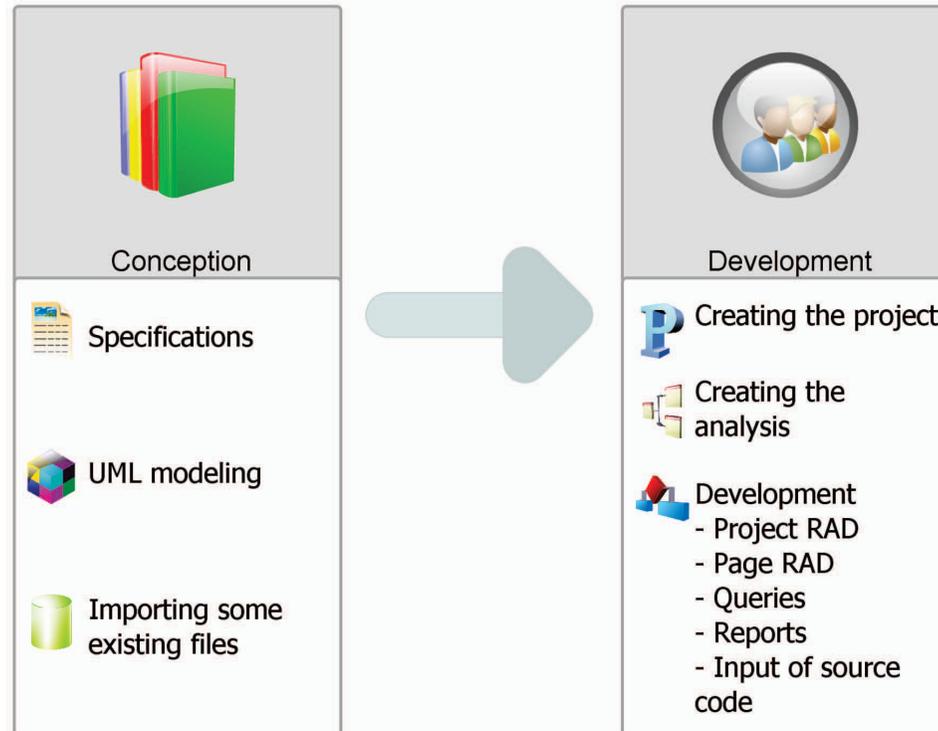
arrow).

If the print format is modified (from A4 to A3 for example in the properties of the printer), the documentation pages can be adapted to the new format. To do so:

1. Define the print areas.
2. Display the options for documentation layout ("File .. Documentation layout").
3. Select "Adapt the size of the print areas for the opened documents".
4. Print the documentation.

# Development cycle of a site

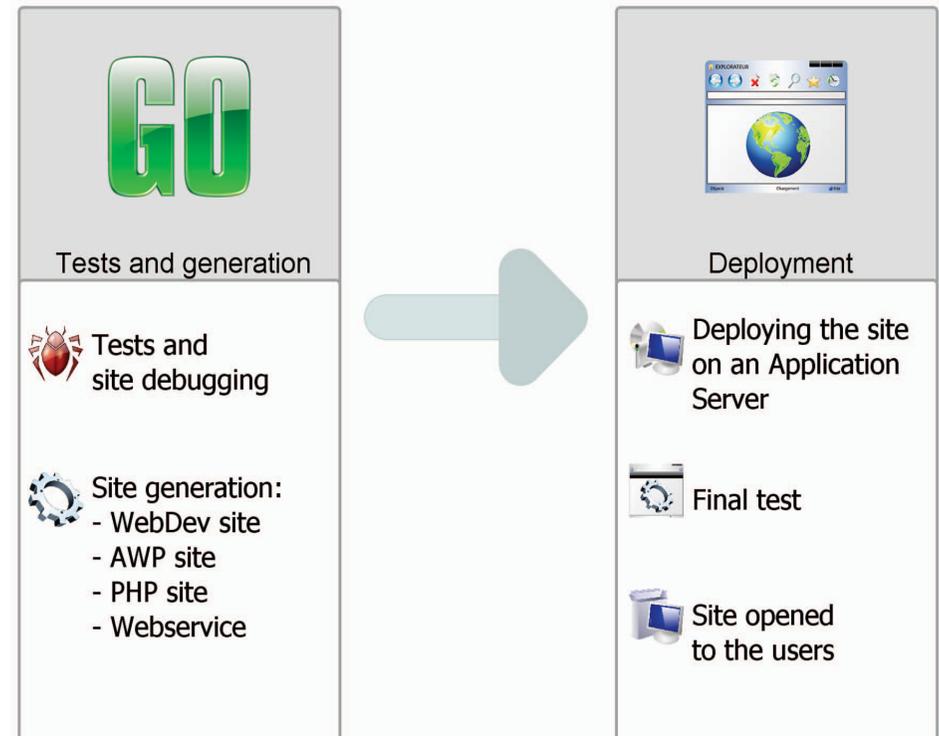
WebDev covers the entire development cycle of a site:



## Details of the different steps:

**Design step:** A site can be designed from specific requirements, from UML modeling of processes or even from existing data files.

**Development step:** The creation of the project and analysis is done via very extensive wizards. The development can be done in RAD mode (Rapid Development Application) with automatic generation of the code and GUI or it can result from the manual creation of the project elements.



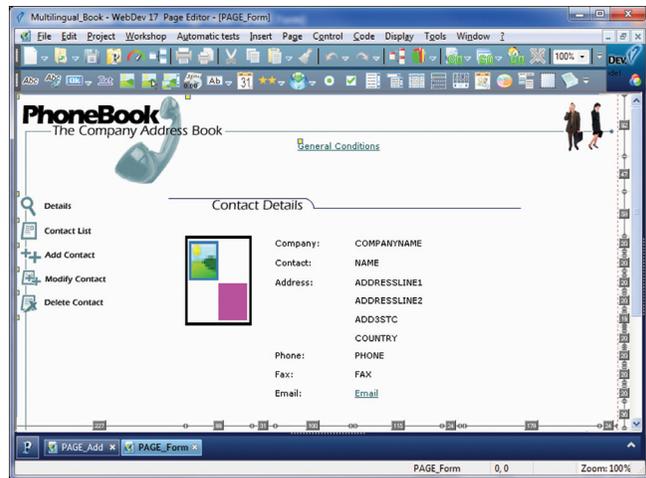
**Test and generation step:** WebDev offers several automated testing tools to guarantee the reliability of the applications and insure that there is no regression during the development phase.

**Deployment step:** A dynamic WebDev site is deployed on a WebDev application server. Once the site is deployed, you have the ability to run the last tests before releasing the site to the users.

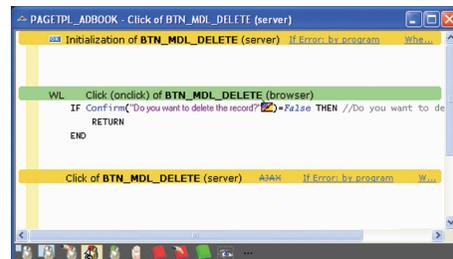
# I want to create a page in the editor ...

The following steps are required to create a page in WebDev:

- 1 Creating the page in the page editor of WebDev.



- 2 Entering the code (optional step).



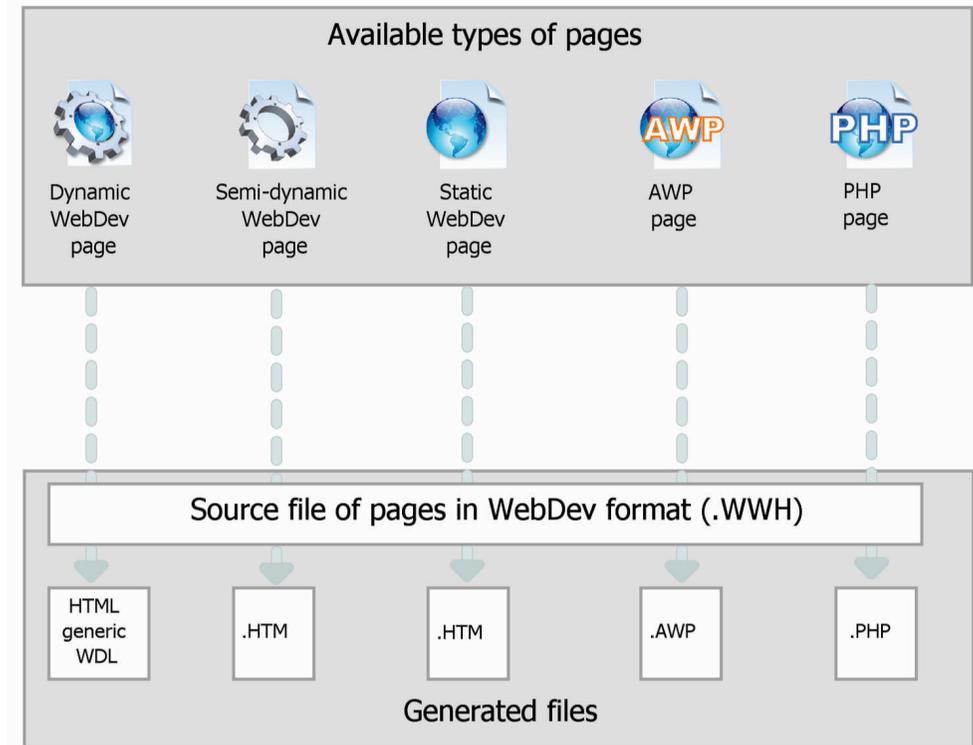
Browser code



Server code

- 3 Saving the page.

# The following files are automatically created:



**Legend:**

**.WWH: Source format of pages**

This file contains the full description of the page (control, browser code and server code). This file is used by the editor and it remains on the development computer.

**.WDL: Project library**

The library is generated when the dynamic site is deployed and it contains the server code of the project pages. This file will be found on the server.

**.Generic HTML**

This file contains the full description of the page (control, browser code and server code). Some parameters of this page are generic parameters and they will be filled dynamically when the page is displayed.

**.AWP: AWP page**

This file is generated from the WWH file and it contains the description of controls, the server code, the browser code. This file will be found on the server.

**.HTM: HTML page of the site**

This file is generated from the WWH file and it contains the description of controls in HTML and the browser code (if the code was entered in WLanguage, it is automatically translated into Javascript). This file will be found on the server.

**.PHP: PHP page**

This file is generated from the WWH file and it contains the description of controls, the server code, the browser code in PHP format. This file will be found on the server.

# Editing a page: zoning mode

Editing a page in zoning mode is easy and powerful: all you have to do is split the page into "zones" via the "Pencil" tool. These zones are automatically configured and let you define the page's architecture: header, footer ... These zones can easily be handled in the editor: increasing the size of the header automatically moves the body of the page.



When printing, the relative position of the controls found in each zone is respected as well as their anchoring.



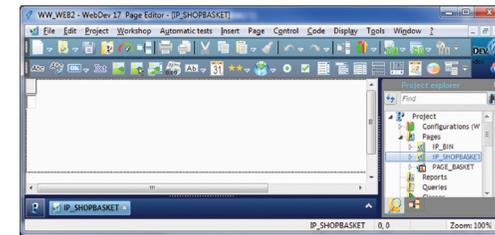
A semantic HTML 5 information can be associated with each zone: this improves the relevance of the referencing with some search engines.

# Internal page

The Internal Page control is used to include a page (and its code) in another page. At run time, the internal page will be dynamically merged to the host page.

## 1. Creating an internal page

An internal page is a specific page with no title bar and no menu. All types of controls can be used in this page.

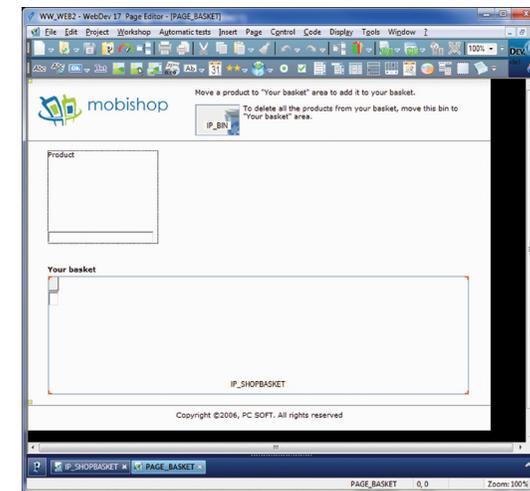


In the example, an internal page is used to manage the basket.

## 2. Using an internal page

To use an internal page, you must:

- create an internal page control.
- in the control description, select the internal page to use and validate.

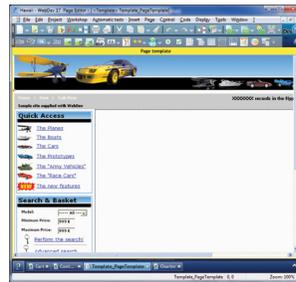


### Notes:

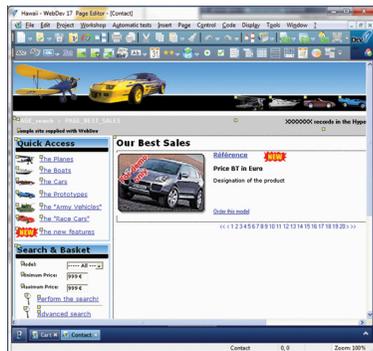
- The internal page used in the "Internal page" control cannot be modified by programming.
- The host area is rectangular and no overload is allowed. To perform overloads, we recommend that you use the control templates.

# Page templates

WebDev allows you to create page templates. A template is intended to contain the elements common to a set of site pages. The modifications performed in a page template are automatically applied to all the pages that use this template. A page template enables you to comply with the style book defined for a site.



Defining a page template. The template is bordered by a yellow line in the editor.



Using the template in several pages. The elements belonging to the template are identified by a yellow square.

To create a page based on a template, select the template that will be used when creating the page. **Note:** The programming associated with the template elements can be directly performed in the template.

The characteristics of the elements can be dissociated from the template. For example, the position of a control can be dissociated from the template so that the control can be located somewhere else while it still benefits from the other features (code, style, ...). We talk of **inheritance**. In this case, the elements are identified by a blue square.

## Pages in practice

The purpose of a page is to display, consult and enter information. This information can come from the data files found in an analysis, external files, queries, ...

Several methods can be used to create the pages of a project:

- Creating a blank page with the wizard.
- Building all the site pages from the analysis description.
- Creating a page from the analysis description (with or without its code).
- Creating pages based on a template, ...

Regardless of the method used, the page can be modified after its creation: you have the ability to add, modify or delete controls, to modify the characteristics of the page.

For more details, see "The controls in practice", page 49.

This chapter presents the following topics:

- "Creating a page", page 39
- "Simple operations on a page", page 40
- "Main characteristics of a page", page 40
- "Internal page", page 40
- "Page templates", page 41
- "Refreshing a dynamic page", page 41
- "The page contexts", page 41
- "Opening a page in a WebDev site", page 42
- "Processes associated with the pages", page 43
- "The menus", page 43

### 1 Creating a page

WebDev proposes several methods for creating a page:

- Creating blank pages, without control.
- Creating preset pages (dynamic RAD, semi-dynamic RAD, PHP or standard RAD).
- Creating pages via the project RAD.

**Note:** The main characteristics of pages are presented in details in the online help.

#### 1.1 Creating blank pages

To create a blank page:

1. Select "File ..New .. Page".
2. Choose the "Blank" type.
3. Specify the page template.
4. Validate the creation of the page. A blank page is created.
5. Save the page ("File .. Save").
6. Specify the name of the page. The page corresponds to a ".WWW" file. By default, this file will be created in the main directory of the project. This name will be used to handle the page.
7. Create the controls in the page.

The created page is a dynamic page by default. The type of the page can be modified in the description window of the page. For more details, see "Main characteristics of a page", page 40.

The different types of controls are presented in "The different types of standard controls", page 45.

#### 1.2 Creating preset pages

To create a preset page:

1. Select "File ..New .. Page".
2. Choose the type of page to create. This page can be:

- **Blank/Template:** Used to create a blank page that uses the selected page template.
  - **Internal page:** Used to create internal pages that will be used in controls whose type is "Internal page". The internal pages are used to include a page (and its code) in a control of another page.
  - **Dynamic RAD:** This page will be used to view and/or modify the records of one or more data files.
  - **Semi-dynamic RAD:** This page will be used to view the records of one or more data files.
  - **RID:** This page will be used to view and/or modify the records of one or more files. Only the interface of the page is generated. The code must be written by the developer.
  - **Frameset:** Used to create different types of Framesets. Each frame must be associated with the pages of your project in the editor.
3. Depending on the selected type of page, enter the requested information in the different screens of the wizard (these screens depend on the selected type of page).
  4. Validate the creation of the page.

### 1.3 Creating the project pages by automatic site building (R.A.D.)

For the dynamic sites, the project pages can be created automatically in a single operation from the analysis description, by automatic site building (RAD).

See the online help for more details.

To create the dynamic site by R.A.D.:

1. Select "Workshop .. Full Application RAD".

**Caution:** To use this option, the project must be

associated with an analysis that was generated at least once.

2. Select the RAD pattern to use.

3. Select the data files used for the generation.

4. Select the main data files that will be the entry points in the application (these data files will be mainly used to implement the menu options).

5. Associate (if necessary) an image with each data file. This image will be used to represent the data files.

6. Validate the creation of your application by RAD.

## 2 Simple operations on a page

The page editor allows you to perform the following operations on the pages:

- Opening a page in the editor: Double-click the name of the page displayed in the "Project explorer" pane.

- Saving and copying a page.

- Importing and exporting a page.

- Modifying the navigation order.

See the online help for more details.

## 3 Main characteristics of a page

A page can accept all types of controls.

A page can:

- have a **background image**: this image is displayed in the page background and the controls are displayed on the top of this image.
- have a **drop-down menu**: this menu will give the users the ability to quickly access the main features of the site. To create a drop-down menu in a page, all you have to do is select "Insert .. Menu". The first two options are displayed in the

page. The options can be handled via the popup menu of options. Then, the menu options can be handled by programming. For more details, see "Handling the menu options by programming", page 44.

- be **multilingual**: all the languages defined for the page will be available for all the page controls. A page can be associated with more languages than the project (pages shared among several projects for example). See for more details "Multilingual sites", page 136.

## 4 Internal page

The internal pages are used to dynamically share the same interface section inside a site or among several sites.

The interface that must be used several times is created in an "Internal Page" page.

This interface is used in the different pages of your site via the "Internal page" control.

**Note:** The page to merge can come from a component.

See the online help for more details.

## 5 Page templates

WebDev allows you to create page templates. These templates contain the graphic elements and the code common to all the pages of your site.

The modifications performed in a page template are automatically applied to all the pages that use this template.

A page template enables you to comply with the style book defined for a site.

Using some page templates in your sites allows you to:

- simplify the creation of the site pages.

- simplify the layout of the site pages.
- simplify the update of the style book defined for the site.

The method for creating a page template is similar to the method for creating a page. To create a page via a template, all you have to do is choose the "based on a template" option.

By default, any modification performed in the template is applied to the pages that use it. However, special cases can be managed in a specific page by overloading the template elements.

See the online help for more details.

## 6 Refreshing a dynamic page

### 6.1 Overview

Refreshing a page consists in re-displaying a page that is already opened in the browser. This action is identical to the use of the "Refresh" button in the browser.

When refreshing a page, the dynamic data found in the page is updated according to the page context found on the server. Only the modified data is re-displayed.

For example, when entering an order in a sales application, the page that displays the summary of the order (the basket) is refreshed whenever a new item is ordered.

### 6.2 Implementation

To refresh a page, all you have to do is use the WLanguage function named **PageRefresh**

When refreshing a page, the following operations are automatically performed:

1. Checking the existence of the page context on the server.
2. Re-displaying the page according to its context found on the server.

Using **PageDisplay** to refresh a page is not recommended because it takes much longer to load the page.

## 7 The page contexts

### 7.1 Overview

In a Windows application, the information relative to a window is stored with the window.

In a dynamic WebDev site, a context page exists on the server for each page displayed in the browser.

The page context groups the information relative to the page:

- content of controls,
- local variables,
- global variables,
- WLanguage "server" code, ...

### 7.2 Implementation

#### Automatic management

By default, WebDev automatically manages the contexts of dynamic pages:

- A page context is opened when the dynamic page is displayed in the browser.
- The context of a page is updated according to the information entered by the Web user in the browser. This update is performed during the validation of the page (via a "Submit" button or via **PageSubmit**).

- The existing page contexts are closed when using **PageUse** or **FramesetUse**. Then, the page contexts corresponding to the pages to display are opened.

### Advanced management

WebDev enables you to perform an advanced management of the contexts via the ContextXXX functions. You can:

- open the context of a page without displaying it in the browser (**ContextOpen**),
- check the existence of a page context on the server (**ContextExist**),
- close the context of an unused page (**ContextClose**).

## 8 Opening a page in a WebDev site

### 8.1 Overview

Several methods can be used to open a new page in a WebDev site:

- from the description of the control that must open the page (button, link, menu, ...)
- by programming in WLanguage.

### 8.2 Opening a page from a control description

When describing a button, a link or a clickable image, you have the ability to define the action performed as well as the destination of this action.

To open an "xxx" page from a button, a link, an image, ... :

1. Display the description of the element ("Description" from the popup menu of the element).
2. Select the action to perform: "Display the xxx page".
3. Select the destination of the action: current page, current browser, ... (specific frame for a frameset). The action that was previously selected will be performed in this destination.
4. Validate.

This advanced management of page contexts can be used to:

- Save memory space on the server (by closing the contexts of pages previously opened but not used for example).
- Prepare the pages intended to communicate with the other applications (the page receiving information back from a site for secured payment for example).
- Handle the pages from other pages.

### Special case

When using the browser "Back" key, a desynchronization may occur between the page and its context. For more details, see "Security management: managing the "Back" button", page 151.

### 8.3 Opening a page by programming

Several WLanguage functions can be used to open a page.

- **PageRefresh**  
Refreshes the page displayed in relation to its context.
- **PageDisplay**  
Opens and displays a new page in the browser of the Web user.
- **PageUse**  
Closes all the current pages and their contexts, and opens a new page.

When opening the page, you have the ability to pass parameters to the page. See the online help for more details.

**Note:** By default, when opening a page by programming, the selected destination is:

- the destination defined in the description of the object that triggers the opening of the page (button, link, ...).
- the destination defined for the current page.

### 8.4 Case of dynamic pages

When displaying a dynamic page, the following actions are performed:

1. Checking the existence of the page context on the server.

2. The context is closed if it exists.

3. Opening the context of the page. All the variables, controls, ... linked to the page are re-initialized.

4. Displaying the requested page in the browser.

## 9 Processes associated with the pages

### 9.1 Processes managed by default

WebDev manages the following processes by default (in the order in which they appear in the code editor):

- **Global declarations (server code):**  
Declaration of the variables global to the page. This is the first code run when the page is opened.
- **Initialization (server code):**  
Run when the page is opened. The initialization process of the page controls is run before this code.
- **Load (onload) of the page (browser code):**  
Browser code run when the page is displayed in the browser.
- **Unload (onunload) of the page (browser code):**  
Browser code run when a new page is displayed in the browser.
- **Synchronization of the page (server code):**  
Server code used to manage the synchronization of the page when using the browser "Back" key.
- **Closing of the page (server code):**  
Run when closing the page.

- **Assignment through the ..Value property (internal page only)**  
Run when using the Value property on assignment on the Internal Page control.
- **Retrieve the ..Value property (internal page only)**  
Run when using the Value property in read mode on the Internal Page control.

### 9.2 Optional processes

Several optional processes can be managed.

To manage an optional process, all you have to do is select it in the icon bar of the code window.



You have the ability to manage:

- the rollover of the control by the mouse,
- the left mouse button down, up, with double click,
- the right mouse button down, up, with double click,
- the mouse wheel, etc.

## 10 The main menus

WebDev enables you to create a drop-down menu in your HTML pages. This menu can be oriented vertically or horizontally. From the different options of this menu, you can:

- display pages,
- run the code of the buttons found in the page,
- ...

WebDev enables you to:

- Create a main menu
- Edit a main menu
- Delete a main menu.

See the online help for more details.

## 11 Menu options

### 11.1 Overview

A menu includes one or more options and sub-options. Each option is used to run a WLanguage code.

To handle the menu options, all you have to do is edit the relevant menu.

To edit (or display in the editor) the main menu of a page:

1. Open the relevant page in WebDev.
2. Click the menu twice: a yellow border appears.

### 11.2 Handling the menu options in the editor

The page editor allows you to easily perform the following operations:

- Add a menu option
- Add a separator
- Add a sub-menu
- Delete a menu option

The options can also be handled in the window editor and their characteristics can be modified via a

description window.

The description window of the menu options allows you to:

- modify the caption of the option.
- check or uncheck a menu option
- associate an image with a menu option
- associate a keyboard shortcut with a menu option.

You also have the ability to associate a WLanguage code with a menu option. Only the menu options without sub-menu can start a WLanguage process.

See the online help for more details.

### 11.3 Handling the menu options by programming

To handle a menu option by programming, use the following notation:

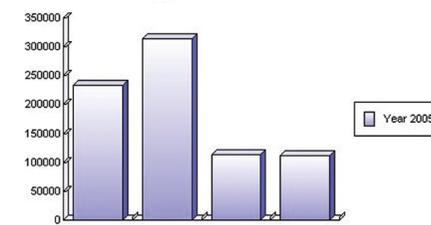
- "<OptionName>" if the menu option is handled from the page.
- "<PageName>.<MenuName>.<OptionName>" if the option is handled from another page.

# The different types of standard controls

You want to...	Use the following control...
Display a text, a title, ... 	HTML static, Static
Define a color area	Cell, HTML table
Display a price, a quantity, an address, a date, a time, ...	Formatted display control
Select a value from a list (Country, City, ...) 	Radio button, Combo box, List box
Select several values from a list (recipients of a message, files to download, ...)	Check box, List box
Display a graphic image (photo, ...) 	Image, Clickable image

Display an animation (animated Gif)	Image
Display a page according to the area that was clicked in an image	Clickable image
Display the content of a data file in a table (list of customers, order details, ...)	Memory table or browsing table, Memory loop or browsing loop
Display an image refreshed on a regular basis	Web Camera
Repeat controls in a page (product catalog with photo, ...)	Loop
Use an existing HTML code	HTML control
Use a Java applet (clock, ...)	Java control
Enter information	Edit control
Display ranked information (content of a directory, ...)	TreeView control

Display a Flash animation	Flash control
Align controls (HTML table)	Cell, HTML table
Program an action in a page (display a page, validate an input, ...)	Button, Link control
Select and display a date on a calendar	Calendar control
Display a video	Video control
Display a page from another site in one of your pages	IFrame
Display a bar chart, a line chart, a pie chart	Chart control



Display a page from your site inside one of your pages	 Internal page
Display an automatic menu (built while the site is browsed)	 Site map path
Allow the Web user to give or view a rate. 	 Rating control
Display thumbnails	 Thumbnail control
Display Flex files	 Flex control
Include SilverLight controls	 SilverLight control
Expand/Collapse a display area	 HideShow control
Upload one or more files	 Upload control
Display a list of objects as a scrolling horizontal list.	 Linear control      Looper
Create a visual effect in a page by presenting an image partially covered by another one.	 Peeling control      Corner

## The controls in practice

WebDev proposes several types of controls that can be easily included in your pages. These controls can be created either via the "Control" toolbar, or

via the "Insert" menu of the page editor. All the controls can be handled by programming.

### 1 Creating a control

The following controls are available in the page editor of WebDev:

- HTML static,
- Static,
- Button,
- Link,
- Image,
- Clickable image,
- Formatted control,
- Edit control,
- Check box,
- Radio button,
- List box,
- Combo box,
- Table,
- Looper,
- Pager,
- Captcha,
- Rating,
- TreeView,
- Cell,
- HTML table,
- Horizontal rule,
- Tab,
- Web Camera control,
- Java Applet control,
- Flash control,
- IFrame control,
- Internal page control,
- Site map path,
- Site map
- Supercontrol,
- Chart,
- Calendar,
- Scheduler,
- Bar code
- Control template.

#### 1.1 Creating a new control

To create a control:

1. Select the type of control to create:
  - via "Insert .. Control/Special".
  - via the corresponding icon in the "Controls" toolbar.
2. The shape of the new control appears below the mouse cursor.
3. Click the position where the control must be created in the page. The control is automatically created.

**Note:** Other creation modes are available. See the online help for more details.

#### 1.2 Creating a control associated with an item

Except for the following controls, all the page controls can be associated with an item found in a data file:

- Button,
- Tab
- Horizontal rule
- HTML control
- Java Applet control
- Flash control
- pager
- cell and HTML table
- Site map

Several methods can be used to create a control associated with an item (or to retrieve an item):

- Drag and Drop from the "Analysis" pane.
- Using the "File" tab of the control.

## 2 Characteristics of a control

The description window can be displayed for all the controls found in a page. This window includes several tabs, that group the configurable characteristics of one or more controls.

**Note:** You can also view and/or modify the characteristics of one or more controls in the modifier. See the online help for more details.

### 2.1 Displaying the characteristics

To display the description window of a control:

- double-click the control.
- select "Description" from the popup menu of the control (right mouse click).
- select the control and select "Display .. Description of the selection".
- select the control and press [Alt]+[Enter].
- select "Display .. Options .. Modify the options" ([F12]), select the control and click the "Edit" button.

Notes:

- The description window can be displayed for a set of selected controls. Only the characteristics common to the selected controls will be displayed.
- Several description windows can be displayed at the same time. Each description window displays the characteristics of one or more controls.

### 2.2 Characteristics by tab

This paragraph briefly presents the different categories of characteristics displayed by each tab.

See the online help about the description windows for more details.

#### General tab

The "General" tab is used to specify the name of the control and all the display characteristics of the control (caption, input mask, ...).

#### GUI tab

The "GUI" tab is used to define the different parameters of the control:

- Initial status of the control when the page is opened.
- Visibility of the control
- Type of the control

#### Details tab

The "Details" tab is used to define the different parameters of the control for the execution:

- Automatic completion.
- Mandatory input.
- Accessible by TAB.

#### Link tab

The "Link" tab is used to select the item to which the control is linked. Depending on the current record, the content of the linked item will be displayed in the control.

The link can be single-file or multi-file.

#### Content tab

The "Content" tab is available for the page controls only.

The "Content" tab is used to define:

- the initial content of the control (for the edit controls only).
- the data source used to fill the control (for the list boxes, combo boxes and tables only).

#### Note/Help tab

The "Note/Help" tab is used to:

- describe the operating mode of the control. This information will be printed in the program documentation (project documentation, page documentation, ...).
- configure all the types of help associated with the control. A control can have:
  - A tooltip, displayed when the control is hovered by the mouse cursor.
  - A help message, displayed in the status bar when the control takes focus.

#### Advanced tab

The "Advanced" tab is used to enter the HTML code generated before and after the control.

#### Style tab

The "Style" tab is used to define the style of the different control elements. This screen is used to:

- Modify the aspect of a control: all you have to do is select the control element to modify, then its style characteristics. Only the aspect of the current control is modified.
- Create or modify a style.
- Choose a style.

## 2.3 Dynamic control, static control

WebDev proposes the following options for all types of controls:

- **Automatic:** the control will automatically adapt to the type of use.
- **Static:** the control cannot be modified by programming: the properties associated with the control will have no effect.

- **Dynamic:** the control can be modified by programming: all the properties associated with the control can be used.

**Note:** if a control is defined as being dynamic, the HTML code of the page will contain specific code to manage the programming of the control in WLanguage. Therefore, the HTML file corresponding to the page will be larger than if the control is defined as static.

## 3 Handling the controls in the editor

### 3.1 The handles of controls

When a control has just been created or when it is selected, it is displayed with handles. The handles are used to:

- view the size of a control,
- modify the size of a control.

The colors of the handles are used to identify the actions that can be performed:

- The **black handles** are used to view and resize a control.
- The **white handles** only let you view a control. For example, when selecting several controls, the white handles indicate that the size of the selected controls cannot be modified. This type of handles is also displayed when a page is in read-only for the developer: the controls cannot

be modified.

- The **gray handles** show the first control selected during a multiple control selection. This control will be the reference control.

### 3.2 Available operations

The page editor enables you to:

- Select one or more controls (via the lasso for example).
- Group the selected controls.
- Move a control.
- Modify the caption of a control.
- Display an advanced tooltip when a control is hovered by the mouse cursor. This tooltip contains: the name of the control, its position, its size, its initial status (if the control is invisible).

## 4 Alignment of controls

The alignment of controls is used to create "professional" and outstanding interfaces. Several tools help you create some standard interfaces:

- the rulers.
- the grid.
- the real-time interface checker (automatically proposed when positioning a control).
- the advanced interface checker.
- the alignment options.

- the configurable zoom.
- anchoring.

### 4.1 The rulers

The alignment rulers can be displayed in the page editor.

These rulers include snap-on guides: any control that comes close to a guide is automatically "snapped" by it. This feature enables you to easily position, align or resize the controls found in a page.

### Displaying the rulers

To display the rulers, select "Display .. Rulers". Two types of guides can be used:

- the **guides**, used to align and organize the controls inside the pages.
- the **border guides**, used to define a border of identical size on each side of the page.

**Note:** To turn the guides on or off for a page, use the "Guide visible" option from the popup menu of the rulers. Whether they are visible or not, the guides are always enabled: any control moved toward a marker is automatically snapped by it.

### Handling the markers

The markers can be easily handled with the mouse in the editor. When a marker is hovered by the mouse, the cursor turns into a double arrow.

### 4.2 Snap-on grid

The snap-on grid is used to place some vertical and horizontal markers in the page. The created controls are attracted to these markers, as if the controls and the markers were magnetic. The snap-on grid is used to align the controls according to the markers.

### 4.3 Real-time interface checker

When a control is moved in a page, some guides are automatically displayed by the real-time interface checker. The objects snap to the guides, allowing you to align the moved control with the other controls found in the page.

### 4.4 Advanced interface checker

The advanced interface checker is a tool used to standardize the layout of the controls in the different pages of a site.

The advanced interface checker proposes, for the current page or for the selected controls, to apply some rules for interface layout issued from the Windows standard: aligning the controls, standardizing the size of the buttons, ...

## 4.5 The alignment options

To align several controls, WebDev proposes:

- a preset alignment (available via "Control .. Alignment.." or in the toolbar for control alignment).
- a custom alignment ("Control .. Alignment .. Custom alignment"). The custom alignment allows you to use specific alignment properties.

When using the custom alignment, you have the ability to define:

- **the horizontal or vertical alignment:** the controls can be aligned on the left according to the beginning of the control or according to the beginning of the input area. They can also be aligned to the right according to the end of the control or according to the end of the input area.
- **the regular spacing:** the spacing between the controls is adapted in order to be identical between each control (horizontally and vertically).
- **the size:** the size of the selected controls is automatically adapted in order for the controls to have the same height and/or the same width.
- **the reference control:** for the alignment and size of the controls, you can choose the reference control to be:
  - **the first selected control**
  - **the last selected control**
  - **the largest selected control**
  - **the control located in the top left position of the selected controls**

## 4.6 The configurable zoom

A zoom factor can be specified to reduce or enlarge the display of the current page.

To specify a zoom factor:

- 1st method:
  - Select "Display .. Zoom..." and choose the requested zoom factor.
  - Enter the zoom factor in the status bar of the editor (bottom right).
- 2nd method:
  - Press the [Ctrl] key.
  - Modify the zoom factor with the mouse wheel while keeping the [Ctrl] key down.

## 5 Options for editing the controls in a page

The edit options ("Display .. Options .. Modify the options") are used to configure:

- the display options.
- the magnetism of controls.
- the click and double-click operations on the

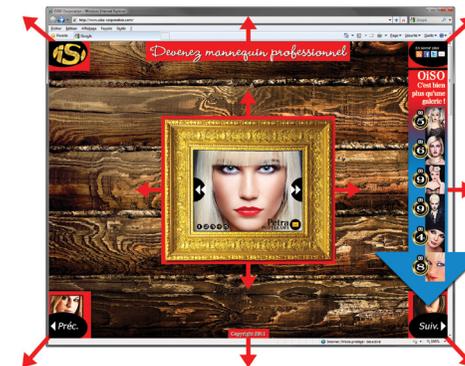
controls.

- the options for control selection.

These display options are used to customize the operations performed on the controls in the page editor.

## 6 Anchoring the controls

The pages of a WebDev site can be resized at run time: the browser can be resized, the site can be displayed on different platforms (IAD, iPhone, PC, ...). The anchoring mechanism is used to automatically adapt the size and position of the controls when the page is resized.



The anchoring of a control is managed via several parameters:

- **the anchoring of the control:** this parameter is used to define the modification that will be applied to the control when the page is resized. The control can move toward the left and/or toward the bottom, increase its width and/or its height.
- **the management of the width and/or height:** this parameter is used to manage the behavior of the control when stretched. The height or width can be adapted to the content of the control or to the browser.
- **the anchor rate:** this parameter is used to specify the percentage by which a control is moved or stretched.

These parameters can be configured in the page editor.

### 6.1 Anchoring a control in the editor

To anchor a control:

1. Select one or more controls to anchor.
2. Define the anchoring of the control(s):
  - in the "GUI" tab in the description window of the control ("Display .. Description of the selection").
  - via "Anchor" from the popup menu (right mouse click) of the control.
3. In this window, choose the type of anchoring that will be used. The options are presented as icons.
4. In this window, choose the type of anchoring that will be used. The most common options are represented by icons:

 **No anchor:** The control is not modified when the browser is resized. No anchoring option is selected.

 **Width:** The control stretches to the right when the browser is enlarged. To change the speed at which a control stretches compared to the browser, use the anchor rate in width.

 **Right:** The control moves to the right when the browser is enlarged. To make the control move slower or faster, use the anchor rate to the right.

 **Horizontally centered:** The control remains centered in width in the browser regardless of the browser size.

 **Height:** The control stretches to the bottom when the height of the browser is increased. To change the speed at which a control stretches in relation to the browser, use the anchor rate in height.

**Width and Height:** The control stretches to the right and to the bottom when the browser is enlarged. To change the speed at which a control stretches compared to the browser, use the anchor rate in height and the anchor rate in width. Select the management mode of the height and width.

**Height and Right:** The control stretches to the bottom and moves to the right when the browser is enlarged. To make the control move slower or faster, use the anchor rate to the right. To change the speed at which a control stretches in relation to the browser, use the anchor rate in height.

**Height and centered horizontally:** The control stretches to the bottom and remains horizontally centered when the browser is enlarged.

**Bottom:** The control moves toward the bottom when the browser is enlarged toward the bottom. To make the control move slower or faster, use the anchor rate to the bottom.

**Width and Bottom:** The control stretches toward the right and moves toward the bottom when the browser gets bigger. To make the control move slower or faster, use the anchor rate to the bottom. To change the speed at which a control stretches compared to the browser, use the anchor rate in width.

**Right and Bottom:** The control moves to the right and to the bottom when the browser is enlarged. To make the control move slower or faster, use the anchor rate to the bottom and the anchor rate to the right.

**Horizontally centered at bottom:** The control remains centered in width in the browser regardless of the browser size. However, the control is anchored at the bottom and it moves to the bottom when the browser is enlarged.

To make the control move slower or faster, use the anchor rate to the bottom.

**Vertically centered:** The control remains centered in height in the browser regardless of the browser height.

**Width and centered vertically:** The control stretches to the right and remains vertically centered when the browser is enlarged.

**Vertically centered to right:** The control remains centered in height in the browser regardless of the browser height. However, the control is anchored to the right and it moves to the right when the browser is enlarged. To make the control move slower or faster, use the anchor rate to the right.

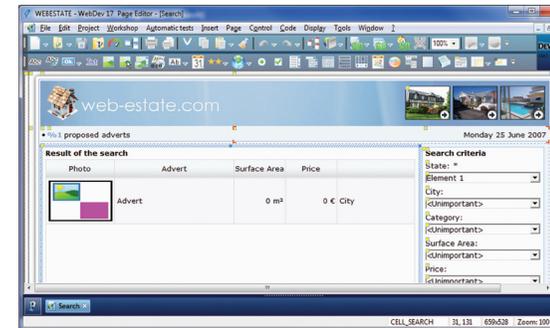
**Centered:** The control remains centered in height and in width in the browser regardless of the browser size

5. Define (if necessary) the different anchor rates.
6. Validate. If "Display .. Options .. Show the anchors" is enabled, the anchoring markers are automatically displayed in the control (red arrows).
7. Validate. The anchor marks are automatically displayed in the control (red arrows).

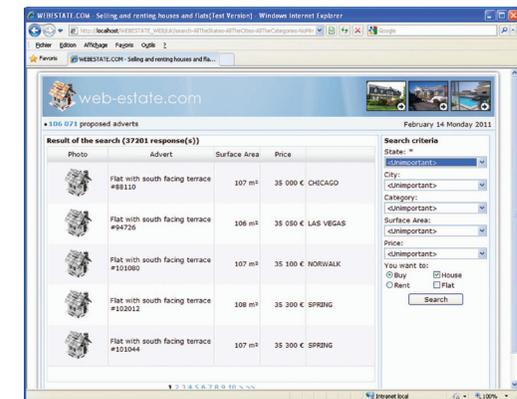
Note: The positioning tables can also be used to manage the anchoring of controls. See the online help for more details.

# Repeating a group of controls: the loopers

The loopers are used to repeat a set of controls in a Web page. Different information can be displayed in each occurrence of the loop. The records found in a database can be displayed on each row.



Creating the loop in the WebDev editor



Displaying the loop control in the browser

Several characteristics can be modified when the controls found in the loop are repeated: caption, color of the text ...

The repetitions can be performed:

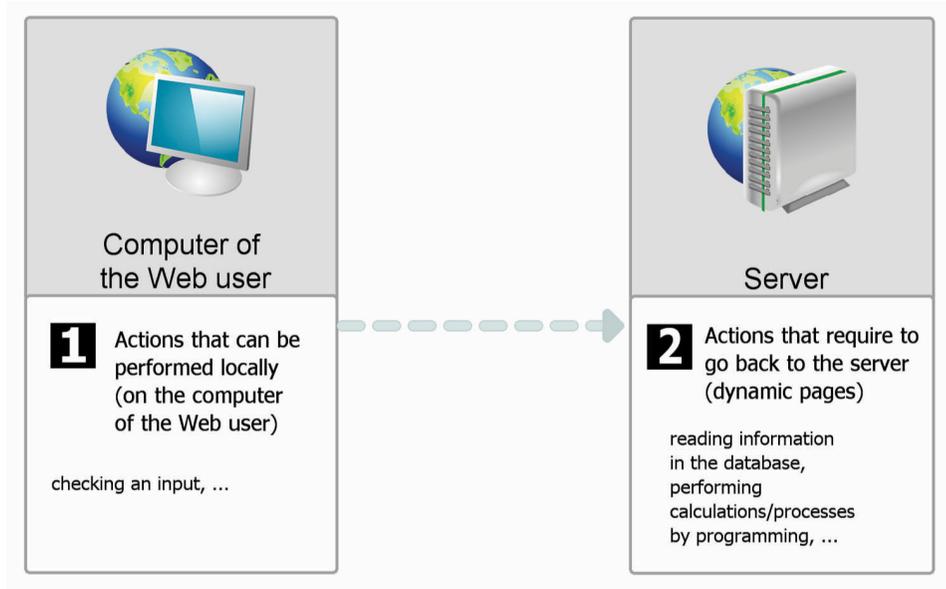
- on a column,
- on several columns. In this case, we talk of "Horizontal loop".

When creating a loop, the loop can be directly linked to a data file or to a query. The loop attributes are automatically defined. No programming is required.

# The two types of code

## Server code or browser code?

Two types of actions can be programmed in a WebDev site:



To manage these two types of actions, the WebDev code editor differentiates between two types of code:

- **Server Code** (yellow or pink code in the code editor): This code is written in WLanguage (yellow code) or in PHP (pink code, available for PHP pages only). This code is run on the server. This code is available in the dynamic pages only.
- **Browser Code** (Green or Blue code): This code is written in WLanguage (green code) or in JavaScript (blue code). When the page is saved, this code is automatically translated into Javascript and included in the WebDev HTML pages. This code is run locally (on the computer of the Web user) and it requires no server action.

## Server code in a PHP page: WLanguage code or PHP code?

The code that runs on the server is represented by a yellow or pink bar in the code editor.



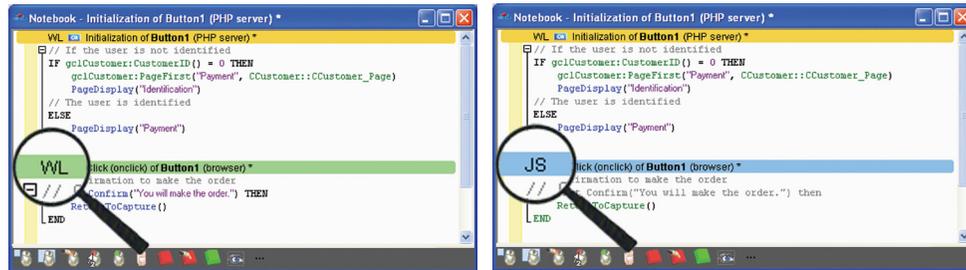
- Yellow bar = WLanguage: the code header is preceded by WL.
- Pink bar = PHP: the code header is preceded by PHP.

To switch between yellow code (WLanguage) and pink code (PHP), click the WL symbol found in front of the code (and conversely for PHP). When the PHP page is saved, the WLanguage code is automatically converted into PHP code.

**We recommend that you develop in WLanguage.**

## Browser code: WLanguage code or JavaScript code?

Code run on the computer of the Web user (in the browser) is represented by a green bar or by a blue bar in the code editor.



- green bar = WLanguage: the code header is preceded by WL.
- blue bar = Javascript: the code header is preceded by JS.

To switch between green code (WLanguage) and blue code (Javascript), click the WL symbol found in front of the code (and conversely for JS). When the page is saved, the WLanguage code is automatically converted into Javascript.

**We recommend that you develop in WLanguage.**

### Additional events

The icon bar found at the bottom of the code window is used to manage the additional events:



These events are added to the code displayed in the code editor by clicking the corresponding code icon. Example of additional events: double-click, key down, key pressed, key up, button down, mouse moved, ...

**Reminder:** the browser code is available in dynamic pages, semi-dynamic pages and static pages.

# WLanguage: a powerful language

WLanguage is the language of WebDev. This language is used to easily program all the requested Web processes without even knowing HTML, JavaScript or PHP.

Available in English and in French, WLanguage is made of simple commands, close to everyday language. It allows for intuitive programming.

An example is used to illustrate the power of WLanguage: checking the input of an address:

**In WLanguage, only a few code lines are required:**

```
IF NoSpace (ADDRESS) = "" THEN
    Error ("Specify your address")
    ReturnToCapture (ADDRESS)
END
```

**Here is the equivalent process in JavaScript:**

```
<SCRIPT LANGUAGE=Javascript>
function EXE_NoSpace (szString, nPart)
{
    var nFirst = 0;
    var nLast = szString.length-1;
    if (nPart & 1) while (szString.charAt(nFirst)==" ") nFirst++;
    if (nPart & 2) while (szString.charAt(nLast)==" ") nLast--;
    return szString.substring(nFirst,nLast+1);
}
// Click (onclick) of BUTTON_OK
function _VALIDORDR_BUTTON_OK_CLI ()
{
    if ( (EXE_NoSpace (ADDRESS, 3)=="") )
    {
        alert ("Specify your address");
        retour;
    }
}
</SCRIPT>
```

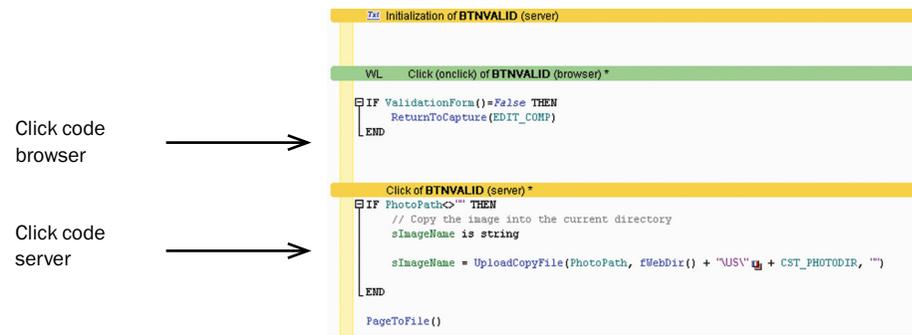
WLanguage code is easier to write, understand and maintain. It is much more reliable. Upgrading is easier.

**Use WLanguage in all your processes.**

# Sequence for running the code of buttons/links

Two pieces of click code are associated with the buttons/links:

- **browser click code**, entered in WLanguage (or in JavaScript) in the code editor. This code is included in the HTML page and it will be run on the Web user's computer.
- **server click code** entered in WLanguage (in the dynamic pages only). This code will be run on the server.



The following table presents the code executed depending on the chosen button/link and the associated action. The aspect of the button/link and its target do not affect the sequence in which the pieces of code are run.

Operation	Sequence of code run	Notes
Send to server (submit)	<b>Associated action:</b> None Run the click code of the button/link  <b>Code run:</b> 1. Browser click code of the control. 2. Browser code of page submit (if it exists). 3. Send the values found in the page controls to update the page context on the server. 4. Server code of the control.	Page context automatically updated on the server (dynamic pages only).
	<b>Associated action:</b> Display the page  <b>Code run:</b> 1. Browser click code of the control. 2. Browser code of page submit (if it exists). 3. Send the values found in the page controls to update the page context on the server. 4. Server code of the control. 5. Display the page.	Page context automatically updated on the server (dynamic pages only). Automatically display the page.
Reinitializing pages (reset)	<b>Associated action:</b> None  <b>Code run:</b> 1. Browser click code of the control. 2. Display the initial page (page with all the controls initialized to empty or to 0 and execution of the initialization code for each control then for the page).	
None	<b>Associated action:</b> None  <b>Code run:</b> Browser code of the control only.  <b>Associated action:</b> Run the click code of the button/link  <b>Code run:</b> 1. Browser click code of the control. 2. Server code of the control.	
	<b>Associated action:</b> Display the page  <b>Code run:</b> 1. Browser click code of the control. 2. Server code of the control. 3. Display the page.	Automatically display the page

# Upload: Sending files to the server

Uploading consists in sending one or more files coming from the computer of the Web user to the server.

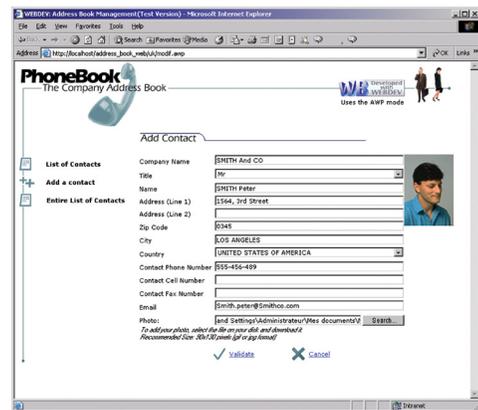
## A simple example: Creating a contact in an address book.

The Web user wants to associate a photo with the contact form. This photo is found on the hard disk of his computer.

The UPLOAD control allows the Web user to select the image via a "Browse" button.

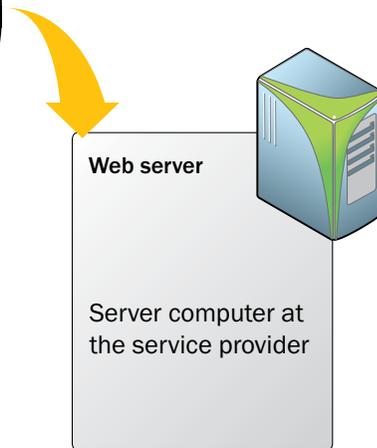
When the form is validated, the selected image file is sent to the server and copied onto the hard disk of the server (**UploadCopyFile**).

Then, the image will be displayed in the contact form.



In the browser of the Web user

When the page is validated, the uploaded file is copied onto the hard disk of the server.



Note: The aspect of the "Browse" button is defined by the browser and it cannot be customized.

**Reminder:** WebDev also proposes an advanced Upload control that requires a Flash player on the Web user's computer. This control is used to manage a progress bar, to select several files, ...

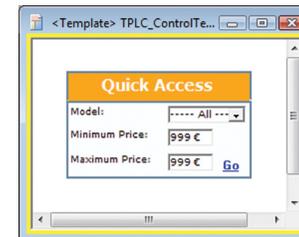
# Control templates

WebDev enables you to create control templates. A control template is a set of controls that can be re-used in several pages.

The modifications performed in a control template are automatically applied to all the pages that use this template.

A control template is used to:

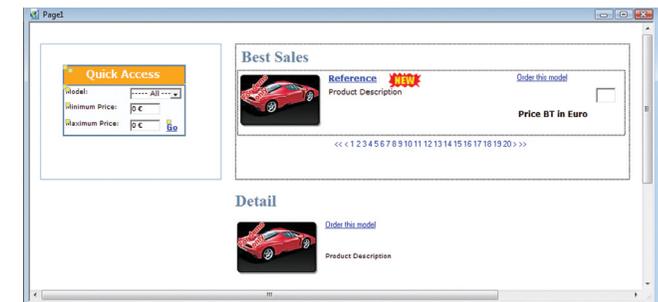
- group a set of controls for a specific purpose.
- make the controls independent of the page hosting them.



Defining a control template: the template is bordered by a yellow line in the editor.

## Using the template in a page.

The elements belonging to the template are bordered by a blue line and identified by a yellow square.



A control template can be created:

- directly ("File .. New .. Page .. Control template")
- from the controls found in the page ("Control .. Refactoring .. Create a control template from the selection").

To use a control template in a page, create a "Control template" control.

**Note:** The programming associated with the template elements can be directly performed in the template.

The characteristics of the elements can be dissociated from the template. For example, the position of a control can be dissociated from the template so that the control can be located somewhere else while it still benefits from the other features (code, style, ...). We talk of **control inheritance**. In this case, the elements are identified by a blue square.

# Reports

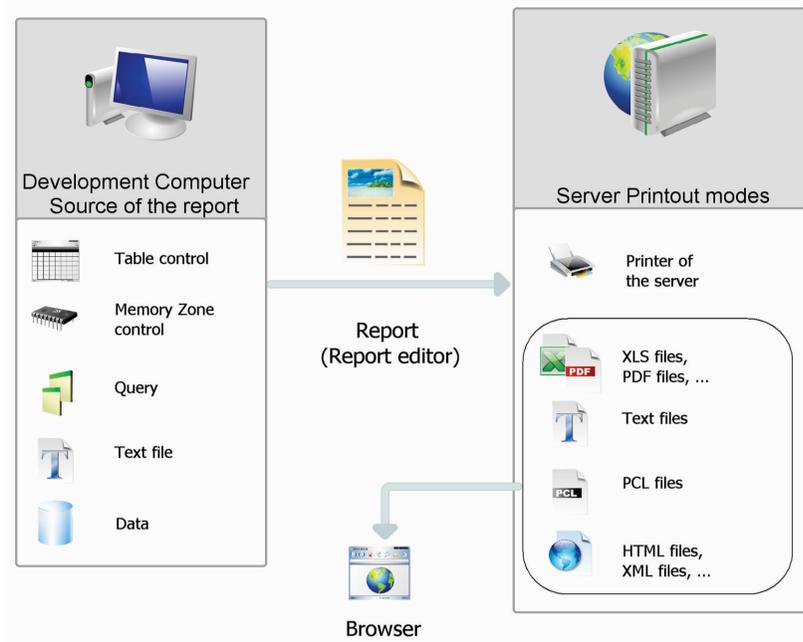
A report provides a custom view of the data: data entered in the database, data found in a text file, data coming from a Table control, ...

The principle for creating a report is as follows:

- the data to print comes from a data source (data file described in an analysis, HyperFileSQL view, query, memory zone or text file).
- the report groups, sorts and formats the data.

The created report can be:

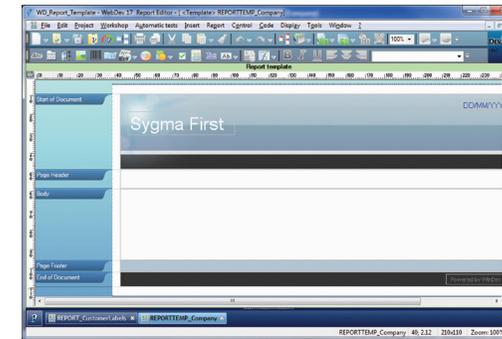
- printed on the printer of the server (for an Intranet site).
- displayed in the browser of the Web user (HTML or PDF format ...).
- emailed to the Web user (PDF format).



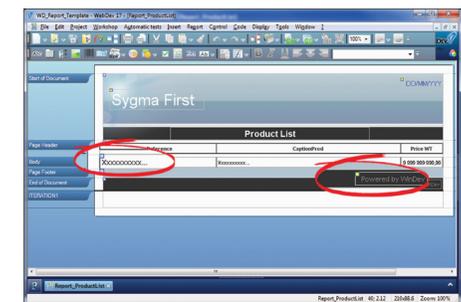
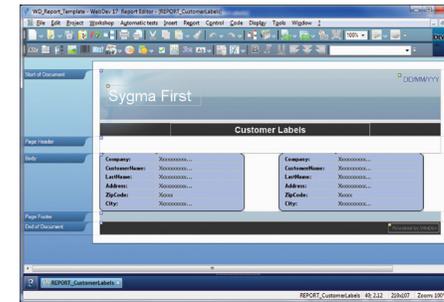
# Report templates

In most cases, the printouts use a standardized appearance and layout: date in the top right corner in a specific format, page footer with print time and file name, logo in the top left corner ...

Report templates are used to easily standardize the layout of your reports.



Defining a report template in the report editor.



Using the template in different reports.  
The elements belonging to the template are identified by a yellow square.  
The overloaded template elements are identified by a blue square.

A report template can be created:

- directly ("File .. New .. Report .. Report template").
- from the current report ("File .. Save as template").

To create a report based on a template, select the template that will be used during the report creation.

**Note:** The programming associated with the template elements can be directly performed in the template.

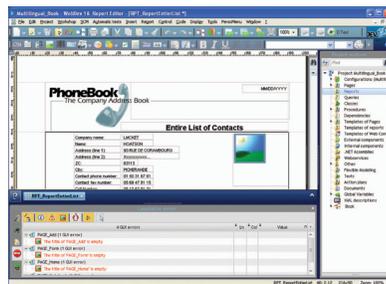
The characteristics of the elements can be dissociated from the template. For example, the position of a control can be dissociated from the template so that the control can be located somewhere else while it still benefits from the other features (code, style, ...). We talk of **inheritance**. In this case, the elements are identified by a blue square.

# Different print modes

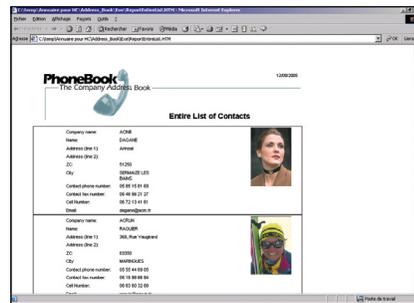
To print the documents generated by your site (invoices, quotes, ...), use the report editor to create reports.

To print these reports regardless of the configuration of the Web user's computer, we advise you to propose at least two print modes:

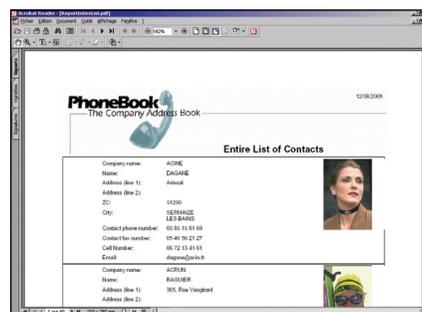
- **Print in HTML format:** The report is displayed in a new page of the site. The report can be printed via the browser "Print" button. No specific software is required.
- **Print in PDF format:** This print mode requires the presence of Acrobat Reader (or similar program) on the computer of the Web user. Printing is performed via this software.



Report in the report editor of WebDev



Printing in HTML



Printing in PDF

# Displaying and sequencing pages

In most cases, a site includes a set of pages. These pages are linked together via buttons (or links). A click performed on a button is used to display a new page.

Two questions must be asked before a new page is displayed:

1. where should the page be displayed (new browser, frame, ...) ?
2. which page should be displayed?

## Which page should be displayed?

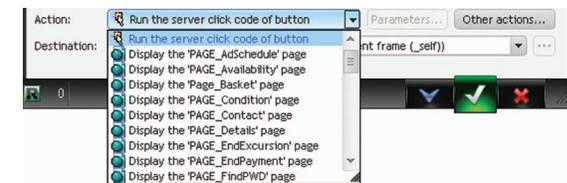
The page to display can be defined:

- in the page editor (no programming is required).
- in the code editor, in a process of the button (or link or clickable image, ...).

## Selecting the page to display in the page editor (most common method)

To define the page to display:

- Display the description window of the control ("Description" from the popup menu of the control).
- In the list of actions, select the page to display.



## Selecting the page to display by programming

The page to display must be selected by programming when:

- several pages can be displayed (an error page if the password is not entered or the next page for instance).
- a specific action must be performed on the server (calculation, read a record, ...).

### Where should the page be displayed?

A page is displayed in a specific "target": current page, new browser and so on., .

The target can be defined:

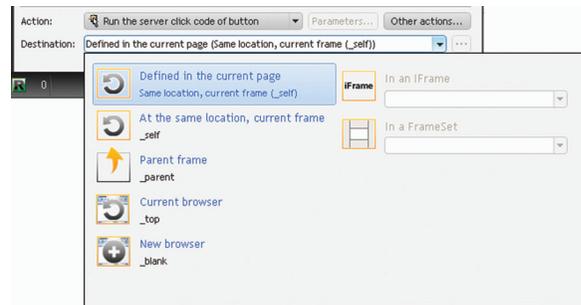
- in the page editor: no programming is required.
- in the code editor, in a process of the button (or link).

Several WLanguage functions allow you to open a new page (or frameset). The main functions are: **PageDisplay** (for pages) and **FramesetDisplay** (for framesets).

#### Selecting the target in the page editor (most common method)

In a button, to define the target of the page to display:

- Display the description of the button or link ("Description" from the popup menu of the control).
- In the list of targets, select a preset target or a frame (for a frameset).



**Note:** The preset targets are presented later in this manual.

#### Selecting the target in the code editor

When the target depends on a choice made by the Web user, this target must be selected by programming. For example, if the Web user does not enter his password, an error page is displayed in a new browser window; otherwise, the next page is displayed in the current browser window.

The **ChangeTarget** WLanguage function is used to modify the target of an action by programming. This function must be used in browser code.

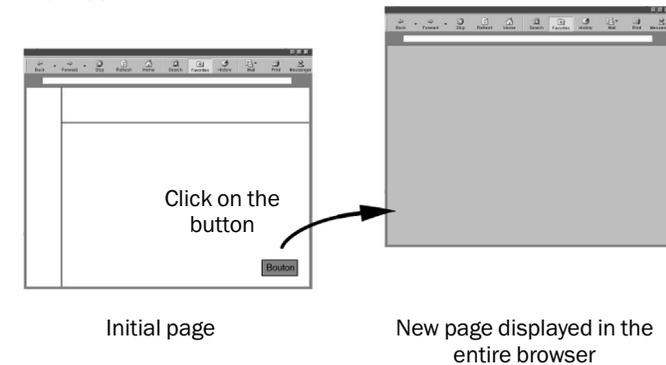
### Choosing a target

When choosing the target for a button, a link, a clickable image, a menu option, ... you can select:

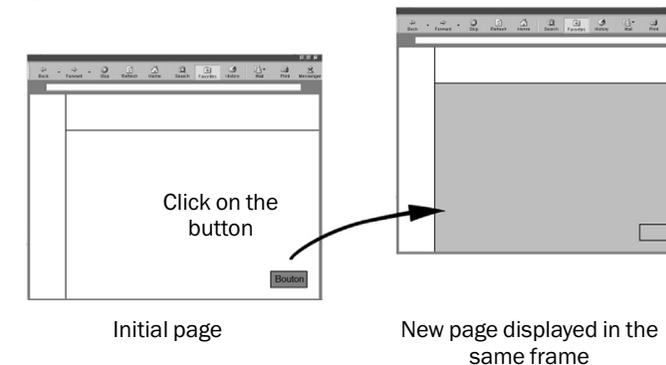
- a preset target: four preset targets are proposed:
  - Current browser (\_top)
  - Current frame (\_self)
  - Parent frame (\_parent)
  - New browser (\_blank). In this case, you will have the ability to choose the characteristics of the new browser window (with or without menu bar, status bar, ...).
- a frame in the current frameset.

**Note:** in the diagrams below, the gray out area represent the area where the page will be displayed when clicking the button.

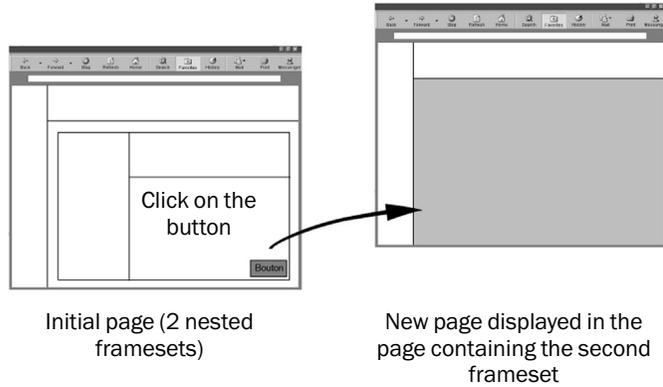
#### Current browser (\_top)



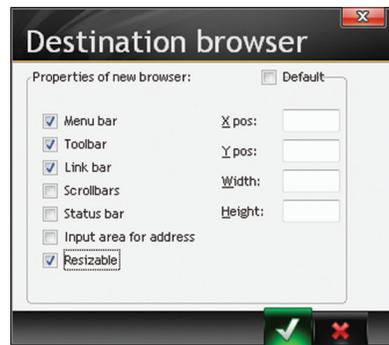
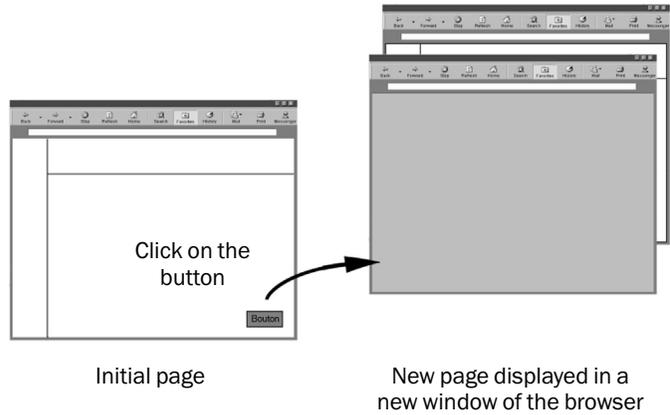
#### Current frame (\_self)



### Parent frame (\_parent)



### New browser (\_blank)

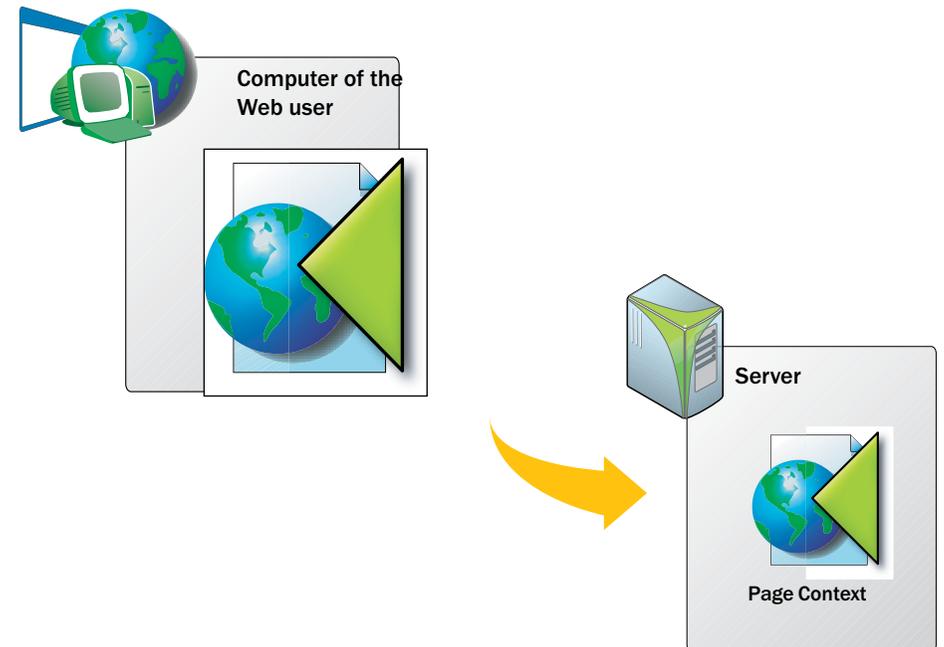


Window for defining the characteristics of the new browser

### Displaying a dynamic page: creating the associated context

We all know that WebDev can be used to create dynamic pages. The dynamic page content is defined "on the fly" at run time (that is when the page is being displayed in the browser).

A "page context" is created on the server whenever a dynamic page is displayed in the browser. This context contains the information regarding the page (value of the current variables, content of the controls, ...).



Each dynamic page (displayed on the computer of each Web user) is associated with a page context.

If the page is used to display different data (product form with browse buttons for instance), the context is modified and the page is refreshed whenever a new record is displayed.

### Displaying or refreshing a dynamic page: PageDisplay or PageRefresh?

Two WLanguage functions are used to display a dynamic page: **PageDisplay** and **PageRefresh**. These functions are used to display or to refresh a dynamic page. These two concepts are completely different.

#### Displaying a page = new page

Displaying a dynamic page means opening a "new page" in the browser. For example, opening the "Show\_Product" page used to display the selected product in a table.

**Caution:** If you "display" a page that was previously opened, the page and its context are closed and re-opened. The data found in the page is reinitialized.

#### Refreshing a page = redisplaying the page (faster)

Refreshing a dynamic page consists in redisplaying a page that is already opened in the browser. The action performed is identical to the use of the "Refresh" key of the browser. The dynamic data found in the page is refreshed according to the contexts found on the server. Only the modified data is redisplayed.

For example: In a business site, the page displaying the list of selected products is refreshed when a new product is ordered.

#### Performance tip:

To refresh the data displayed in a dynamic page, use "PageRefresh". With "PageDisplay", the page takes longer to load.

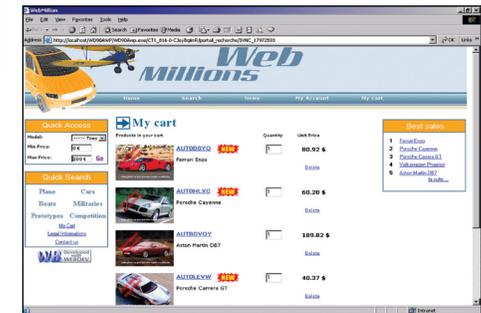
# Performing a process on several successive pages

In a business site, placing orders can be performed on several pages for instance:

- page 1: Selecting the products to order.
- Page 2: Viewing the basket.
- page 3: Validating the order and entering the customer details.



Page 1: Selecting the products



Page 2: Viewing the basket



Page 3: Entering the customer details

In this case, the write-to-file operation must **ONLY** be performed in the code of the button for validating the order: all the order elements must be stored until the final validation. Don't write into data files while they're being processed.

**Note:** You can also use a transaction but we advise you to store the order details.

# Style sheets: for simplifying the layout

The style sheets are templates used to standardize the layout of controls in the pages.

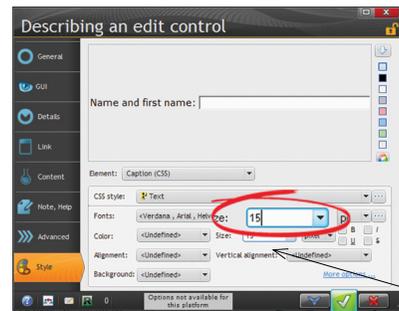
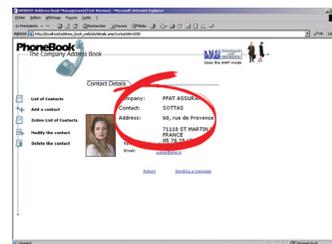
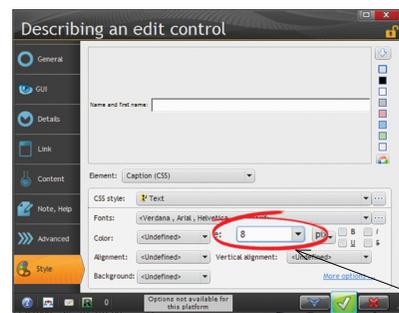
The style sheets are used to define:

- the size and layout of your texts, the font,
- the alignment of your paragraphs, the paragraph indents, ...
- the style of the text (bold or italic), its color, ...

And much more.

When a characteristic of the style is modified, all the controls associated with this style are updated. **This gives you the ability to modify the style book of a site by only making a few modifications to the styles used**

Example of style sheets use: increasing the size from 8 to 15:



**Note:** The "Custom CSS" tab (in the advanced options) enables you to directly enter CSS code. Then, this code will be added to the style sheet.

# Centered or adapted site?

The page layout is one of the most important elements when building a site. To optimize the page layout, the resolution defined on the screen of the Web user must be taken into account.

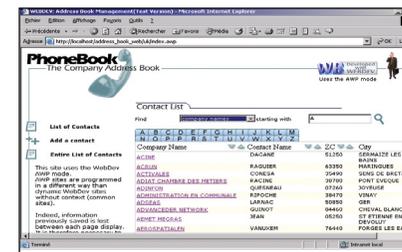
For example, if your site is created for a resolution of 1280 x 1024 pixels, several scroll-bars will be displayed on the screens that use a resolution of 1024 x 768 pixels.

**Overall, at the time this manual is printed, we advise you to develop sites optimized for a 1024 x 768 screen resolution.**

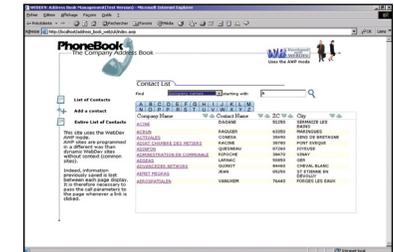
Two types of page layout can be used:

- **centered site:**

The pages are centered in the browser. If the resolution of the Web user is greater than the optimal resolution of the site, some white margins are displayed on both sides of the page.



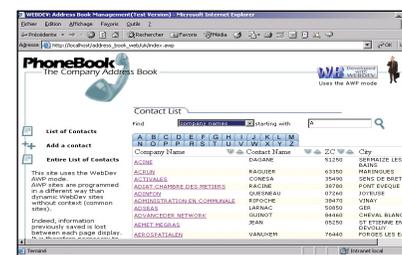
Resolution: 800 x 600



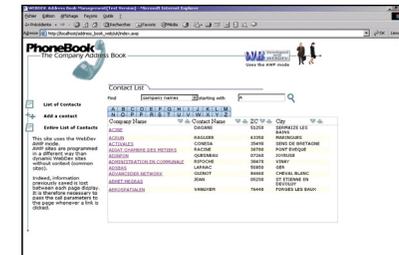
Resolution: 1024 x 768:  
margins come up on each side of the page

- **adapted site:**

The pages are displayed as usual in the browser. If the resolution of the Web user is greater than the optimal resolution of the site, a white margin is displayed on the right side of the pages.



Resolution: 800 x 600



Resolution: 1024 x 768:  
the page remains "stuck" to the left

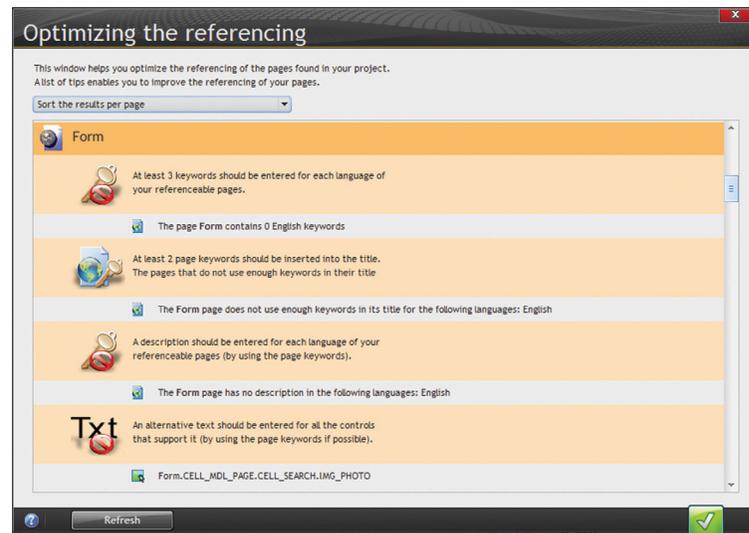
**Note:** The type of page layout (centered or adapted) configured in a page template will be automatically used by the pages associated with this template.

# Referencing a site

To visit your site, the Web users must be able to find it. To do so, your site must be offered when keywords corresponding to your site are typed by the Web user in a search engine.

To help with the referencing of your site, WebDev proposes:

- a referencing wizard. This wizard lists all the optimizations that can be performed to improve the referencing of your site. To start the wizard, select "Project .. Referencing .. Optimize the referencing").



- the ability to enter a description and keywords on each page of your site.
- the ability to include a home page. This home page allows you to enter the description of your site as well as the keywords corresponding to your site.

You have the ability to include a home page in your site:

- when creating the project (in the wizard for project creation).
- when developing the project ("File .. New .. Page" option, "Page" then "Home" option).

Several templates of home pages are proposed.

**Note:** the referencing uses techniques set by search engines (Google, Bing, etc.) that are not explicitly defined and that change very quickly. Don't hesitate to read books or to visit sites dedicated to this subject.

## Referencing a site in practice

### 1 Referencing principles

In order to visit your site, Web users must be able to find it. To do so, your site must be offered when keywords corresponding to your site are typed by the Web user in a search engine.

To help you, WebDev proposes:

- **A referencing of 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,** specifying for each page the improvements that can be performed to optimize the referencing.
- The ability 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.

**Note:** To optimize the referencing of your pages in search engines, we recommend that you use AWP pages. To reference a dynamic site, use:

- a home page.
- a section of the site in static mode. The static site will be used to perform the referencing and to start the dynamic site.
- a section of the site in AWP mode. The AWP site will be used to perform the referencing and to start the dynamic site.

### 2 Referencing (or not) a static or AWP page of a site

To reference a static or AWP page of your site:

1. Display the description window of the page.
2. In the "Details" tab, a specific area is used to manage the referencing of the page.

You can:

- Avoid referencing the current page.
- Reference the current page by associating it with expressions and keywords ("Edit the expressions and the keywords").

### 3 Referencing a page and entering its keywords

To reference a static or AWP page of your site:

1. Display the description window of the page.
2. In the "Details" tab, a specific area is used to manage the referencing of the page. In this area, click the "Edit the expressions and the keywords" button.
3. **In the "Description" tab of the referencing window,** enter the description of the page. In most cases, this description appears when displaying the result of a search performed by a search engine.

Some tips:

- Use short sentences, limit the number of words (up to 200 characters)
  - This description must entice the Web users to display the page.
- Note:** This description can be modified dynamically by **..Description**.

4. In the "Keyword" tab of the referencing window, enter the keywords associated with the page. These keywords (or expressions) will be the keywords used to reference the page: when the Web user enters one of these keywords, the search engine will propose the corresponding page. The keywords are entered as follows: enter the first keyword then press [ENTER] to enter the next one.

Some tips:

- The first three keywords are the most important ones. Use common keywords and pertinent keywords.
  - The keywords must not be repeated.
  - Use variations (singular/plural, noun/verb, ...).  
**Note:** These keywords can be dynamically modified by **..Keywords**
5. Validate the referencing window and the description window of the page.

## 4 Using the referencing wizard

To help you optimize the referencing of your sites and pages, WebDev proposes a referencing wizard. This wizard can be used for a specific page or for all the pages of your project. This wizard analyzes the composition of the site pages and indicates the possible improvements for maximizing the positioning of the pages in the search engines.

Some examples of optimizations detected by the wizard:

- The title of the page must be specified.
- At least three keywords must be specified for the page.
- The description of the page must be specified.
- The alternative text must be entered for all the controls that propose it (by using the keywords defined for the page if possible) ...

**To start the wizard for referencing a page:**

1. Display the requested page in the editor.
2. Select "Page .. Optimize the referencing of the page".
3. The referencing wizard starts. Double-click a suggestion to perform the corresponding optimization. The "Refresh" button is used to update the list of suggestions.

**To start the referencing wizard for all the pages of a site:**

1. Select "Project .. Referencing .. Optimize the referencing".
2. The referencing wizard starts. Double-click a suggestion to perform the corresponding optimization. The "Refresh" button is used to update the list of suggestions. The results can be sorted by page or by type of advice.

## 5 Including a home page in your site

A home page can be included in your site when creating a new page ("File .. New .. Page", "Blank/Templates" tab, "Home" option).

If no home page is associated with your project, you can:

- choose a home page among the static pages of your site
- choose a preset home page.

## 6 Registering your site with search engines

The referencing of your site is performed on the search engine directly.

**Note:** the referencing of a site may sometime carry a fee.

Two types of search engines are available:

- **the "Automatic" search engines:**

These search engines are based on automatic programs for analyzing the content.

**How can I be referenced?**

A "Reference your site" link is often available for this type of search engine. Then, all you have to do is specify the address of the site and the email of the Webmaster. The site will be automatically analyzed (according to the keywords and to the content of your site) and referenced.

- **the "Directory" search engines:**

These search engines reference sites that have been previously checked by human beings.

**How can I be referenced?**

A "Reference your site" link is often available for this type of search engine. In most cases, the procedure for referencing your site is as follows:

1. Choose a category for the site (leisure, culture, ...).
2. Fill out a questionnaire about the site to reference: Internet address of the site, email of the Webmaster, description of the site, ...
3. Once your questionnaire has been sent, your site will be evaluated by a person from the company that manages the directory. This person will list your site if its content appears to be worthwhile.

PART 3

Development environment



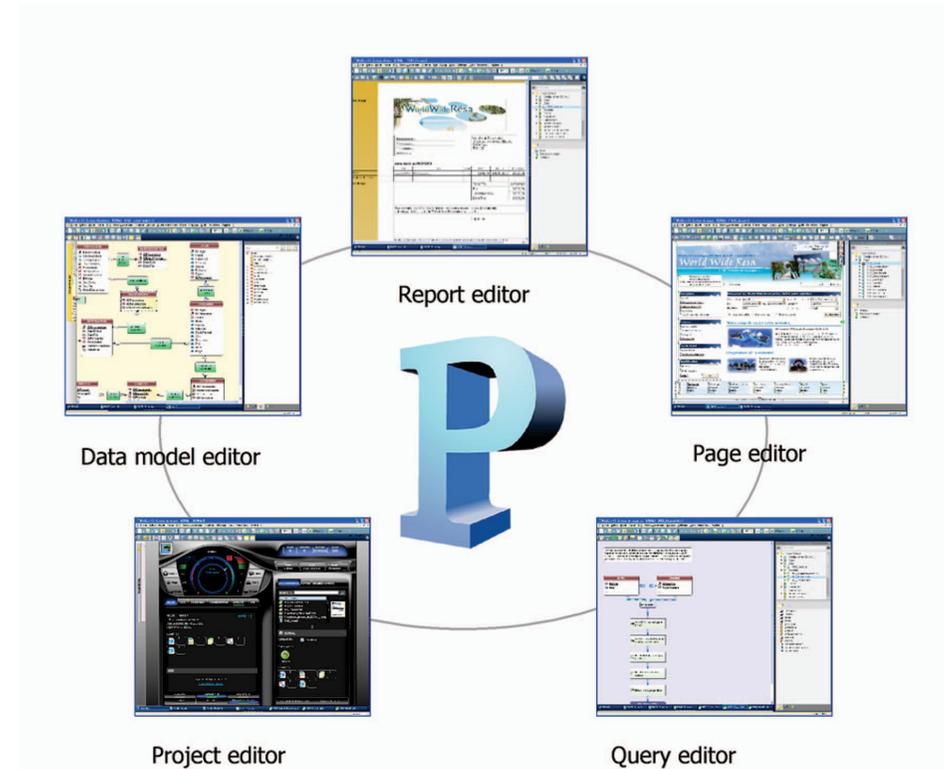
DEVELOP 10 TIMES FASTER

# WebDev editors

To handle a project, WebDev proposes several integrated editors:

- project editor.
- data model editor.
- UML editor.
- query editor.
- page editor.
- report editor.
- ...

These editors are used to easily handle the different project elements.



# User-friendly editors in practice

## 1 Introduction

WebDev is designed around editors, adapted to each developer requirement, unified in a single environment:

- project editor.
- data model editor.
- UML editor.
- query editor.
- page editor

- report editor.
- source code editor, debugger and compiler.
- modeling editor.
- document editor.

These editors will be used to create the different elements (pages, reports, databases, programs, ...) handled by the site.

## 2 The different editors

### 2.1 The project editor

The project editor has two main purposes:

- it is used to quickly view all the elements found in a project (pages, reports and queries).
- it is used to view and create the sequences between the pages and the reports.

Some characteristics of the project editor:

- WYSIWYG editor ("What You See Is What You Get"). All the project elements can be viewed directly.
- The project is managed via a comprehensive dashboard.
- Creation and visualization of the sequences between several elements (pages, reports, ...) of your project via the project graph.
- Interaction between the different panes and the project elements.

### 2.2 Data model editor

The data model editor is used to describe the characteristics and the structure of the data files.

The data model editor supports the Merise method (CDM and LDM).

The data model editor allows two types of database descriptions:

- description of the analysis linked to the project (also called Logical Data Model (LDM))
- description of the Conceptual Data Model (CDM) then automatic generation of the analysis.

Let's see some characteristics of the data model editor:

- WYSIWYG editor ("What You See Is What You Get"). You can view directly all the files and links of your project.
- Creation and description of the files and items.
- Use of an analysis in WinDev 5.5 format.
- Automatic formatting of the analysis links.
- Information about the links via tooltips.
- Simplified retrieval of the description of an external database (SQL Server, Oracle, AS/400, ...).
- Description independent of the data files.
- Encryption of the data files.
- Automatic modification of the data files when the structure of the files is modified.
- Management of the referential integrity.
- Automatic generation of the analysis.
- Zoom in the data model editor.
- Interaction between the different panes and the analysis elements.

### 2.3 UML editor

The UML language is a graphical language allowing you to:

- represent the information system studied as objects.
- generate the object structure of the site (skeleton of application as object classes) corresponding to the information system studied.

The UML editor is mainly used to:

- describe one or more UML diagrams.
- build a UML model by reverse engineering of your project.

See "The UML model", page 120 for more details.

Let's see some characteristics of the UML editor:

- Creation and description of UML diagrams.
- Automatic formatting of the links.
- Automatic generation of a class (or set of classes) from a class diagram.
- Zoom in the UML editor.
- Insertion of comments into a UML model.
- Interaction between the different panes and the elements of the UML model.

### 2.4 Query editor

The query editor is used to automatically create queries based on the data files. No programming hassle: pages, tables, combo boxes, reports, ... can be based on queries.

A wizard helps you create a query: all you have to do is choose the items to include and enter the selection conditions via the wizard. The query is automatically generated (in optimized SQL code) and the test of the query can be run immediately.

You will find more information about the queries in the "Reports and Queries Guide".

Let's see some characteristics of the query editor:

- WYSIWYG editor ("What You See Is What You Get"). You can view directly the query and its result.
- Simplified creation of a query via a wizard.
- Automatic generation of the SQL code of each query.
- Ability to immediately run the query test.
- Zoom in the query editor.

### 2.5 Page editor

The page editor is used to describe the characteristics of the user interface of your project.

Several templates, skins and types of pages are proposed. They help you make your pages more ergonomic and make your sites more user friendly.

Let's see some characteristics of the page editor:

- WYSIWYG editor ("What You See Is What You Get"). You can view directly your page as it will appear to the user.

- Creation and description of the pages and controls.
- "Drag and Drop" allowing you to copy, paste or move controls from a page to another one.
- Page templates and preset control styles.
- Presence of several icon catalogs used to associate the images with the controls.
- The real-time interface checker is used to simplify the positioning of controls.
- Ability to enter the captions of controls in the work area of the editor.
- Management of the context-sensitive help in the pages.
- Zoom in the page editor.
- Interaction between the different panes and the elements of the editor.

### 2.6 Report editor

The report editor is used to easily create printed reports.

You will find more information about the reports in the "Reports and Queries Guide".

Let's see some characteristics of the report editor:

- WYSIWYG editor ("What You See Is What You Get"). You can view directly the reports as they will be printed.
- Simplified creation of report without writing a single code line.
- "Drag and Drop" used to copy, paste or move the controls from a report to another one.
- Skin templates of reports and preset control styles.
- The real-time interface checker is used to simplify the positioning of controls.
- Ability to use a form as report background.
- Creation of multi-column labels.
- Ability to edit a report in HTML format (to publish it on the Internet for example) or in RTF format (to use it in a word processor for example).
- Zoom in the report editor.

## 2.7 Code editor

The code editor is used to enter all the processes in WLanguage (the programming language included in WebDev). It is used to enter the source code:

- of controls,
- of pages,
- of reports
- of local and global procedures,
- of classes and methods, ...

Let's see some characteristics of the code editor:

- Automatic formatting of the information entered.
- Automatic completion.
- Glossary of functions.
- Immediate detection of typos and help for correction.

- Incremental compilation.
- Preview of the different processes of a window, control or report found in the project.
- Insertion of processes specific to the use of the mouse or keyboard.
- Management of the history of code modifications with ability to rollback.
- Ability to open several code windows (to perform code comparisons for example).
- Zoom in the code editor.
- Interaction between the different panes and the elements of the editor.

We recommend you read the "WLanguage" guide for more details.

# Project dashboard

The project dashboard is an essential element for managing WebDev projects. The project dashboard gives an overall view of the progress status of a project.

The dashboard proposes two modes for viewing the project:

- The Developer mode,
- The Project Manager mode.

In Developer mode, the project dashboard includes:

- lights: Everything is OK when the lights are green. The red lights indicate a possible problem.
- lists of elements, giving quick access to the main options of the project.
- counters, used to manage the new features, the requests, ...



In Project Manager mode, the dashboard is used to get graphic information about the progress status of the project and about the quality of the project. This mode is recommended when using the Project Monitoring Center with requirement management.

# WebDev, WinDev, WinDev Mobile: 100% compatible

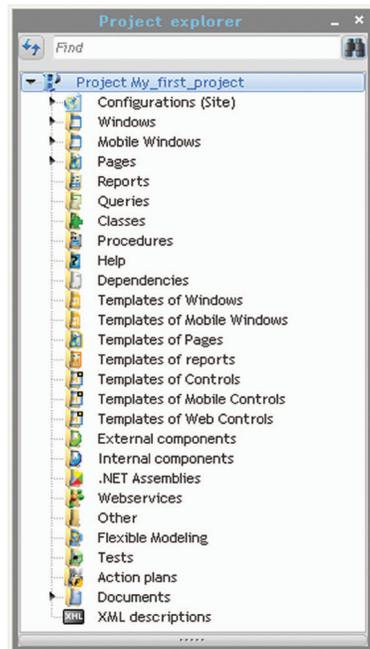
The created projects are often multi-target projects.

For example, for an ERP system intended to operate in Windows, it is very likely that beside the main application, which will be the backbone of the solution, there will be sales people equipped with PDAs or Smartphones, shops that will use mobile terminals to manage inventory and that Intranet and Internet sites will be implemented.

**All the elements, except for the UI (pages and windows), are 100% compatible and sharable among the WinDev, WebDev and WinDev Mobile projects.**

Indeed, the sets of procedures or the classes can be shared among several projects for example.

Regardless of the product used to create a project, it can be opened by the other products.



When a project is opened in a product other than the one where it was created, a wizard is displayed, allowing you to create a project configuration specific to the product used.

For example, if a WinDev project is opened by WebDev, you will have the ability to create a project configuration named "Site", used to group all the elements required by the WebDev site.

You now have the ability to view the elements of each target from each environment. A project in WebDev displays the thumbnails of the WinDev and WinDev Mobile windows for instance. Clicking a WinDev window from the WebDev project editor opens the WinDev window (WinDev must be installed on the computer).

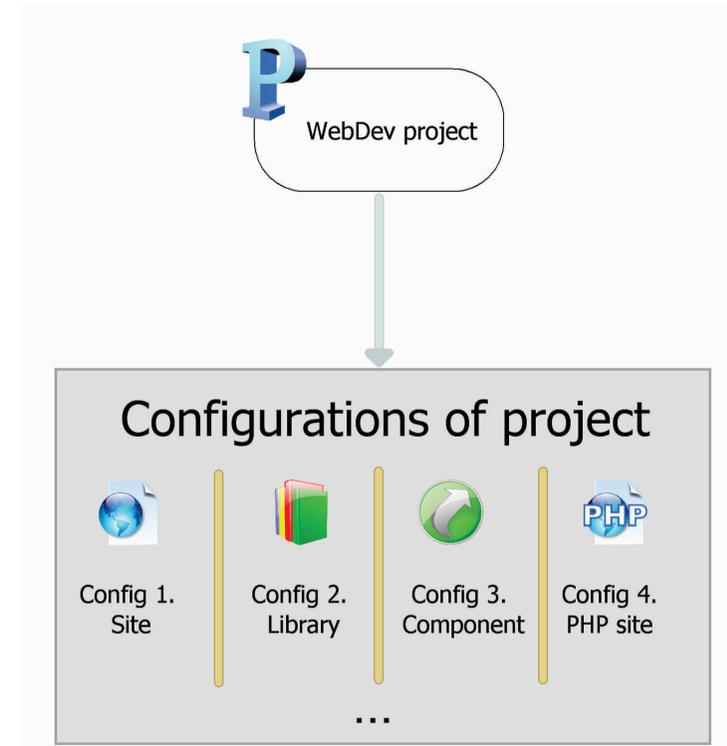
# Project configuration

The project configurations are used to create several "targets" from the same project.

From the same project, you have the ability to create:

- sites that do not contain the same elements, with different names, ...
- different components,
- a Webservice.

You can work on a specific configuration at any time: the elements that do not belong to this configuration are grayed in the project editor.



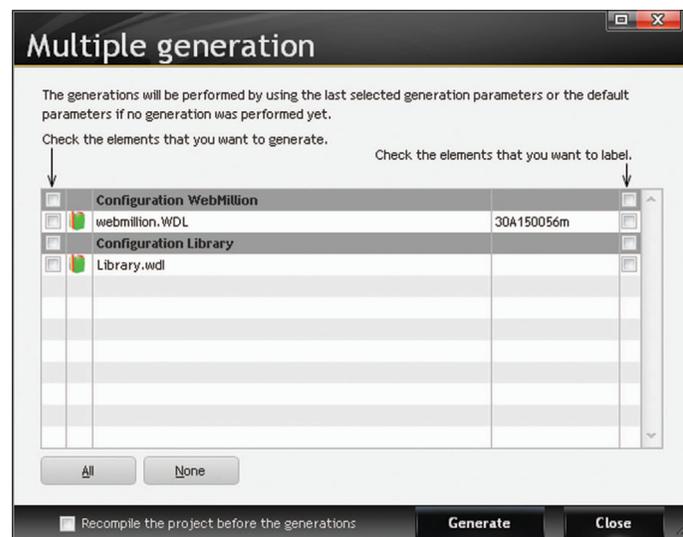
The multiple generation allows you to generate all the project configurations (or some of them) in a single operation.

# Multiple generation

The project configurations are used to easily define the different "targets" of your project. Several sites, several libraries or several components can be defined for the same project.

To generate the result of each configuration, you can select each configuration one by one and generate the corresponding element.

Another faster method is available: **the multiple generation**. The configurations to generate are selected in a single operation and the result is instant.

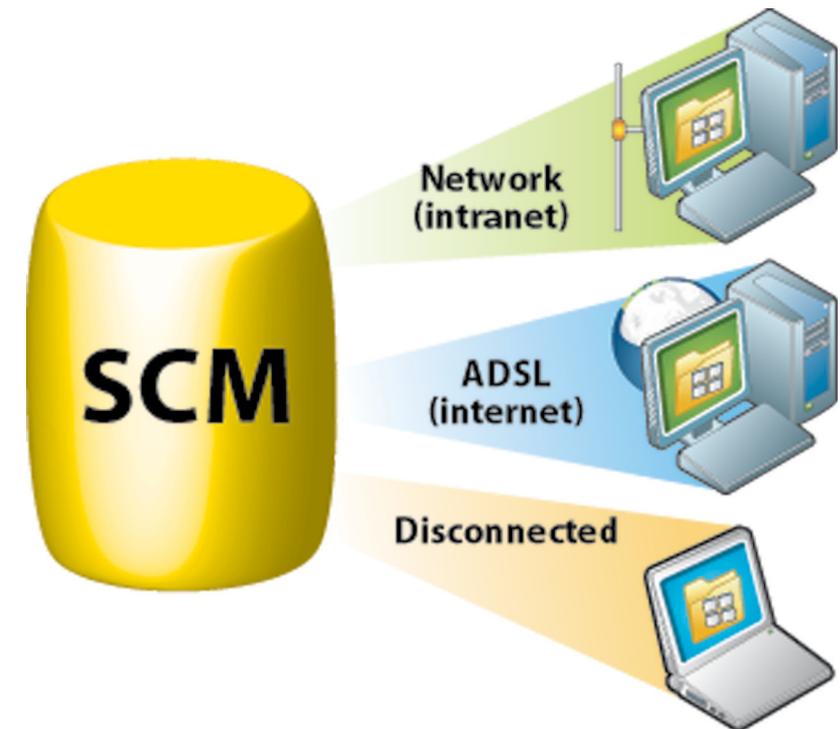


To start a multiple generation, select "Workshop .. Multiple generation".

# Source Code Manager (SCM)

## Overview

To simplify teamwork, a Source Code Manager is available in WinDev. This Source Code Manager allows several developers to work together on the same project at the same time and to easily share elements among several projects.

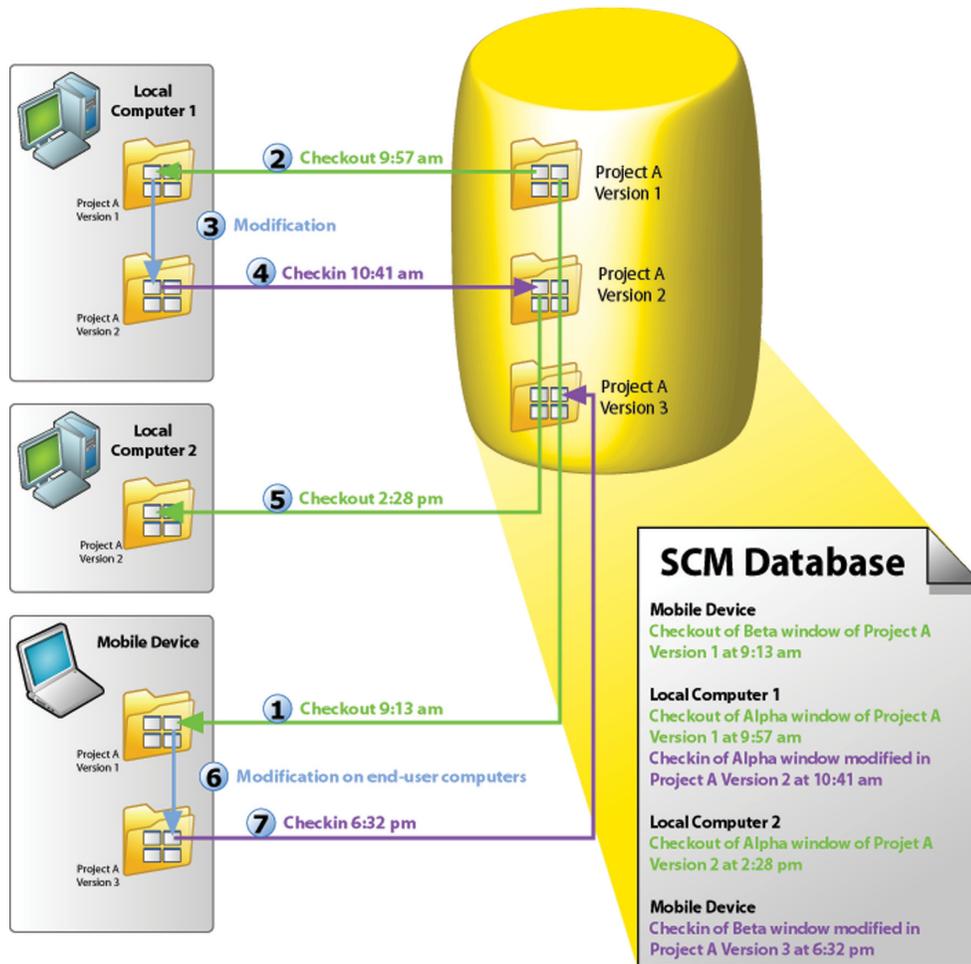


The elements found in the SCM can be shared:

- via a network.
- via Internet.
- in offline mode. In this case, the elements that require specific attention will be checked out from SCM when the laptop is connected to the main system for example.

## Operating mode of the Source Code Manager

The following example presents the Source Code Manager:



If a project element (page, report, ...) is checked out, this element cannot be checked out twice.

Once the checked-out elements are modified, these elements must be checked back in order for the modifications to be taken into account by the source project. Indeed, the SCM database stores a history of all the project elements since their creation.

Whenever an element is checked back in, the version number of the source project is incremented by 1.

## Sharing a project in practice

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 (queries, classes ...).

To share a project, you have the ability to use:

- the source code manager.
- the developer groupware. It is kept for backward compatibility with the earlier versions. It will not be presented in details in this chapter. See the online help (keyword: "Developer groupware") for more details.

## 3 The Source Code Manager

### 3.1 Overview

WebDev innovates in the area of teamwork management with the Source Code Manager (also called SCM).

Fully integrated in the environment, the Source Code Manager (SCM) is used to:

- improve teamwork,
- store the history of modifications and versions,
- save the source code of the development team.

Sized for team between 1 and 100 developers, SCM facilitates and standardizes collaboration between developers (even when developing alone, the SCM is useful because it contains the history of your applications).

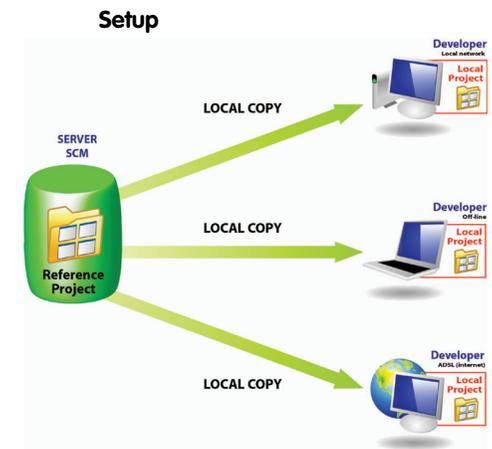
**SCM uses a source database:** procedures, classes, pages, reports, components, analyses, ...

This database can be installed on a server (in HyperFileSQL Classic or HyperFileSQL Client/Server mode) or on a network computer in a shared directory.

SCM can be run locally or remotely, via Internet or via an HTTP or HTTPS connection. This feature gives you the ability to work on a project from an agency or from a remote site without losing the modifications.

The SCM can also be used in offline mode (in a train, on a plane,...).

### 3.2 Principle



All the project elements are saved in the SCM database (on the server). This operation is performed when creating the project or when importing an existing project into the Source Code Manager.

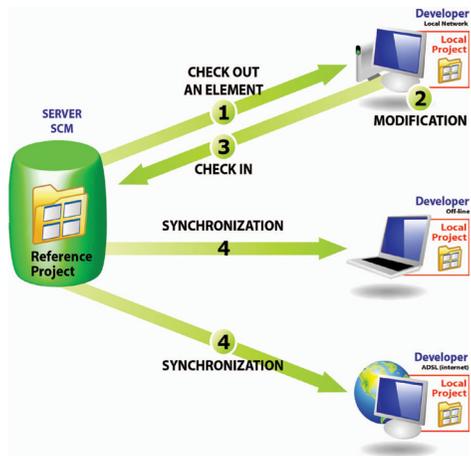
Each developer who uses the Source Code Manager gets a local copy of the project.

#### Use

To work on a project element (page, report, ...), the developer must check out the element from the SCM database, modify it and check the element back in.

If the element is already checked out, the developer can ask the person who performed the check-out to check the element back in (via the messaging software).

To benefit from the modifications performed, the other developers must synchronize their local project with the reference project (found in the SCM database).



### Advices

The source code of your applications is essential. This source code must be handled with great care! Tips for configuring the server that will host your source code:

- Use a dedicated server with a comfortable size disk (200 GB at least)
- Use the Source Code Manager (SCM) in Client/Server mode.
- The hard disks may encounter physical problems: if possible, use a RAID 1 system on your server (several disks storing the same information)
- Use a UPS to protect the power supply of your server.
- Perform regular backups of the SCM database (at least once a week)
- Place the server in a "secured" area and use a firewall.

## 4 Creating a database for SCM

### 4.1 Overview

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.

The operating system and the file system on which the SCM database is installed must support files exceeding 4 GB.

The SCM database can be created:

- in a shared directory on the network: all the SCM users must have all the rights on this directory. The SCM database will be in HyperFileSQL Classic format. To do so, specify the network directory where the SCM database must be created.
- on a HyperFileSQL Client/Server server: the SCM database will be in HyperFileSQL Client/Server format. In this case, you must specify:
  - the server
  - the database
  - the port used
  - The name and password of an administrator of the HyperFileSQL database.

### 4.2 When should a SCM database be created?

The SCM database must be created once only.

WebDev allows you to create this SCM database at different times:

- when installing WebDev.
- when creating a project that uses SCM.
- when importing a project into the source code manager.
- in the SCM administrator directly.

Once the SCM database is created, all the shared WebDev projects can be imported into this SCM database.

### 4.3 Backups

We recommend you perform backups of the SCM database on a regular basis. These backups can be performed via the SCM administrator.

## 5 Configuring the project in order to work with SCM

Some operations are required before an existing project can be used by the Source Code Manager.

### 5.1 Adding a project into SCM

To add a local project into the SCM:

1. Open your project in WebDev.
2. In the project editor, select "SCM .. Add the entire project".
3. In the wizard, select the SCM database to use.

**Note:** If no SCM database was created yet, click the "Create a database" button.

Specify whether you want to use:

- a source database found on a network share (SCM database in HyperFileSQL Classic format). Specify the directory of the SCM database (network computer or shared directory).
- a SCM database found on a HyperFileSQL Client/Server server. The SCM database will be in Hy-

perFileSQL Client/Server format. In this case, you must specify:

- the server
  - the database
  - the port used
  - The name and password of an administrator of the HyperFileSQL database.
4. Then, click the "Create" button.

### 5.2 Sharing resources

A first project was imported into the Source Code Manager. This project contains elements shared with other projects (classes, pages, procedures, style sheets, ...).

The share is an important concept of SCM. Several methods are available to perform the share. See the online help for more details.

## 6 Working with the SCM

### 6.1 Project options affecting the SCM

Several options are used to configure a project handled by the Source Code Manager. These options are grouped in the "SCM" tab of the project description ("Project .. Project description"). These options are checked by default.

- **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 offers 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, ...**

This option enables you to display when creating a library, ... the list of checked-out elements in order to check them back in and to get the latest version of these elements. Therefore, the component or the library can contain the most updated elements.

By default, the generation of the library, ... is performed with the project elements currently found on the local computer.

- **Check out/Check in the project automatically:** This option is used to automatically manage the "project file". If this option is checked, the project file is checked out only if the action performed requires it. Once the action has been performed on the project, the project file is automatically checked back in. This option allows you to not have the "Master/Guest" management on the project. This option can also be enabled via "SCM .. Project management .. Manage the project checkout automatically".

## 6.2 Checking out an element

### The different check-out modes

The SCM proposes two modes for checking out the project elements:

- **the standard mode:** if you display a SCM element that has not been checked out, a dialog box indicates that this element must be checked out before it can be modified. The element can be checked out immediately (checkout button found in the panel).
- **the automatic mode:** if you try to modify a SCM element that has not been checked out, SCM automatically offers to check this element out. Once the check-out is validated, the element can be modified.  
Note: this mode is not recommended when the SCM is used with a slow Internet connection.

To change the check-out mode, select "Tools .. Options .. General options of WebDev", then select the "General" tab. The "Checking out elements during the first modification" option is used to switch all the projects opened from the current product to the automatic mode

### Opening a project element to modify its characteristics

To modify the characteristics of a project element managed by the SCM:

1. Check out the element from the Source Code Manager.
2. Select the check-out mode of the element. The check-out mode can be:
  - **exclusive:** nobody else will be able to check out this element until it is checked back in. The element can be checked out for test only.
  - **for test:** the element can be modified but the modifications will not be checked back in.
  - **multiple:** the element can also 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.
3. Validate. The element is opened. The title bar indicates that the element is checked out.

## 6.3 Checking in an element

The elements checked out from the Source Code Manager are bordered by a red line in the project editor.

To check in an element, all you have to do is select "Check in" from the popup menu of the element (in the project graph or in the "Project explorer" pane).

When checking in an element, a screen allows you to perform the following actions before the element is checked back in:

- find out the modifications performed
- compare the element found in SCM database with the local element (checked out)
- access the history of the element in the SCM database.

You have the ability to check in the modifications made to the element while keeping the element checked out ("Keep the element checked out").

## 6.4 Management modes of the project

Two management modes of the project are available with SCM:

- Management of the project in Master/Guest mode
- Management of the project in automatic mode

### Master and guest

The Source Code Manager distinguishes between 2 types of users:

- the master: the master is the user who initially stored the project in the Source Code Manager.
- the guests: the guests are the developers who handle the project from the Source Code Manager.

There is no need to be connected in master mode on a project. The "Master" mode is required to:

- modify the characteristics of the project and check in these modifications into the SCM database.
- check in all the elements to create the setup program of the site.

**To switch from the master mode to the guest mode,** select "SCM .. Project management .. Become guest on the project (and check all in)". In this case, the Source Code Manager proposes to check in all the project elements (including the .WWP file).

**To switch from the guest mode to the master mode,** select "SCM .. Project management .. Become master on the project".

### Caution: Modification of the project options:

All the users of the project (master or guests) can modify the characteristics of the project (first page of the project, animation, programming charter, ...). These modifications will have to be checked back in into the Source Code Manager by the master of the project.

The modifications made by a guest will be lost when the project is updated from the SCM database.

### Automatic mode

With the automatic mode, the project file is checked out only if the action performed requires it

(regardless of the user). Once the action has been performed on the project, the project file is automatically checked back in.

The automatic mode avoids you managing the "Master/Guest" mode on the project.

## 7 Working in offline mode with the SCM

The Source Code Manager enables you to easily work in offline mode (or mobile 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 you disconnect, select "SCM .. Remote work .. Disconnect for a mobile use". Before the

disconnection, we advise you to check out the different elements that will be modified (therefore, these elements will be "already checked out" for the other users). You can work on your project locally. The different project elements can be handled directly.

- when you reconnect, select "SCM .. Remote work .. Reconnect and synchronize". Then, all you have to do is check in the modified elements.

See the online help for more details.

## 8 SCM administrator

The SCM administrator enables you to handle the different projects included in the Source Code Manager.

It allows you to:

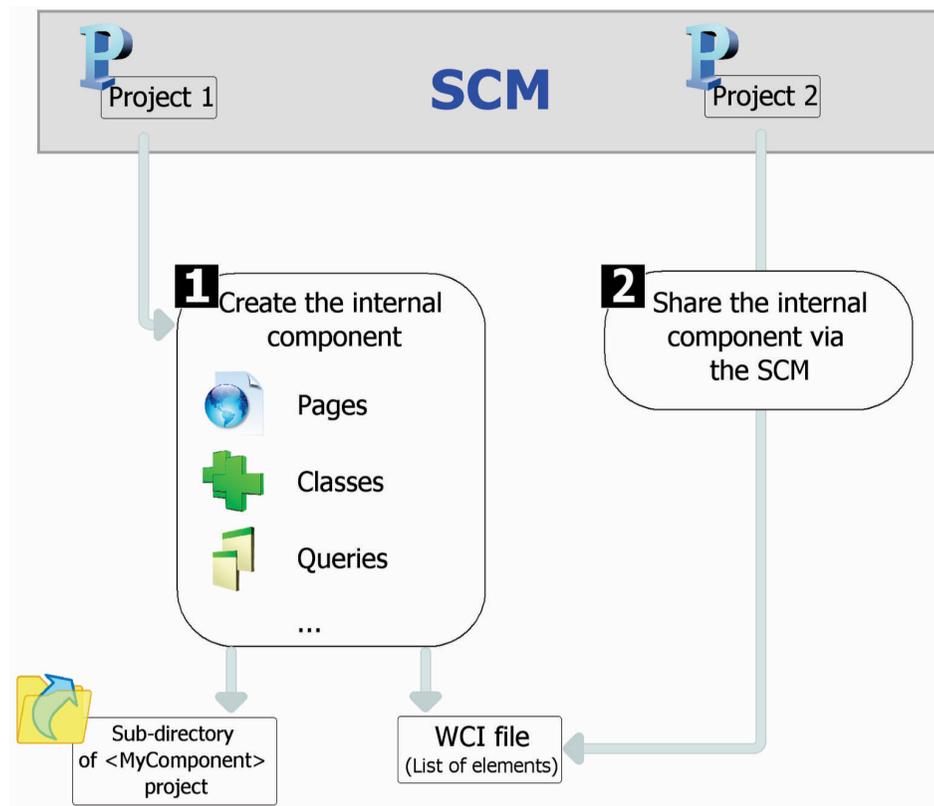
- manage the SCM databases (creation, connection to a SCM database)
- manage the branches
- manage the files and the directories found in a project of the SCM database (add, delete, rename, ... files and directories)

- manage the different files of the scm database (check-in, check-out, share, ...).
- run some tools (options, maintenance, ...).
- restore a project version.
- see the history and the differences of versions.
- cancel the check-outs (in administrator mode)
- clear a database, save it, restore it
- add some files of any type into the database (.doc, .xls, .pdf, ...)

# Internal component

An internal component is used to group several project elements. This grouping allows you to:

- Organize a project: you have the ability to create internal components to group the project elements by feature for example.
- Share the elements among several projects via the SCM (Source Code Manager).



The elements found in an internal component can be private or public:

- The private elements can be handled by the other elements of the component.
- The public elements can be handled by the elements of the project that uses the internal component.

## Internal component in practice

### 1 Overview

An internal component is used to group several project elements. This grouping allows you to:

- **Organize a project:** you have the ability to create internal components to group the project elements (by feature for example).
- **Share elements among several projects:** The elements found in an internal component can be shared among several projects via SCM. See "Sharing the internal components (via the SCM)", page 100 for more details.

One of the benefits of the internal component compared to a standard component is that the internal

component can be debugged directly from the project that uses it.

**Difference with a standard component:** An internal component allows you to include all the component elements in the interface of the project containing the internal component. All the "public" elements of the internal component can be handled directly in the editor.

When using a standard component, the "public" elements of the component cannot be handled directly. To modify the standard component, the corresponding project must be opened.

### 2 Creating an internal component

#### 2.1 The different steps

To create an internal component:

1. Select "Workshop .. Internal component .. Create an internal component". The internal component creation wizard starts.

2. Specify the characteristics of the internal component:

- its name. The name of the internal component will be used for the WCI file corresponding to the description of the internal component. This name will also be used to create a sub-directory in your project containing all the elements of the internal component.
- its caption.
- its description.

3. Indicate the elements that must be included in the internal component. An internal component can contain all the types of elements found in a project: pages, reports, templates, ...

**Note:** If the elements included in the internal component contain links to the elements found in the project or in another internal component, a compilation error will indicate that a project element cannot be accessed from an internal component.

4. Specify the elements of the internal component that will be directly accessible in the code and in the preset actions of the project that hosts the internal component.

**Note:** The accessible (or "public") elements of the internal component will be automatically proposed by the completion. They can be used by the elements found in the project or from another internal component. The inaccessible elements (or "private" elements) can only be used by another element of the internal component (the completion will propose these "private" elements only from the elements of an internal component).

5. Specify the management mode for the data and runtime contexts. Three management modes of data and runtime contexts are available:

- **Full share:** The internal component accesses the data files of the project. The internal component and the project use the same runtime contexts. This mode corresponds to the default mode if the component uses no analysis.
- **Data share only (advanced mode):** The internal component accesses the data files of the project. The internal component and the project use different runtime contexts. This mode is reserved to specific cases.
- **Standalone:** The internal component accesses its own data files. The internal component and the project use different runtime contexts. This mode corresponds to the default mode if the component uses an analysis. In this case, the analysis used by the internal component must be specified. You also have

the ability to create a new analysis directly. This analysis will be associated with the internal component.

6. Validate the creation of the internal component. You will have the ability to:

- Modify the characteristics of the internal component via the description window of the internal component.
- Handle the internal component and its elements.

**Tip:** An internal component contains no global variables declaration code. A set of procedures can be used to initialize the internal component.

### 2.2 Internal component and analysis: case of full autonomy

An internal component can be linked to its own analysis. In this case, the project that includes the internal component can have several analyses:

- the analysis of the project
- the analysis of the internal components. This analysis is defined when creating the internal component. It can also be selected from the description window of the internal component.

The elements generated by RAD for the internal component will be generated in the directory of the internal component. If the analysis of the internal component is modified, a new generation by RAD will automatically propose to generate the elements corresponding to the modifications.

## 3 Sharing the internal components (via the SCM)

The internal components can be shared among the projects via SCM.

To share an internal component via SCM, the project containing the internal component must be found in the SCM. You can:

- create the internal component from a project found in the SCM. The internal component will be

automatically found in the SCM.

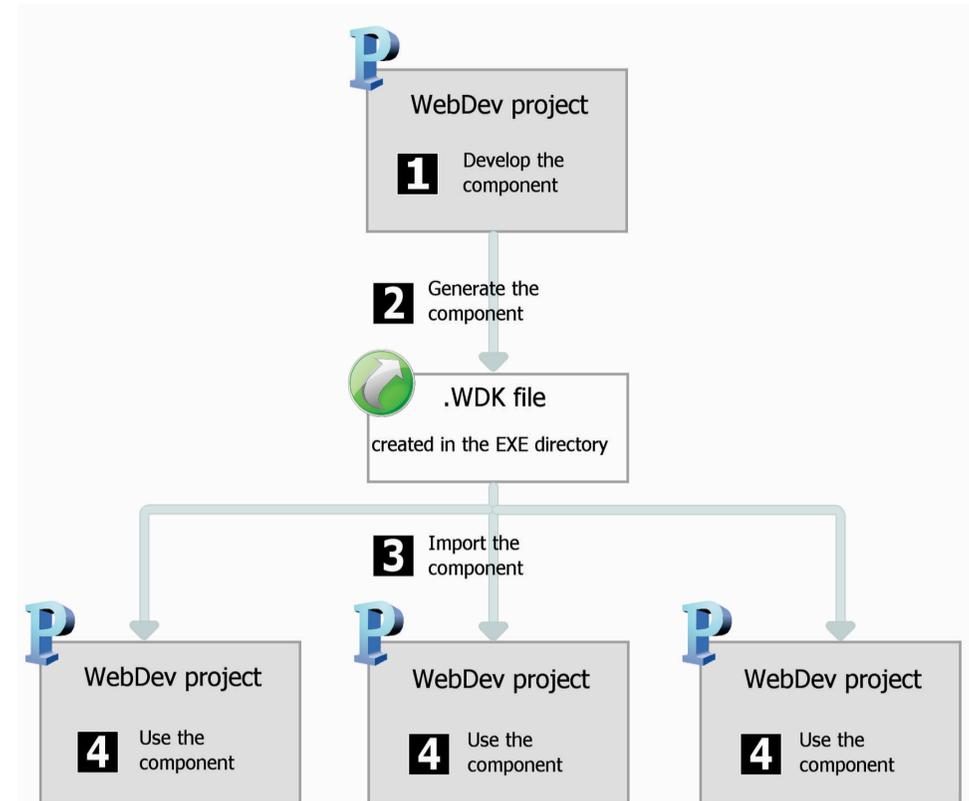
- import a project containing one or more internal components into the SCM. The internal components will be automatically included in the SCM.

# External component

An external component is a set of WebDev elements: pages, reports, analysis, ... This set of elements performs a specific feature. For example, an external component can correspond to one of the following features:

- Sending faxes,
- Sending emails,
- ...

An external WebDev component can be redistributed to other WebDev developers (free of charge or not). These developers will be able to easily include the feature proposed by the external component in their site without having to access to the corresponding source code (if it has not been distributed as well). The external component will be included in the site and distributed along with it.



# External component in practice

## 1 Overview

### 1.1 Definition

An external component is a set of WebDev elements: pages, reports, analysis, ... This set of elements performs a specific feature. For example, a component can correspond to one of the following features:

- Sending faxes
- Sending emails, ...

**Notes:**

- To optimize the organization of your projects, you have the ability to use the internal components. See "Internal component", page 98 for more details.
- In the rest of this chapter, "component" means "external component".

**An external component can be distributed to other WebDev developers** (free of charge or not). These developers will be able to easily include the feature proposed by the component in their site. The component will be included in the application or the site and distributed along with the site.

When creating the component, the author specifies:

- the origin of the component elements. The developer can build a component from a WinDev project linked to a WebDev project and/or to a WinDev Mobile project.
- the component elements visible (or not) to the user of the component. The visible elements will be accessible in the project graph or by programming.
- the mode for using the component (how to use the component, the parameters used to access the component features, ...). A short documentation (to be completed) is automatically generated and associated with the component.

**Note:** Several component examples are provided with WebDev. These components can be accessed from the "Wizards, Examples and Components" pane ("Components").

### 1.2 What is an external component made of?

A component is defined and generated from an existing WebDev project. It can then be included in other WebDev projects.

All the component elements are independent from the project where the component is included. The component can have its own analysis, pages, reports, data files, etc. When creating the component generation, specify whether these elements will be made available or not to the WebDev project that includes this component.

The component is made of three files:

<ComponentName>.WDK	<b>Component file.</b> Contains all the elements of the component. This file is required to include the component in a project. This file must also be supplied in the setup procedure of the site that uses the component.
<ComponentName>.WDI	<b>Description of the component</b> (when developing sites only). This file is required to include the component in a project. This file must not be supplied in the setup procedure of the site that uses the component.
<ComponentName>.WDO (optional file)	Optional file. File in text format containing the list of the additional elements supplied with the component (data files, .INI files, ...). See "The WDO file", page 107.

### 1.3 What is an external component made of?

An external component contains:

- the different elements to distribute (pages, reports, ...). These elements can be accessible (or not) when the component is imported into a WebDev project.
- a short description of the component.
- a help allowing this component to be re-used. This help is generated from the code comments.

**Note:** By default, if a component uses an analysis, files, ... the HyperFileSQL functions will handle these elements in an independent context. These

parameters can be modified in the advanced options of the component. See the online help for more details.

## 2 Creating and generating an external component

An external component must be created from a project containing all the elements required for the component to work. **Therefore, we recommend that you use a specific project to create each component.**

The creation of a component is performed in several steps:

1. Developing the component elements.
2. Creating the component
3. Defining the advanced options of the component.
4. Generating the component.

The component can be distributed and re-used.

## 3 Distributing an external component

### 3.1 Overview

Once the component has been created, checked and generated, it can be made available to the developers. Several methods are available:

- **directly use the component**  
Example: the component is directly used on the computer of the developer who created it.
- **simple distribution**, by directly supplying the necessary files.  
Example: the component is intended to be used within the same company, by several developers. In this case, the necessary files can be copied onto a network drive.
- **setup procedure of the component** (with a setup procedure of the component identical to the one used for applications).  
Example: this distribution mode is recommended if the component is intended to be sold or distributed on a large scale, with regular updates.

files, text files, ...) used by the component. These files referenced in <ComponentName>.WDO will be automatically copied into the EXE directory of the project that uses the WebDev component.

### 3.3 Simple distribution of an external component

The simple distribution of a component consists in providing via simple copy the files required for the component to operate. The files are copied into a specific directory. This directory can be found on a network server for example.

When the component is imported into a WebDev project, select the WDI file corresponding to the component in the distribution directory.

To distribute a component, you must supply:

- The files automatically generated by WebDev (<ComponentName>.WDK and <ComponentName>.WDI files).
- If necessary, the specific files handled by the component (data files, initialization files, ...) as well as <ComponentName>.WDO. The <ComponentName>.WDO file contains the list of files that must be supplied with the component. See for more details "The WDO file", page 107.

### 3.4 Distribution via a setup procedure

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.

### 3.2 Direct use of the external component

The component is created and used on the same computer.

When the component is imported into a WebDev project, all you have to do is select the WDI file corresponding to the component. This file is available in the EXE directory of the component project.

**Caution:** if the component uses specific files (data files, text files, ...), you need to create a <ComponentName>.WDO file after generating the component. This file contains the list of external files (data

This setup mode is used to automatically manage:

- the WDO file and the setup of specific files (data files, ...) used by the component.
- the setup of specific 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.

To propose a setup procedure for a component:

1. Create the setup procedure of the component:
  - directly via "Workshop .. External Component .. Create the setup procedure of a component".
  - from the list of components found in the project ("Workshop .. External Component .. List of components generated from this project", "Setup" button).

## 4 Using an external component in a site

An external component can be re-used at any time in any WebDev site: all you have to know is the directory of the component files.

When a new version of a component is available, all you have to do is install the new files of this update in the setup directory of the component (according to the setup mode used).

See "Distributing an external component", page

## 5 Deploying a site containing an external component

### 5.1 Overview

The method for deploying a site that uses one or more components is the same as the method for deploying a standard site ("Workshop .. Create setup procedure").

When deploying a site that uses a component, the following files are automatically installed on the server:

- <ComponentName>.WDK,
- the files required for the component and for the application to operate.

2. In the wizard planes, specify:

- the setup media of the component.
- the languages proposed in the setup procedure.
- the default setup directory of the component.
- the optional modules to install. You have the ability to modify the list of files that will be distributed with the component. The WDO file will be automatically created from this list.
- ...

By default, the files required for installing a component are created in the INSTALL COMPO sub-directory of the project.

**Reminder:** when creating an application's setup program, the required files are created in the INSTALL directory of the project.

103 for more details.

If the component was published and imported via SCM, the updates are proposed when opening the project that uses the component.

If the component was distributed with a WebDev site, the component must be updated. See "Deploying a site containing an external component" for more details.

### 5.2 Updating the components and the deployed sites

When updating a component and/or a deployed site, you have the ability to install on the deployment computer of the site:

- the site, the component and the files required for them to operate.
- the site and the files required for the site to operate.
- the component and the files required for the component to operate.

See the online help for more details.

Two methods can be used to update a component on the end-user computers:

### 1. The recompilation of the host project

Recompile the project that uses the component and redistribute the site with the component. In this case, no version problem or compatibility problem will occur. The recompilation is required in the following cases:

- New features have been added into the component and they must be taken into account.
- The parameters of some procedures have been modified.

- Incompatibility between the new version and the earlier versions of the component.
- ...

### 2. Direct distribution of the .WDK file

Supply an update of the component (.WDK file) to the users without recompiling the project. In most cases, this possibility applies when:

- The new version is used to correct the problems of an earlier version.
- New features have been added to the component but they are not required for the site to operate.

## 6 Modifying an external component

### 6.1 Overview

A component that was created and generated can be modified at any time. The following modifications can correspond to:

- the addition of elements into the component.
- the deletion of elements from the component.
- the modification of rights on the component elements.
- the modification of one of the component elements.

In any case, after this modification, the component must be regenerated in order to take the modifications into account.

### 6.2 The different types of compatibility

The management of the compatibility is linked to the versions of a component. Indeed, if some modifications have been made to the component, the

sites that use the component may encounter runtime problems if they are not synchronized with this component.

The management of compatibility is an important step in the modification of a component. There are two types of compatibility:

- **backward compatibility:** the version of the component used (.WDK file) to compile sites using this component must always be greater than or equal to the current version used in deployment. This compatibility mode is automatically managed. If the version of the component installed with the site is less than the one used by the site, the site will not work.
- **Forward compatibility:** using a new version of a component with projects compiled with an earlier version can be forbidden. Therefore, the projects must be recompiled in order to use the new version of the component.

## 7 Advanced characteristics of an external component

### 7.1 Automatic documentation

A component must be supplied with a documentation in order to be reusable.

WebDev simplifies the creation of this documentation by proposing:

- **a general overview of the component.** This general overview is entered when the component

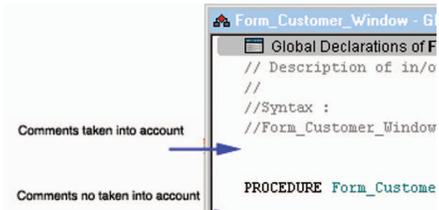
is generated. When using the component in a WebDev site, this overview will be visible when hovering the component listed in the "Wizards, Examples and Components" pane. See the online help for more details.

- **an automatic generation of the technical documentation** from the comments inserted into the source code of the component elements.

### What code comments are taken into account?

The following comments are automatically taken into account for the documentation of the component:

- The comments found at the beginning of the WLanguage procedures



- The comments found at the beginning of the following processes:
  - Initialization code of pages
  - Initialization code of reports
  - Initialization code of classes
  - Initialization code of sets of procedures.

### At what time is the documentation generated?

The documentation of the component is generated during the first generation of the component ("Workshop .. External Component .. Generate a component").

When generating this documentation:

- the comments found in the code are used to create the documentation.
- if no comments are found, WebDev will automatically document the accessible elements of the component by specifying the input/output parameters expected by each element. The corresponding comments are automatically created in the different elements.

**Note:** Whenever the component is re-generated, you have the ability to re-generate the documentation associated with the component ("Regenerate" button in the "Component history" window).

**Caution:** If the "Regenerate" button is used, the modifications performed in the generation wizard will be deleted.

### How do I access the documentation of a component?

The **general overview of the component** is automatically displayed when the component found in the "Wizards, Examples and Components" pane is hovered by the mouse cursor.

The **technical documentation** of the component is available:

- when a component is included in a project ("Workshop .. External component .. Import a component into the project").
- when double-clicking the icon of the component found in the "Wizards, Examples and Components" pane or via "Description" from the popup menu of the component.
- in the description of the components included in the project ("Workshop .. External component .. List of components imported into the project"). The "Description" button is used to get information about the component.

The **documentation specific to an element** of the component can be displayed:

- by double-clicking this element ("Project explorer" pane or project graph) or by pressing [F2] from the code of this element.
- in the list of components included in the project ("Workshop .. External component .. List of components imported into the project").

### 7.2 Visibility of a component element

When creating a component, you have the ability to define the component elements that will be accessible (or not) by the user of the component.

- If the element is accessible, the user of the component will see this element in the list of project elements. The user of the component will be able to handle these elements by programming (as if it were an element created in the project). **Note:** The code of this element is not visible though.
- If the element is not accessible, the user will not even know that this element exists.

**Caution:** Depending on the declaration mode of the project elements (class, set of procedures, ...), the accessible elements may vary. The table below presents the different methods for accessing an element according to its declaration mode.

### 7.3 The WDO file

When generating the external component, two files are automatically created in the EXE directory of the current project:

<Component-Name>.WDK	Contains all the elements that must be redistributed (pages, reports, ...) with the component
<Component-Name>.WDI	Interface of the component. This file contains: <ul style="list-style-type: none"> <li>- a help for using the component when it is checked in</li> <li>- the elements required for using the component in the project (compilation information, ...)</li> </ul>

These two files must be distributed along with the component.

**If the component uses additional elements (data files, ...),** you must add into the EXE directory of the project:

- a <ComponentName>.WDO file: this file contains the list of external files (data files, text files, ...) used by the component. These files must be supplied and installed with the component.
- the files that must be distributed with the component. These files can be placed in a specific tree structure. In this case, the code of the component must manage the access to these files in this tree structure.

### 7.4 What is this WDO file?

The <ComponentName>.WDO file is a file in TXT format that can be created and modified at any time. This file can be created and modified with Notepad, the standard text editor of Windows.

This file contains the list of external files (data files, text files, ...) used by the component and that must be supplied and installed with the component. These files must be copied into the EXE directory of the projects that use the component.

This ".WDO" file can contain:

- the full name of the file. For example: C:\Components\PickerComponent\InitialStatus.INI
- the name of the file. This file will be sought in the current directory of the component. For example: InitialStatus.INI
- a file name that uses a relative path. The possible syntaxes are:
  - Directory\FileName.xxx to specify a sub-directory of the current directory
  - .\FileName.xxx to specify the current directory
  - ..\FileName.xxx to specify the parent directory. For example: \PickerComponent\InitialStateINI

This file will be used when the component is included in the project. The paths specified in the WDO file must correspond to the path where the files are installed on the development computer of the component.

When including the component, the tree structure specified in the WDO file will be stored and reproduced in the EXE directory of the project. See for more details "Using an external component in a site", page 104.

**Example:** The "Zip Code" component uses a data file named "Cedex" (Cedex.fic and Cedex.ndx files). In the project for component creation, this data file is found in the EXE directory of the project.

In order for the component to be supplied and installed with the data file, the WDO file must be created in the EXE directory of the project for component creation. This file must contain the following lines:

```
.\CEDEX.FIC
.\CEDEX.NDX
```

**Distributing a component with WDO**

To distribute a component that uses a WDO file:

- If no setup procedure is used for the component, you must supply:
  - the WDK file
  - the WDI file
  - the WDO file
- all the necessary files referenced in the WDO file.

- If you are using a setup procedure for the component, the WDO file will be automatically created when creating the setup procedure of the component.

To do so, you must in the wizard:

1. Request the modification of the files to install ("Modify the list of files to install" in the "Additional modules").
2. Select the additional files to install. The WDO file will be automatically created and installed with the component.

# Generation modes

In addition to the Web sites, WebDev allows you to generate several types of projects.

**Sites**

The sites are the most often used generation mode. WebDev allows you to generate:

- static sites, semi-dynamic sites, dynamic sites
- PHP sites

**Libraries**

A library is a unique file that groups several elements of a WebDev project: pages, reports, etc. You can generate stand-alone libraries that can be used by other sites.

**Windows Vista/Windows 7 gadgets**

A gadget is a special type of web site meant to be integrated as a mini app on the desktop of Windows Vista or Windows 7. WebDev provides the necessary elements for creating and distributing gadgets.

**External components**

External components are application bricks allowing you to share one or more specific features among several applications. A component generated by WinDev can also be used in a WebDev or WinDev Mobile project.

**Webservices**

A Webservice (also called an XML Web service) can be generated from a WebDev project. A Webservice exposes a set of functions (one or more sets of procedures). It makes them accessible via the Web (or via a private network) by using the standard HTTP and SOAP protocols.

**Note:** A Webservice must be deployed on a WebDev Application Server before it can be used.



## Specific iPhone sites

The Web sites intended to be viewed on the Apple iPhone can benefit from specific features: splash screen at startup, full-screen mode, ... These features make them look like real applications. WebDev allows you to easily configure these special features while keeping the compatibility with the other browsers (mobile or not) found on the market.

# PHP generation in practice

## 1 Overview

WebDev enables you to generate PHP sites without even knowing the PHP language. The site can be entirely developed in WLanguage, like any standard WebDev site. However, you have the ability to enter your own PHP code (server code).

### 1.1 What is a PHP site?

A PHP site is compiled in PHP. If this site uses a database, the access to this database will be done via an ODBC driver (MyODBC, ODBC for Oracle, ...) or via the native access for MySQL.

The ODBC driver is specific to each database (it can be found on the site of the database publisher).

A PHP site is made of ".php" pages and it requires a PHP engine to run.

To compile (in PHP) a PHP project with WebDev, a PHP engine must be installed on the development computer. Several PHP engines are available on

the Internet. You must download one and install it. For example, you can use EasyPHP (<http://www.easyphp.org/>). This engine is used for example purposes, you can use any other PHP engine.

**Note:** PHP version 4.3.2 (or later) is required. A MySQL database is required. The PHP sites generated by WebDev can also be used by the PHP servers version 5.

### 1.2 Why develop a PHP site with WebDev?

For some sites (especially small ones), it may be useful to develop these sites in PHP in order for them to be hosted at no extra cost. You now have the ability to do this with WebDev.

**Note:** the number of WLanguage functions that can be used in a PHP project is limited in this version. This number of functions will increase in forthcoming versions.

## 2 Features of a PHP project

### 2.1 Creating a PHP project

To create a PHP project, all you have to do is specify in the project creation wizard ("File ... New ... Project") that you want to generate a dynamic PHP site.

As soon as the project is configured as a PHP project, all the tools available in WebDev are automatically configured for the PHP project:

- The PHP RAD that generates the PHP code from a project for a several WebDev features.
- The compiler indicates the functions or controls that cannot be used in PHP. A purple message is displayed in the "Compilation errors" pane.
- The deployment of the PHP site via WDDeploy.

**Note:** we recommend you create a new WebDev project to develop a PHP site rather than transform a WebDev site into PHP.

### 2.2 PHP RAD

Like for any project associated with an analysis, RAD can be used to generate the windows of your site.

RAD takes into account all the specific features of the PHP generation (controls, processes, functions available in a PHP site generated from WebDev).

### 2.3 Running the test of a PHP site in the editor

During the test of a PHP site, the site is compiled in PHP and started in your browser. You directly view the PHP pages generated by WebDev.

Reminder: A PHP engine is required on the development computer.

Compiling a PHP site allows you to:

- find out the WLanguage programming errors
- view the controls, processes, WLanguage functions not available in PHP generation. These errors appear in purple in the "Compilation errors" pane. See the online help for more details.

### 2.4 Deploying a PHP site

The method for deploying a PHP site is identical to the method for deploying a static site. When creating the setup procedure, WDDeploy starts and helps you implement your PHP site.

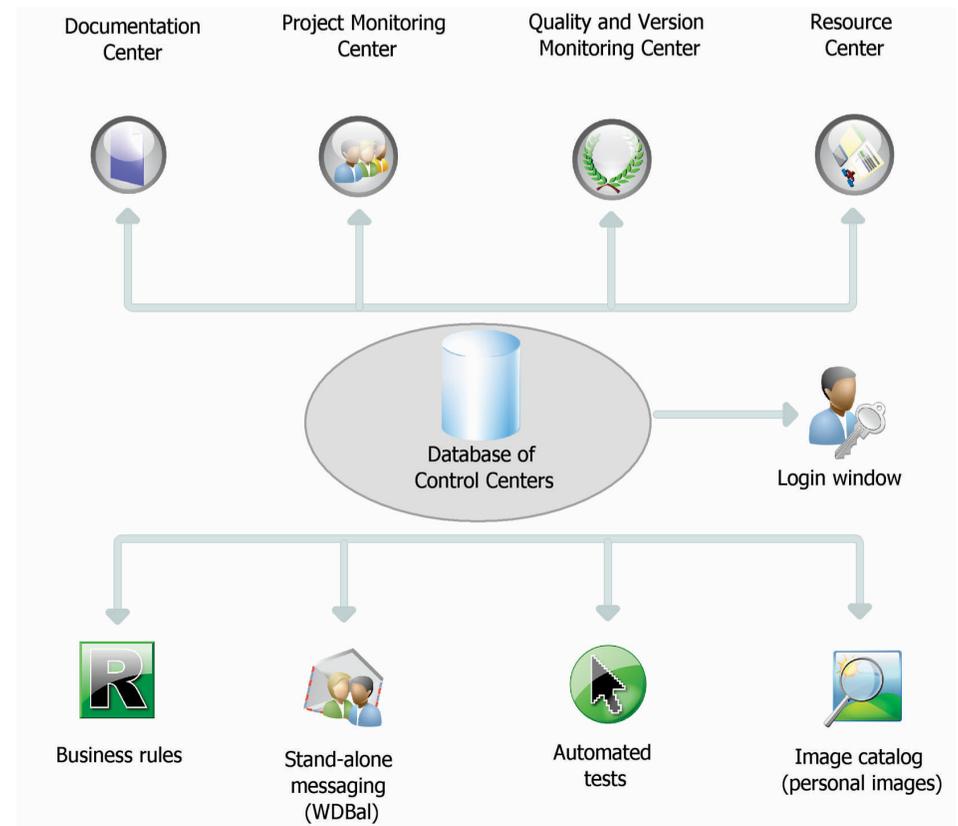
See the online help for more details.

# The Control Centers

To manage the life cycle of your development projects, WebDev enables you to use the Control Centers. The Control Centers are a set of tools intended to handle:

- the requirements of a project (assessments)
- the development, test and documentation tasks
- the tracking of bugs reported by the users
- the business rules used by the company in its developments

The Control Centers operate with a HyperFileSQL Classic or Client/Server database shared by several tools available in WebDev but also in WinDev and in WinDev mobile:



When installing WebDev, the setup program proposes:

- to create the database of Control Centers. This database will be automatically created in HyperFileSQL Classic format in the specified directory.
- to share an existing Control Center database.

# Managing the requirements

A requirement is a need that a product (or a service) should fulfill.

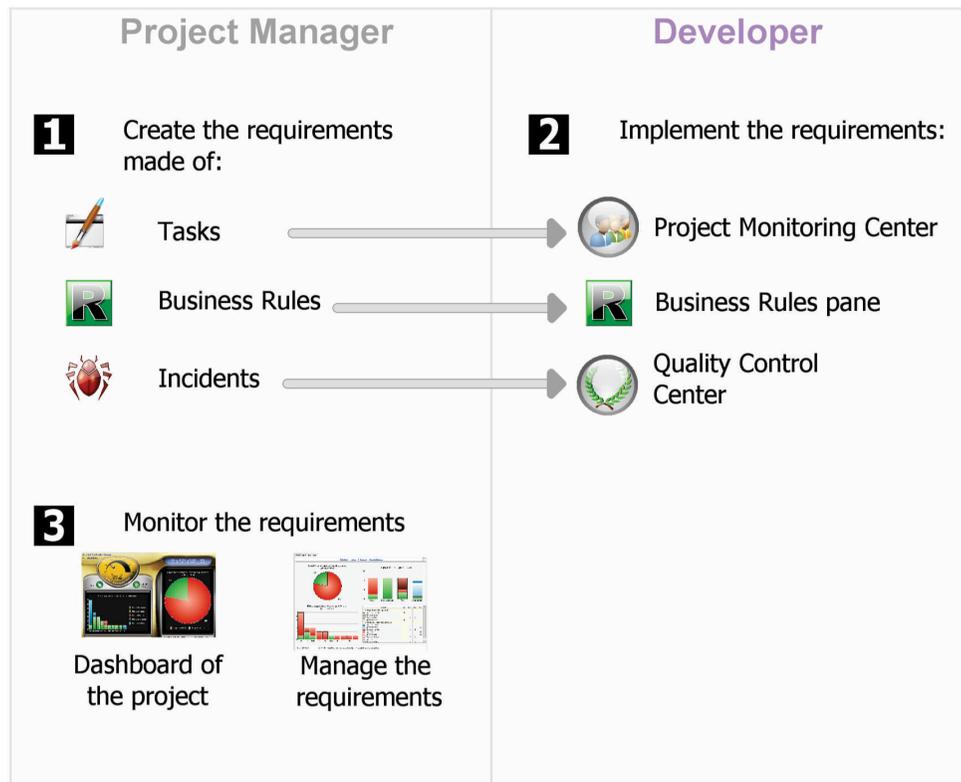
The Control Centers allow a project manager to manage the requirements of a development project.

To do so you must:

- define the different project contributors.
- define the requirements (with the different elements associated with them).

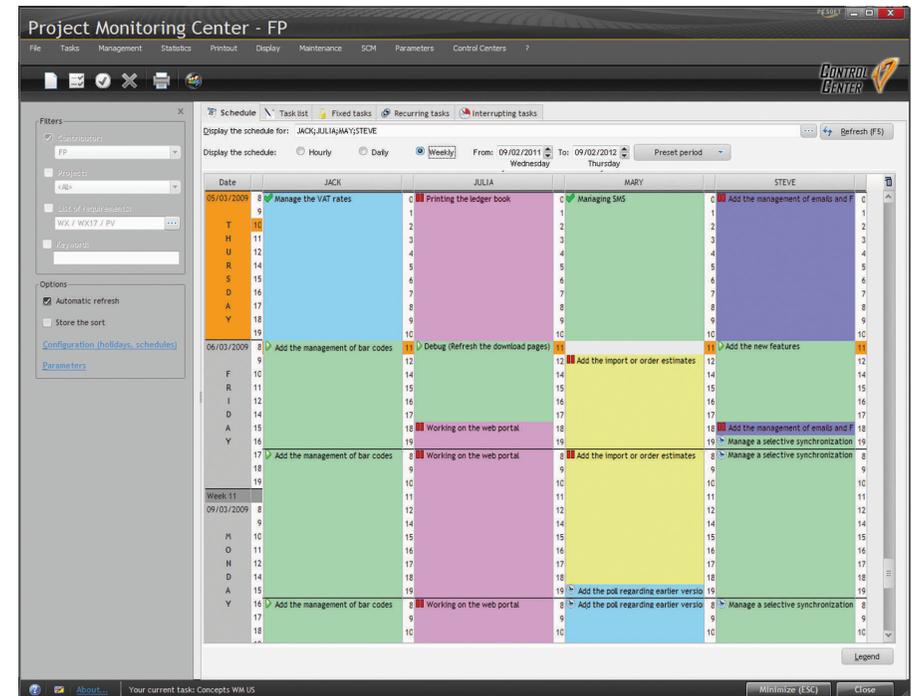
Each developer performs the different tasks assigned to him.

The project manager can follow the progress status of the project at any time.



# Project Monitoring Center

The Project Monitoring Center allows the contributors of a project to manage their task schedule. These tasks can be linked to requirements and they can correspond to several projects.



## Operating mode of the Project Monitoring Center

Once all the project tasks have been defined, the Project Monitoring Center takes care of everything. Entering the time spent on a task is almost automatic, it requires no specific action and it generates no particular constraint.

When the relevant project is opened, the Project Monitoring Center requests or indicates the current task. As soon as a task is completed, all you have to do is indicate that this task is over and specify the new task.

A task can be linked to a project element (page, report, ...). Whenever the relevant element is opened, the time spent on this element is counted and stored in the Monitoring Center. Conversely, the element corresponding to the task that you want to perform can be automatically opened from the task list.

Each developer can also see his own task list in the "Project Monitoring Center" pane.

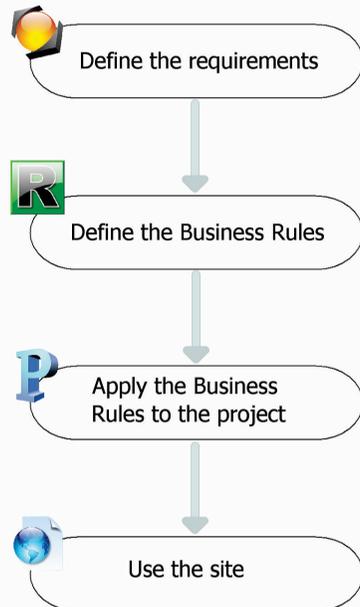
# Managing the business rules

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 shipping costs, a sales commission, a discount rate, a decay coefficient ...

A business rule can be simple or complex.

The business rules can come from:

- the specifications (corresponding to the requirements).
- the suggestions made by the users of the application.



During development, the business rules defined for the project are directly displayed in the "Business rules" pane of the development environment. This pane presents the number of project elements to which the business rules apply and the percentage of rule currently implemented.

## PART 4

# Developing Web sites: advanced concepts



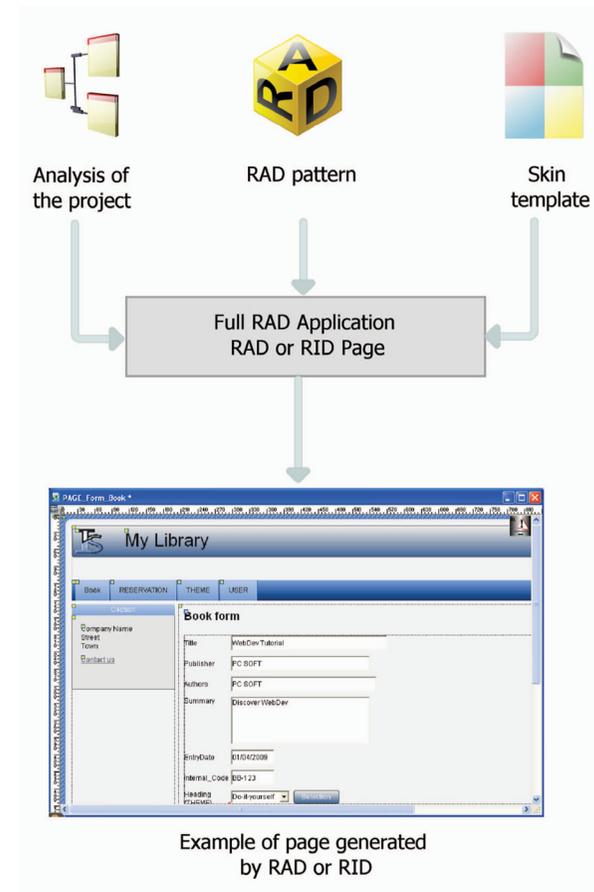
# RAD RID

RAD (Rapid Application Development) and RID (Rapid Interface Design) are used to create pages from:

- the analysis linked to the project,
- standard or custom RAD patterns,
- skin templates.

**In RAD generation**, the generated pages contain all the code needed for them to work. The test of these pages can be immediately run with the data found on the development computer.

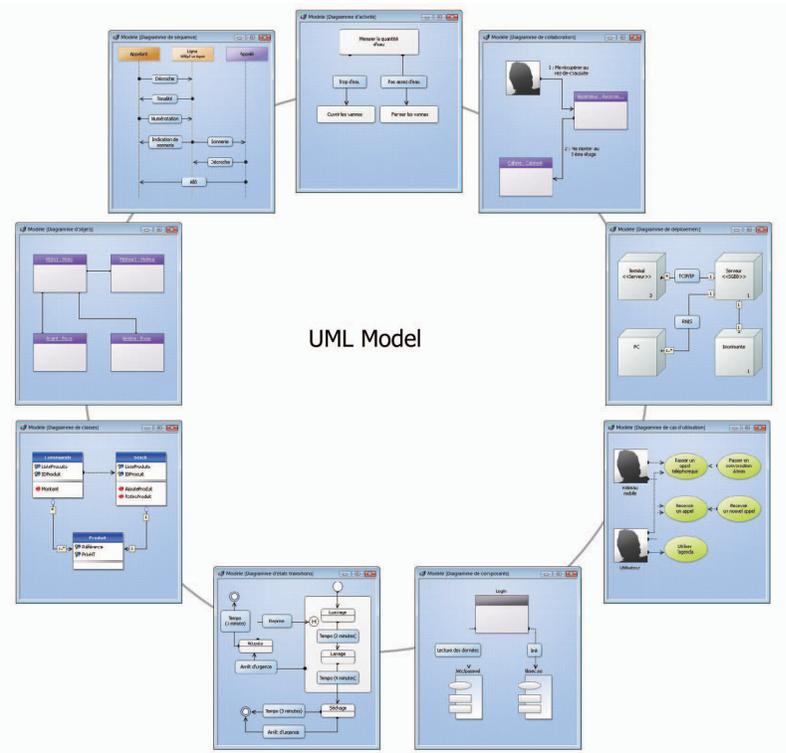
**In RID generation**, the generated pages only contain the controls linked to the items from the analysis. The source code required for these pages to work must be written by the developer. Your custom code can be entered directly.



# The UML model

WebDev allows you to create the nine types of UML models:

- **Class diagram:** describes the overall structure of a system.
- **Use case diagram:** represents the features of the system from the user's stand-point.
- **Object diagram:** represents a set of objects and their relationships at a given time.
- **Component diagram:** describes the physical and static architecture of a computer application.
- **Activity diagram:** represents the behavior of a method or the progress of a use case.
- **Sequence diagram:** presents the chronological order of the messages sent and received by a set of objects.
- **Collaboration diagram:** represents the structural organization of the objects that send and receive messages.
- **State-transition diagram:** represents a sequence of states.
- **Deployment diagram:** describes the hardware (nodes) used in a system implementations and the execution environments deployed on the hardware.



## The UML model in practice

### 1 Overview

The main objective of a development team is to create optimized applications, capable of satisfying the constantly evolving needs of their users.

The modeling of an application is used to specify the structure and the expected behavior of a system. It helps understand its organization and detect simplification and re-use opportunities as well as manage potential risks.

A model is a simplification of reality. It enables you to better understand the system to develop.

A diagram is the graphical representation of a set of elements that constitute the system. To view the system under different perspectives, the UML language (Unified Modeling Language) proposes nine diagrams, each one representing a system state.

WinDev allows you to create these nine types of UML model:

- Class diagram
- Use case diagram
- Object diagram
- Component diagram
- Activity diagram
- Sequence diagram
- Collaboration diagram
- State-transition diagram
- Deployment diagram

This chapter only provides an overview of UML. See a specific documentation about the UML language for more details.

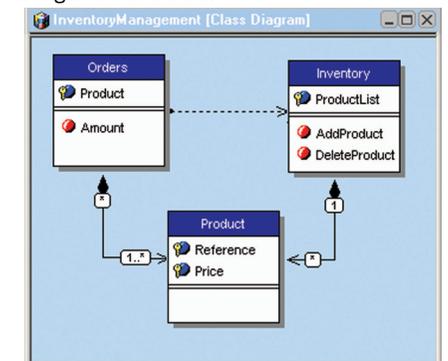
### 2 The different UML diagrams

#### 2.1 Class diagram

A class diagram is used to model the structure of a system via classes and via relationships between these classes.

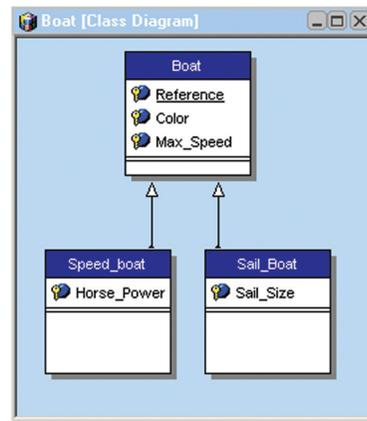
The class diagrams are the most common diagrams in the modeling of object-oriented systems.

For example, the following diagram presents the management of stocks:



A class diagram contains the following elements:

- **Class:** represents the structures of the application. Each class is divided into three compartments:
  - *the name of the class* indicates what the class is and not what it does.
  - *the attributes of the class* give the characteristics of the class.
  - *the class operations* represent the possible actions on the class. For example, the *Stocks* class contains the *ListProducts* attribute. This class also groups the *AddProduct* and *RemoveProduct* operations. These operations can be applied to the instances of the class.**Note:** The UML language defines three visibility levels for the attributes and for the operations:
  - **Public:** the element is visible by all the other classes.
  - **Protected:** the element is visible by the class itself and by the sub-classes.
  - **Private:** the element is visible by the class only.
- **Relationship:** describes the behavior of classes among themselves. Three types of relationships are available:
  - **Association:** Structural relationship between classes. For example, the *Orders* class is linked to the *Product* and *Customer* classes. A Customer can place several Orders. An order contains several products. An order must necessarily contain at least one product.
  - **Dependence:** Use relation establishing that a class's instances are linked to the instances of another element. For example, the *Orders* class uses the *Stock* class: before adding a product into an order, we must check whether this product is in stock.
  - **Generalization:** Relationship between a general class (parent) and a specific class (child) that derives from it. For example, the *Sail Boat* class and *Speed Boat* class are derived from the *Boat* class.



- **Package:** used to divide and organize the representation of the diagram (as the directories organizes the files). Each package can contain classes and relationships.

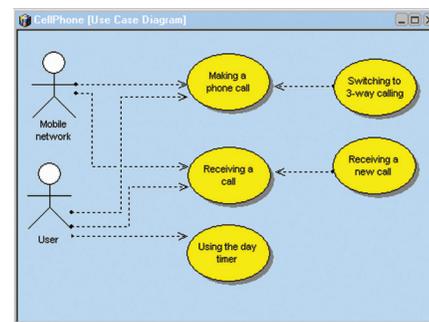
Via the generation of a class diagram, it is possible to create the structure of the WebDev classes used in your application.

### 2.2 Use case diagram

A use case diagram is used to view the behavior of a system in such way that:

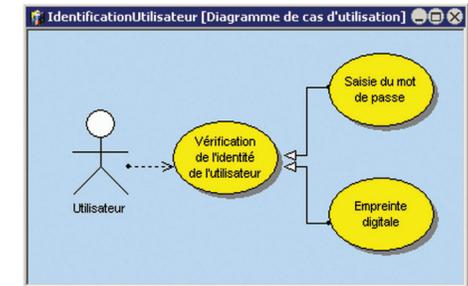
- the user can understand how to use each element.
- the developer can implement these elements.

For example, you can describe the behavior of a cell phone with a use case diagram.



A use case diagram contains the following elements:

- **Actor:** represents the role of the application users. For example, a person who works in a bank will be the *loan manager*. If this person has an account in this bank, he will also play the role of the *Customer*.
- **Use case:** describes a sequence of actions performed by the application. For example: Place an order, Enter an order, Create a new customer form ... A use case describes what an application does but does not specify how the application does it.
- **Relationship:** describes the behavior of the actors in relation to the use cases. Three types of relationships are available:
  - **Association:** Structural relationship between two linked elements.
  - **Dependency:** Relationship establishing that an element uses another one. For example, a bank *customer* may get cash from an ATM. In this case, the *Get Cash* action depends on the *Customer*. In order to get cash, the *Customer* must enter his *PIN number*. In this case, the *Get Cash* action depends on the *Password Input*.



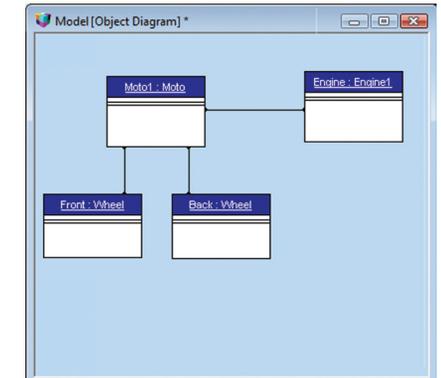
- **Package :** divides and organizes the representation of the diagram (as the directories organizes the files). Each package can contain actors and use cases.

### 2.3 Object diagram

An object diagram presents a set of objects and their relationships at a given time.

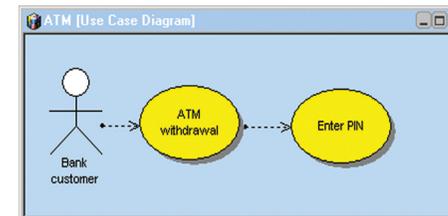
An object diagram is used to show a context (before or after an interaction between objects for example).

For example, the diagram below presents a part of the general structure of motorcycles:



An object diagram contains the following elements:

- **object:** represents the instance of a class. Note: If a class diagram is open, you have the ability to create an object from a class found in this diagram (drag and drop from the "UML analysis" pane).
- **composite object:** visually represents an object made of other objects. For example: a window that contains scrollbars, buttons, ...
- **link:** represents the relationship between the various objects.

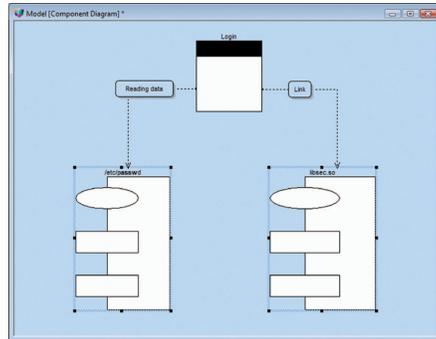


- **Generalization:** Relationship used to organize the elements according to a hierarchy. For example:
  - the Customer actor can be one of two types: individual customer or enterprise customer.
  - checking the identity of a user can be performed in two ways: password input or fingerprint check.

## 2.4 Component diagram

A component diagram describes the physical and static architecture of a computer application. For example: source files, libraries, executables, ...

For example, the diagram below presents the operating mode of a program allowing you to log in text mode in Unix.



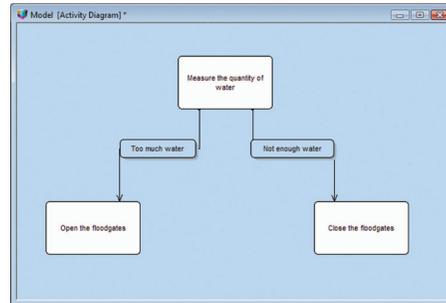
A component diagram contains the following elements:

- **module**: represents the different physical elements that constitute an application. For example: a file, a library, ... A module can be represented:
  - by a **specification** that shows the interface of the module. This specification can be **generic** for the customizable classes.
  - by its **body**, that presents the implementation of the module.
- **task**: represents a component having its own control thread.
- **main programs** of the application.
- **sub-programs**: groups the procedure and functions that do not belong to classes.

## 2.5 Activity diagram

An activity diagram represents the behavior of a method or the flow of a use case.

For example, the following diagram presents the flow of a dam:



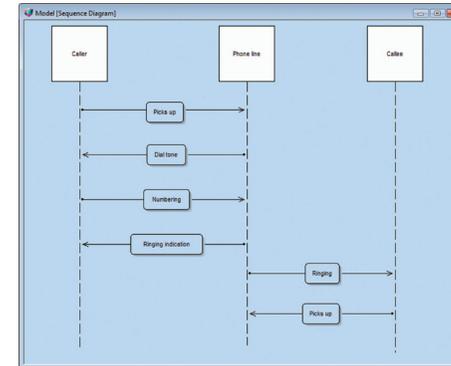
An activity diagram contains the following elements:

- **activity**: presents a specific step in the execution of a mechanism. For example: "Create an estimate", "Open a window", "Check knowledge", ...
- **synchronization bar**: used to synchronize the different activities:
  - by indicating the activities that must be performed before a given activity. For example: "Press the clutch" and "Change gear" before "Release the clutch".
  - by indicating the activities to perform in parallel.
- **object**: used to attach activities to the object that performs these activities. For example, the "Order" and "Pay" activities are attached to the "Customer" object; the "Teach" and "Check the knowledge" activities are attached to the "Teacher" object.
- **send signal**: represents the sending of a signal to an object.
- **wait for signal**: represents the wait for a signal coming from an object.
- **transition**: represents the passing from one finished activity to another. For example: "Too much water", "Enough money",...

## 2.6 Sequence diagram

A sequence diagram represents the chronological order of the messages sent and received by a set of objects.

For example, the following diagram represents the beginning of a phone call:



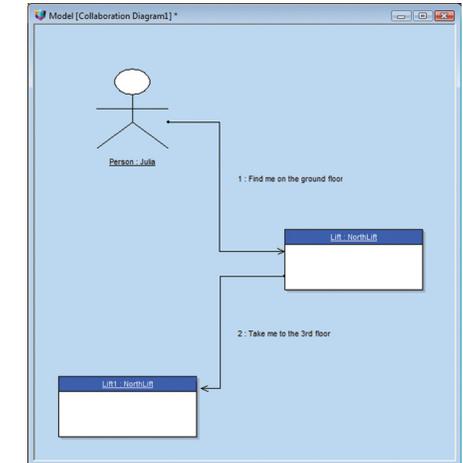
A sequence diagram contains the following elements:

- **object**: represents the different objects used. Each object is represented by a square at the top of a dotted line. This line represents the lifespan of the object. For example: "Caller", "Called", ...
- **activation period of an object**: activation periods can be inserted into the line representing the lifespan of an object. These periods represent the times when the object is active.
- **message**: represents, via horizontal arrows, the message exchanged between the different objects. These arrows are oriented from the issuer of the message to the recipient. The order in which the messages are sent is given by the position of the arrows on the vertical axis. For example: "Pick up", "Ring", ...

## 2.7 Collaboration diagram

A collaboration diagram presents the structural organization of the objects that send and receive messages.

For example, the diagram below presents the use of an elevator by a person:



A collaboration diagram contains the following elements:

- **object**: represents the different objects used.
- **actor**: represents an element external to the system. A person for example.
- **message**: represents the message exchanged between the various objects.

## 2.8 State-transition diagram

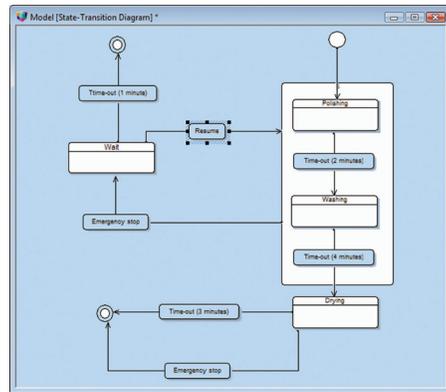
A state-transition diagram presents a sequence of states that an object goes through during its life-cycle. It is used to describe the changes of states for an object or for a component.

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.

For example, the diagram below presents the different steps of a car wash:



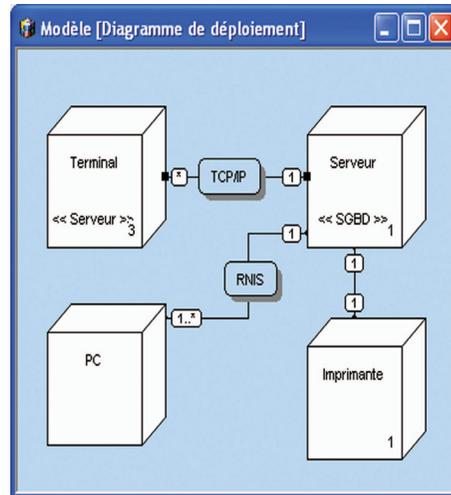
A state-transition diagram contains the following elements:

- **state**: represents the value of the object attributes at a given time.
- **initial state**: represents the state when the system is started.
- **final state**: represents the state of the system at the end of the operation.
- **super-state**: used to structure the diagram by indicating several distinct levels between states.
- **history**: represents the last active state of a super-state.
- **stub**: used to symbolize the states found in a super-state. This enables you to link these states to other states that do not belong to the super-state.
- **transition**: represents the passing from a state to another one.

### 2.9 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.

For example, the diagram below presents the different hardware devices used in a company:



A deployment diagram contains the following elements:

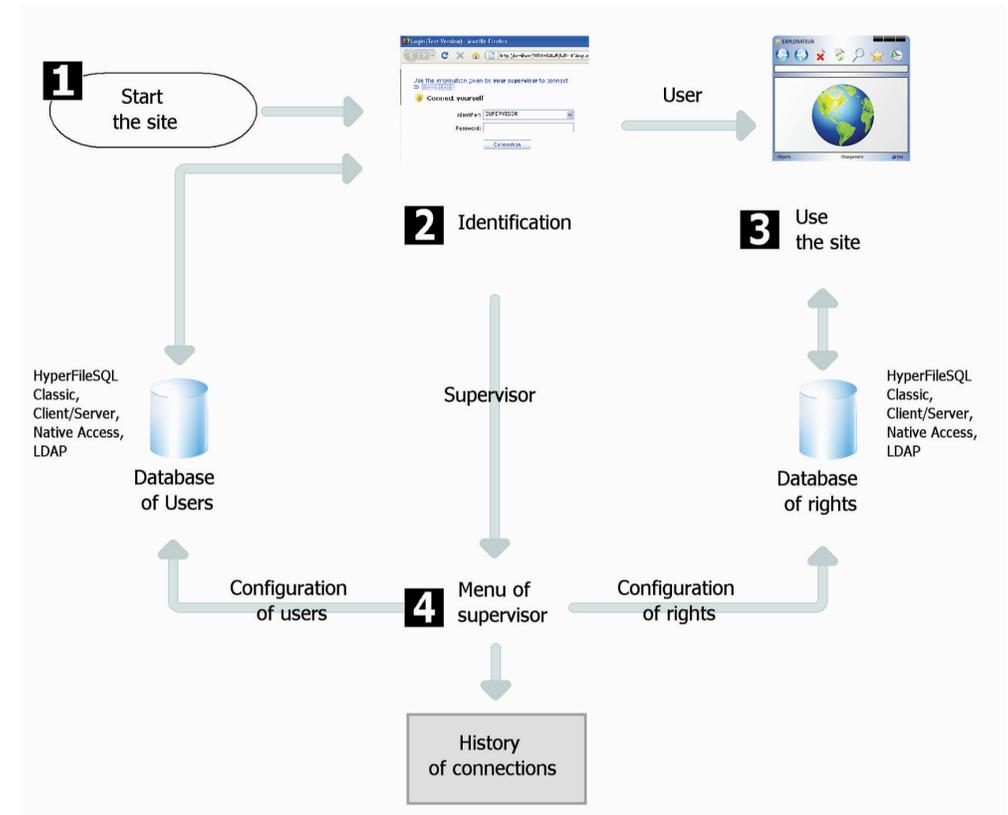
- **node class**: represents a class of hardware resources. For example: a server, a PC, a printer, ...
- **node instance**: represents a hardware resource. For example: server number 3, printer number 7, ...
- **connection**: describes the communication support between two nodes. For example: TCP/IP.

# User groupware

The User Groupware is used to easily include a management of users and rights in a WebDev site.

Two types of users are taken into account:

- simple user, who directly uses the site.
- supervisor, who can configure the users and their rights.



# The user groupware in practice

## 1 Overview

An Intranet or Internet site requires the definition of the role of the different participants. It is often necessary to define several access levels according to the user.

Indeed, all users do not have the same responsibilities or the same requirements. Their ability to use some of the application features can be customized.

Let's take a simple example: when implementing an application for sales management, the application proposes the following features:

- Seeing the price list,
- Modifying the price list,
- Entering the orders,
- Entering the customers.

The accesses differ according to the user. Some examples:

- the administrative assistants can see the price

list and create 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.

To manage these access levels in your Web-Dev sites, all you have to do is include the user groupware in your site. In just a few mouse clicks, a standard application can be changed into an application that manages several access levels. This feature can be easily implemented.

When running the site, the manager will be able to create users (identified by their login and password) and to give them access to some features of the site.

Note: the user groupware is available in the dynamic WebDev sites running on Windows or Linux. The user groupware is not available for the static, semi-dynamic, PHP, AWP or Ajax sites.

## 2 How does the user groupware operate?

A site using the user groupware has two use levels:

- the user level.
- the supervisor level.

### 2.1 The user level

Users connect to the application via a login page and they can access the features that have been allowed for them.

Note: the user groupware enables you to connect using an LDAP directory.

### 2.2 The supervisor level

The supervisor connects to the application via a login page and he can access an advanced menu allowing him to configure the users and their rights, or to start the application.

The configuration page allows the supervisor to:

- start the application.
- configuring the groupware.

Configuring the groupware consists in creating users and groups, associating the users with the groups as well as managing the rights granted to each user (or group) on each page found in the application. The rights can be defined for the menu options, for the groups of controls and for the controls. The supervisor can gray these elements, or make them inactive or invisible. These configurations have priority over any configuration defined in the program.

See "Configuring the site using the user groupware", page 130 for more details.

**Note:** The information about the users and their rights is stored in data files in HyperFileSQL format. The data files regarding the users can be common to several sites. The data files containing the rights granted to the users on the different pages of the site are specific to the site and they cannot be shared.

## 3 Implementing the user groupware

### 3.1 Adding the user groupware into a site

To implement the developer groupware in a Web-Dev site, all you have to do is select: "Workshop .. Configure the user groupware". The window for configuring the user groupware is used to define:

- The management mode of the user groupware.
- The way to start the user groupware.
- The location of the data files of the user groupware.

#### Mode for including the User Groupware

Two methods can be used to include the user groupware:

- **Default integration:** the programming of the user groupware is automatically included in your application.
- **A new page is included in your project: the login page.** It comes up when starting the site and allows the user to connect to the site. You can customize this page by applying the skin template of your site for instance.
- **Custom integration:** all the user groupware programming is automatically included in your site. **All the pages required to manage the user groupware (login and user management pages) are automatically added to your project.** You can customize all the pages used by the user groupware.

The different pages used by the user groupware are presented in details in "The WebDev pages", page 132.

#### Starting the user groupware

Two modes are available for starting the user groupware:

- **Auto run:** the login page of the user groupware is displayed before any other page of the application. The initialization code of the project is run once the login page has been opened and validated.
- **Manual start:** the login page will only be opened if the **gpwOpen** function is used. This option allows you to run the project initialization code for instance, before opening the login page. This enables you to display a page requesting the runtime language of the site for example.

#### Automatic login in test mode

You can specify the login and password that will be used in "Automatic test" mode. This information will be used if an automatic test is run on the site.

#### LDAP directory

If the company of your customer uses an LDAP directory, you have the ability to point the user groupware to this directory. Therefore, the account and the password found in the LDAP directory will be automatically requested from the user.

Two modes are available:

- **Without rights management:** In this case, no groupware data file will be created. If the application is started in automatic mode, the login page will ask the user to identify himself. If the user is saved in the LDAP directory, the site will load; otherwise, it will close. A supervisor cannot configure the rights on the pages.
- **With right management:** only the users found in the LDAP directory will be able to connect. The rights can be configured on the pages.

See the online help for more details.

**Note:** The user groupware operates with Active Directory. It does not work with openLDAP.

#### Data files of user groupware

Two categories of data files are managed by the user groupware:

- **the data files used to identify the users.** These data files can be common to several applications.
- **the data files used to manage the user rights** on the different pages of the site. These data files are specific to the site.

#### The password of the data files

By default, the data files of user groupware have a specific password. This password is: "PCS-GPW2001".

To change this password, enter the new password. The characters typed are displayed as stars.

**Note:** This password is used when re-indexing or opening the data files with WDMAP for example.

See "The data files", page 132 for more details.

### 3.2 Running the test of the site

When running the test of a site that manages the user groupware, the first page displayed is the login page (regardless of the first page defined in your application).

By default, a single user exists: the supervisor. To login as the supervisor, just enter the following information in the login page:

- Name: SUPERVISOR
- Password: SUPERVISOR

You can now run the test of your site or configure the user groupware.

#### Notes:

- If you do not want the first page of your site to be the login page, clear "Auto run" in the User Groupware's options. To do so, use the `WLanguage` function named `gpwOpen` to display the login page.
- The pages for managing the user groupware are run before the initialization code of the project.
- The first page of your site defined in the project will be run after the login page (when the user is not the supervisor).
- If you do not want to start the user groupware during the test of the site, uncheck "Workshop

## 4 Configuring the site using the user groupware

Once the site is deployed, the user groupware is configured by the application supervisor.

This configuration consists in creating users and groups of users, and in granting them specific rights for each control found in each page of the application.

These rights are used to gray controls, groups of controls or menu options, and to make them visible or invisible.

To configure the user groupware, simply:

1. Start the site and connect as the supervisor:
  - Name: SUPERVISOR
  - Password: SUPERVISOR
2. Select "Configure the groupware".

.. Configure the User Groupware". If the user groupware is re-enabled later, the data files previously used by the user groupware will not be erased.

### 3.3 Installing a site that uses the user groupware

To install a site that uses the user groupware, all you have to do is select "Workshop .. Deploy the site .. Deploy the site".

A screen specific to the user groupware enables you to configure the location of the data files for the user groupware.

#### Notes:

- If the data files of the user groupware have been configured for your client, they must be selected when preparing the setup program. Therefore, you must customize the list of files to install.
- If the groupware files have not been configured, only the Supervisor user will be able to login when the application is started for the first time.

**Tip :** if you do not configure the different use levels of the user groupware for your clients, we recommend that you provide them with a document listing all the controls found in your pages in order to get an optimized configuration.

### 4.1 Managing the users

To create a new user, you must specify:

- the name of the user (mandatory)
- the first name of the user
- the login of the user. This login corresponds to the identifier of the user when he connects to the application.
- the password of the user. This password is optional and it can be entered by the user himself during his first login (check the corresponding option).

The user can be defined as being a supervisor of the site.

A user can be modified or deleted. When deleting a user, you have the ability to delete the entire user or to delete his rights only.

The users can be organized in groups. When creating a group, you have the ability to copy the rights previously defined for another group or user.

A group can be created or deleted. When deleting a group, you have the ability to delete the group and its users or to delete the group only. In this case, the users are no longer associated with a group.

#### Notes:

- The supervisor password should be changed when it is first used.
- The "Default" group is the group proposed by default in the user groupware. By default, this group has all the rights on the application.
- The <None> group and the DEFAULT group cannot be deleted.
- If you are using an LDAP directory, you have the ability to import the users found in the directory in order to manage the rights of these users.

### 4.2 Managing the rights

The supervisor can manage the rights granted to each user (or group of users) on the elements found in the project pages.

For each association between a user and a page, a specific state can be defined for all the elements found in the page.

The elements managed in the pages, internal pages, and page templates are as follows:

- the controls
- the groups of controls
- the menu options

The states available for each element are as follows:

- **Default:** the behavior of the element corresponds to the default behavior, defined in the application.
- **Disabled:** the element is displayed but no input can be performed.
- **Grayed:** the element is grayed. No input can be performed.
- **Invisible:** the element is not displayed.

## 5 Tips for a site managing the user groupware

### 5.1 Using groups of controls

To simplify the configuration of the user groupware according to the users, the controls should be organized in groups of controls.

In your pages, you have the ability to create groups of controls according to the controls that must be displayed for a specific type of user. The ability to associate a control with several groups of controls increases the number of possible combinations.

These groups of controls can be created in your site only for managing the groupware, no specific programming is required.

### 5.2 Visibility of controls

When developing your application, you can define the visibility options for the elements found in your pages:

- when describing the element (7-tab window),
- by programming ( `..State` or `..Visible` properties).

When configuring the user groupware, the supervisor can define different visibility characteristics. The characteristics defined by the supervisor have priority.

For example, a button is used to make a control active. This control was grayed by the supervisor. Your code will be ignored and the control will not be active.

### 5.3 Defining the rights

To get a definition of the rights corresponding to the features of your site, we advise you to:

- configure the rights for your application before you create the setup program. Then, all you have to do is add the data files of the groupware when creating the site setup.
- supply a programming documentation that gives the names of the controls, groups of controls and options that must be configured according to the level of use defined for the site.

## 6 The elements of the user groupware

### 6.1 The WebDev pages

Depending on the integration mode of the user groupware in your application, one or more pages can be included in your project. These pages are as follows.

**Note:** the operating mode of these pages is presented in details in "Configuring the site using the user groupware", page 130.

#### Login page (GPWLOGIN\_WB.WDW)

This page is used by all the Web users to connect to the site. In this page, the Web user must enter his name and password (optional) that were previously defined by the site supervisor. During the startup, you have the ability to use the "SUPERVISOR" login and the "SUPERVISOR" password.

This page also allows:

- the Web user to change his password
- the supervisor to configure the rights and run the site.

#### User Form page (GPWUserForm\_WB.WDW)

This page is available in custom mode only. This page allows the supervisor to enter a new user and his personal details.

#### User and Configuration Association page (GPWAssociationConfiguration\_WB.WDW)

This page is available in custom mode only. This page allows the supervisor to manage the Web users, the groups and the rights.

#### Configuration Details page (GPWConfigurationDetails\_WB.WDW)

This page is available in custom mode only. This pages allows the supervisor to configure for each page the rights granted to a Web user (or to a group) on all the controls, groups of controls and menu options found in the page.

#### Configuration Choice page (GPWConfigurationChoice\_WB.WDW)

This page is available in custom mode only. This page allows the supervisor to copy a configuration that was previously defined (for a user or for a group).

#### Configuration Form page (GPWConfigurationForm\_WB.WDW)

This page is available in custom mode only. This page allows the supervisor to create a user or a group.

#### Connection History Form page (GPWConnectionHistory\_WB.WDW)

This page is available in custom mode only. This page allows the supervisor to view the connection history (and to purge that history if needed).

### 6.2 The data files

Two categories of files are managed:

- The files used to identify the users. These files can be common to several applications.
- The data files used to manage the rights granted to the users on the different application elements. These files are specific to the application and they cannot be shared with another application.

By default, these data files are created in HyperFileSQL Classic format, but you can use HyperFileSQL Client/Server data files. See the online help for more details.

**Password:** By default, the data files of the user groupware have a specific password. This password is: "PCSGPW2001".

#### To change this password:

1. Display the window for configuring the user groupware ("Workshop .. Configure the user groupware").
2. In the "Files" tab, enter the new password. The characters typed are displayed as stars.

**Note:** This password is used when re-indexing or opening the data files with WDMAP for example.

#### User data files

These files are the following ones:

GPWUser	File of users. Contains all the characteristics of the users (login, last name, first name, password, ...).
GPWUserConfiguration	File of groups of users

GPWConnectionLog	File containing the history of connections. This file is found only if the history of connections was requested.
------------------	--

The items of these data files are presented in details in the online help.

#### Where are these data files created?

- **By default**, these data files (HyperFileSQL Classic mode) are created in a subdirectory of your project (EXE\GPW\_<Project Name>). When including the user groupware, a different directory can be specified to test your site.
- When creating the setup program, you can configure the location of these data files. Therefore, a common directory can be selected for several sites that manage the user groupware.
- If "Manual start" is selected, *gpwOpen* is used to start the login page of the user groupware as well as to customize the directory of the data files for user groupware. This directory, if it is specified, will be taken into account.

**Note:** if no specific setting is performed, the user data files (HyperFileSQL Classic mode) will be created in a sub-directory of the setup directory of the library named \GPW\_<Project Name>.

#### Data files of rights

These files are the following ones:

GPWConfiguration	File containing the description of the available configurations. Each configuration is associated with a group of users.
GPWElement-Configuration	Status of each site element for each configuration. For example, allows you to store whether the "Statistics menu" must be grayed for the "Assistant" user.
GPWElement	File containing the different elements of the site to configure (page controls, menu options, ...).

The items of these data files are presented in details in the online help.

#### Where are these data files created?

- **By default**, these data files (HyperFileSQL Classic mode) are created in a subdirectory of your project (EXE\GPW\_<Project Name>).
- When creating the setup program, you can configure the location of these data files. This enables you to select a specific directory for the files containing the sites rights.
- If "Manual Run" is selected, *gpwOpen* is used to start the login window or page of the user groupware as well as to customize the directory of the data files for user groupware. This directory, if it is specified, will be taken into account.

**Note:** if no specific setting is performed, the data files for the rights will be created in a sub-directory of the setup directory named \GPW\_<Project Name>.

## 7 Modifying the user groupware

### 7.1 Customizing the groupware pages

#### The login page

The login page being displayed to all the users who load your application, you may want to customize it. If you have chosen to include this page in your project, simply apply the style or the skin templates used by your site to it.

**Caution:** do not rename this page.

#### All the pages of the user groupware

If all the pages of the user groupware have been included in your project, they can all be customized. However, only the login page will be viewed by all the users. The groupware configuration pages are viewed only by the application supervisor.

**Caution:** do not rename this page.

### 7.2 User groupware and multilingual application

If your site is a multilingual site, this feature must be managed in the pages of the user groupware. By default, the pages of the user groupware support English and French.

To use the user groupware in a multilingual site, you must:

1. Configure the user groupware as follows:
  - all the pages of the user groupware must be included in the project.
  - the "Manual start" option must be selected.
2. Translating the different pages of the user groupware.

By default, the pages of the user groupware are available in English and French.

To translate these pages into other languages, all you have to do is translate the captions, ... (by using the WDMMSG tool if needed to extract the messages to translate).

**Reminder:** WDMMSG is an optional WebDev tool used to extract and re-integrate all the messages to translate.

3. Run the site in the requested language. To do so, you can:

- Ask the user (if necessary) for the language in which the site must be run (via a special page for example).
- Configure the runtime language of the site with **Nation**.
- Open the first page for managing the user

groupware with `gpwOpen`

See "Multilingual sites", page 136 for more details.

### 7.3 Modifying the first page of the application according to the user

To change the first page of the application according to the user, use `gpwGetUserInfo` in the initialization code of your project. This option enables you to get information about the user such as his name or password for example.

Depending his name, you have the ability to open a specific page.

Example:

```
IF gpwGetUserInfo(...
  gpwLogin) = "FP" THEN
  PageDisplay("PAGE_Management")
ELSE
  PageDisplay("Menu")
END
```

### 7.4 Start the login page from a page of the site

When the user groupware is included in an application, you can choose whether the user groupware will be automatically started or not. This option can be modified at any time.

By default (auto run), the login page is the first page of the application. It is run with the initialization code of the project.

In some cases, a configuration page must be opened before the login page. To do so you must:

1. Select "Manual start".
2. Open the login page anywhere in the code with `gpwOpen`

### 7.5 Configuring the data files of the user groupware

Two types of data files are used by the user groupware:

- data files of users.
- data files of rights.

The location of these data files can be configured:

- when including the user groupware for the tests.
- when creating the setup procedure of the site.
- by programming.

See "The data files", page 132 for more details.

### 7.6 Reinitializing the data files of the user groupware

To reinitialize the data files of the user groupware, you must delete the data files specific to the user groupware (file of users and file of rights). These data files are automatically re-created when loading the site.

#### Notes:

- If only the user data files are deleted (GPWUser-Configuration and GPWUser), only the supervisor will be able to start the site.
- If only the data files of rights are deleted (GPWElementConfiguration, GPWConfiguration and GPWElement), all the users will have all the rights on the entire site.

See "The data files", page 132 for more details.

### 7.7 Modifying the data files used by the user groupware

The structure of the data files specific to the user groupware cannot be modified.

**Note:** These data files contain confidential information and therefore they are encrypted.

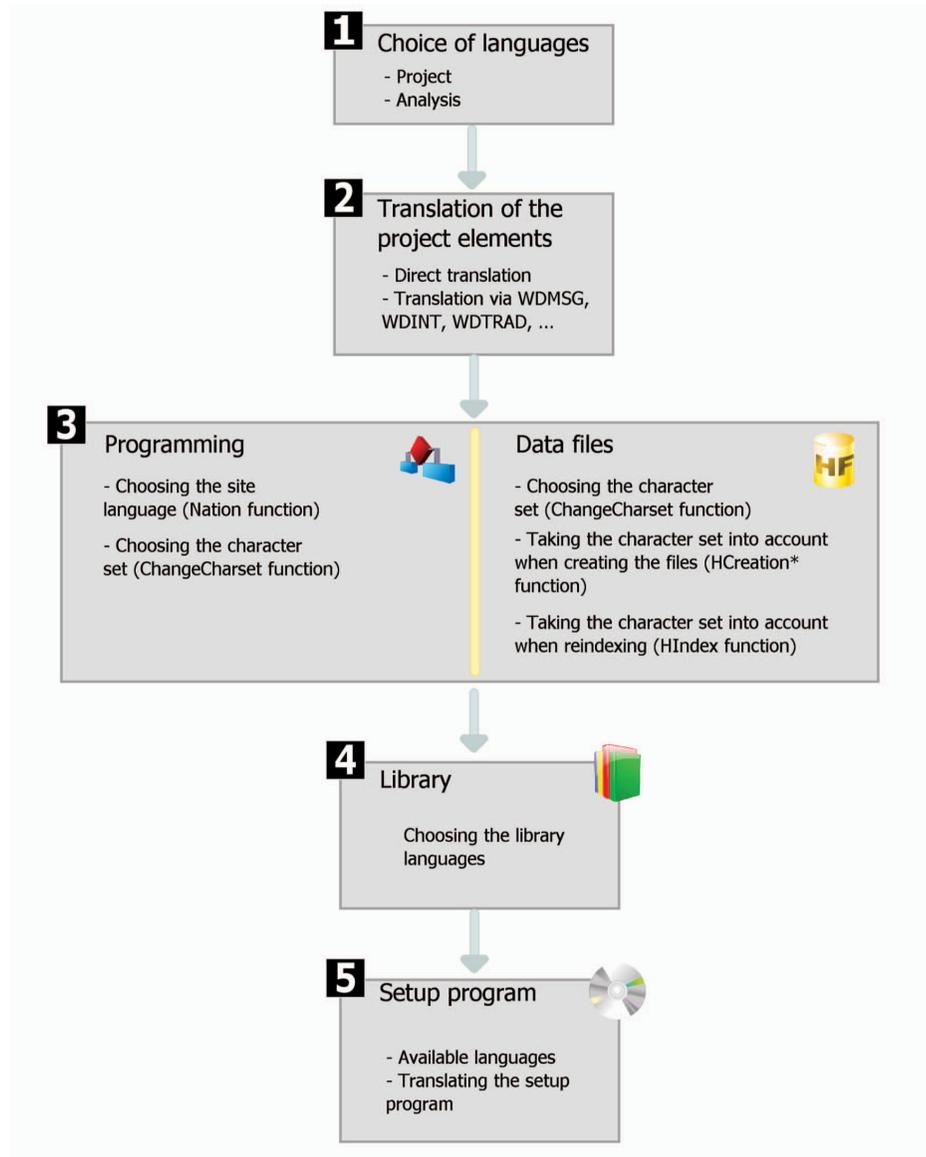
### 7.8 Managing the analyses

If a specific analysis is open while the user groupware is being configured, you need to re-run the `gpwInitAnalysis` function (used to open the user groupware analysis).

**Note:** At run time, several analyses cannot be opened at the same time.

# Multilingual sites

WebDev enables you to easily create some multilingual sites from the same project.



## Multilingual sites in practice

### 1 Overview

A multilingual site can be distributed in several languages. The different languages of the site will be taken into account during the different development steps.

The main steps for developing a multilingual site are as follows:

1. Choosing the languages supported by the project and the analysis.
2. Entering the different project elements (pages, code, ...) in the different languages of the project.
3. Defining the project language by programming.
4. Managing specific character sets in the data files.
5. Creating the library and the setup program.

**Notes:**

- If the operating system supports several languages (Hebrew, Arabic, Greek, ...), the corresponding character set is automatically used when the translations are entered in these languages.
- If your site is a multilingual site, you need to manage this feature in the user groupware pages, and in the pages for automatically managing the HyperFileSQL errors. See the online help for more details.
- Unicode support is available in the HyperFileSQL data files and the page controls.

### 2 Choosing the languages supported by the project and the analysis

#### 2.1 Languages supported by the project

The different languages supported by the project are defined in the project description ("Project .. Project description .. Languages").

The selected languages will be proposed for all the multilingual resources that can be translated (captions of controls, menu options, help message associated with a control, ...).

When the project languages are modified, the modifications will be automatically taken into account by:

- all the new elements or objects created in the WebDev editor,
- all the elements or objects opened in the WebDev editor.

The main language corresponds to the language used by default at run time.

#### Linguistic options

The display of the numeric data found in the project (number, currency, date, ...) can be configured in each language supported by the project.

The parameters used by default are the ones defined in the linguistic options of Windows (accessible from the control panel of Windows).

This setting is performed in the "Languages" tab of the project description ("Project .. Project Description").

**When creating an edit control or a table column** displaying numeric type data (number, currency, date, time, duration, ...) the input mask used will automatically be defined in the project's language options. This option is available in pages.

**At run time**, when an edit control or a table column has a mask "Defined by the project", the input mask and/or the display mask will automatically adapt according to the options selected in the project for the language displayed at run time.

**Note:** The linguistic options are also used to define the writing direction and the character set used ("Various" option).

#### 2.2 Languages supported by the analysis

If your project uses an analysis, the choice of languages supported by the analysis is performed in the data model editor. Indeed, the same analysis can be shared among different projects that do not propose the same languages. Therefore, the number of languages defined for the analysis can be greater than the one defined for the project.

The configuration of the languages supported by the analysis is performed via "Analysis .. Analysis description .. International".

The different languages configured in the analysis will be proposed:

- when configuring the shared information of items. The description of the controls linked to the items (options, captions, ...) can be entered in the different languages supported by the analysis. When generating a "full application RAD" or a page RAD, this information is automatically taken into account by all the languages common to the analysis and the project.

### 3 Translating the interface into several languages

Once the languages supported by the project have been selected, the information displayed by the site must be translated into these different languages.

To translate the interface into several languages, several elements must be taken into account:

- The selected translation mode.
- The supported languages (languages that use a special character set or not)
- The use of specific pages (user groupware, automatic management of HyperFileSQL errors)
- The messages displayed by programming.

#### 3.1 Selected translation mode

WebDev offers several translation modes:

- Automatic translation of the interfaces via a specific translation tool, found on the development computer. This translation is performed directly in the WebDev editor.
- Check out all the messages to translate via WDMMSG and check them back in after translation.

#### Automatic translation of the interfaces

For each object, different multilingual areas are displayed in the description windows of the object. The multilingual areas enable you to enter the information in the different languages supported by the project.

To translate this information from WebDev:

1. Configure the translation options of the software used ("Tools .. Options .. General options of WinDev .. Translation").

- for the information printed in the program documentation (notes of the data file or item).
- for the information managed by "Reports and Queries".

#### 2.3 Languages supported by the different project elements

By default, the various project elements (pages, reports, code, classes, set of procedures, and so on) support the same languages as the project where they were created.

An element may support more languages than the project (when an element is shared among several projects that support different languages for instance).

These options enable you to define:

- the default translation tool (and the transmission mode for the text to translate and for the translated text).
- the source language and the destination language for the translation.

2. If a translation tool is specified, a "Translate" button will be displayed in each element containing information to translate. This button enables you to translate the selected text via the specified translation tool.

3. To perform the translation, all you have to do is select the text to translate and click the "Translate" button.

#### Checking out and checking in the information to translate

WDMMSG is used to extract all the project messages (captions of controls, ...) and check them back in once they have been translated.

Contact PC SOFT Sales Department for more details about the conditions for using this product.

#### 3.2 Languages that use a special character set

If your site supports languages that use special character sets (Greek, Russian, etc.), you must enter the translation of the messages using these specific character sets.

WebDev automatically manages the use of special character sets in the editor.

Indeed, as soon as the edit cursor of the mouse is located in an input area of a language that uses a

special character set, the corresponding input language (character set used by the keyboard) is automatically displayed.

Therefore, if a caption is entered in the Russian section of the caption description, the keyboard will automatically switch to the Russian character set.

Reminder: To use specific character sets, you must install the files corresponding to the requested character sets in the regional options of Windows (control panel).

### 3.3 Translating specific pages

#### Managing the HyperFileSQL errors

By default, the pages for managing the HyperFileSQL errors are provided in English and in French. To translate them into another language, you must:

1. Include the default error pages in your project. The pages are provided (with their WLanguage code) as examples in the \Programs\Data\Presets\Pages\HyperFileSQL - Automatic Help Pages directory.
2. Customize the error management to use the HyperFileSQL pages for error management (*HONError*).
3. Translate the messages (see the previous paragraph).

#### User groupware

By default, the user groupware is supplied in English and in French.

To translate an application or a site that uses the user groupware, you must choose the "Custom" mode when the user groupware is included in your application ("Workshop .. Configure the user

groupware").

The different pages of the user groupware will be included in your project.

Then, all you have to do is translate these pages.

Note: If needed, clear the "auto run" option; this way you'll be able to display a language selection page before the login page.

### 3.4 Translating the messages found in the WLanguage code

In your code, several WLanguage functions allow you to communicate with users via character strings. These messages must also be translated into the different languages of your site.

To translate a character string entered in the code editor:

1. Select the character string to translate.
2. Select "Translate" from the popup menu of the selection. The window for translating messages is displayed.
3. Enter the translation and validate. The code line is displayed in the code editor:

Click on **Button1** \*  
Info ("WinDev is fantastic" )

The flag indicates that translations exist for this character string. Click this flag to display the translation window.

The number (2 in our example) indicates the number of translations entered for this character string.

**Note:** To transform all the messages found in your code into multilingual messages, select "Code .. Multilingual messages .. Convert simple strings into multilingual messages".

### 4 Choosing the language by programming

A multilingual site is an application that can be distributed in several languages. The user will be able to choose the runtime language of the application. You can for example:

- ask for the runtime language during the first start

tup of the application

- implement an option (menu option or button for example) allowing the user to change the current language for the application

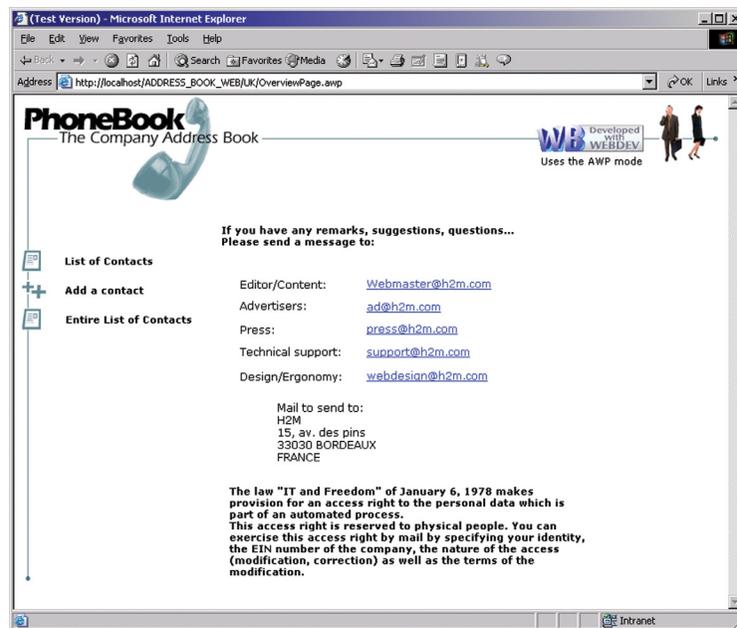
# 10 tips for better interfaces ...

## 1 Polish the look of your site's home page.

The home page is a major element of your site. In some sites, it is just a connection page but in most Internet sites, this is truly the main page for attracting visitors (and future customer of the site). Its content and its presentation must be studied closely. Don't hesitate to read books about user-friendliness of sites to help you.

## 2 Introduce yourself.

When a Web user comes to a site, he must be able to identify the person or the company in charge of the site: display a name, a postal address and an Internet address, ...

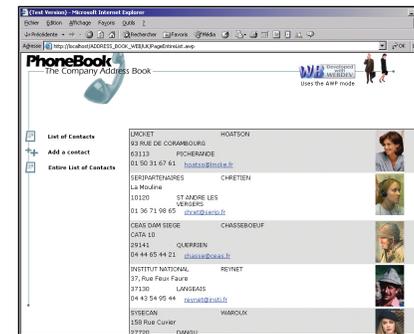


Other helpful information:

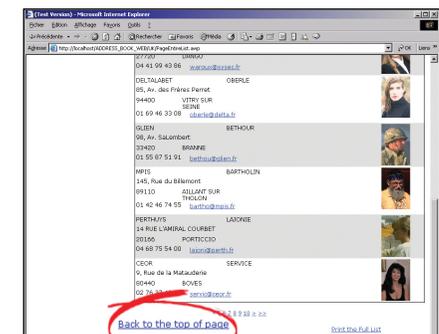
- the person in charge of the company, the registration number of the company ...
- the registration information to the CNIL for sites that collect personal user information.
- the date the site went live or was last updated.
- its purpose, its content and its objective.

## 3 Your pages are very long? Give the web user ways to go back to the top.

If your pages are very long (more than 2 screens), use buttons or links to go back to the top of the page (to avoid using the scrollbar).



Top of the page



End of the page with a link used to go back to the top of the page

## 4 Standardize the look and alignment of your buttons.

Check whether your buttons have the same look and check their alignment. The page editor uses a snap on and alignment mechanism to make this easier.



Buttons not aligned, use different styles

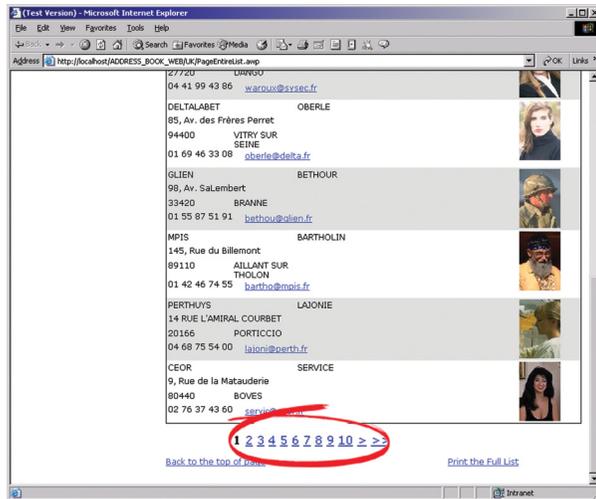
Buttons aligned and standardized

## 5 Avoid using framesets.

Framesets, even though they seem to be more efficient at first look, have specific problems: they are made of several pages. Referencing them with search engines is harder. Navigation (especially the use of the "Back" key) is hard to figure out for users. Today, it is not recommended to use framesets for developing sites.

## 6 Your lists display more than 20 elements? Use several pages to list the elements.

The pager control will let you go from one page to another without having to code anything and the overall interface of your site will be lighter. The site will be more pleasing and faster to view.



## 7 Use images displayed at 100% or Homothetic centered.

To keep the proportions of your images, we recommend you choose:

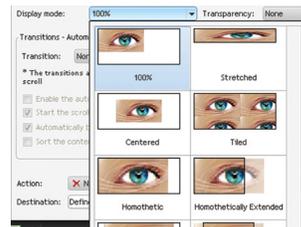
- images displayed at 100% (for static and dynamic images or if the image size is identical to the control size).
- homothetic centered images (for generated images or for the images of different sizes coming from a database): the size of the image will be homothetically adapted to the area defined for the image. The proportions will be respected.



Image at 100 % if the dimension of the control corresponds to the one of the image, otherwise homothetic image



stretched image (not recommended)



Selecting the display mode in the editor (image description, "General" tab)

## 8 You are using images? Test they display properly in your pages.

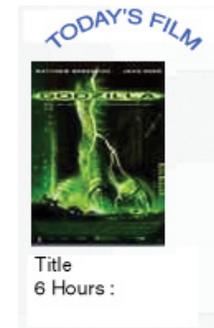


Image in the editor



Image in the browser. Problem displaying the images

If a display problem occurs, check whether:

- the image exists,
- the image is saved in a format recognized by the browser (GIF or JPG for instance),
- the image is found in the sub-directory of the <ProjectName>\_WEB project.

## 9 The user must enter a quantity? Use a combo box to make entering quantities easier.



## 10 You offer a product catalog with the ability to order?

Specify all the information about the product (name, reference, price, etc.) each time, including the button or the link used to order the product.



# Overlaying the controls

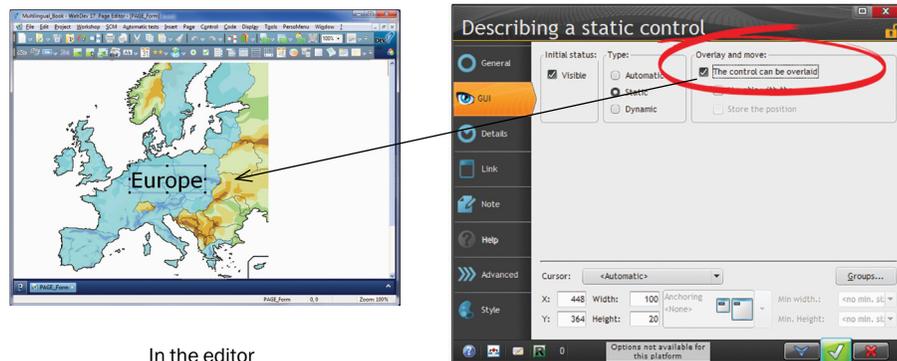
To get special effects (transparency or depth), your controls can be overlaid.

"The control can be overlaid" option is available for each control ("GUI" tab of the control's description window).

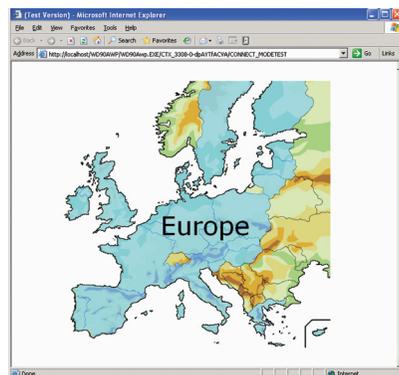
Check this option and the control is no longer linked to the positioning tables found in the page. The control can be moved and placed "above" each control found in your page.

**Caution:** the control with the "This control can be overlaid" option will be on top of the other controls. If several controls have this option selected, the order in which they are stacked up can be defined with the "Up", "Down", "Background" and "Foreground" options of the "Control" menu.

For example, in order for a caption to be displayed on an image, the caption must be made stackable and it must be placed on the image.



In the editor

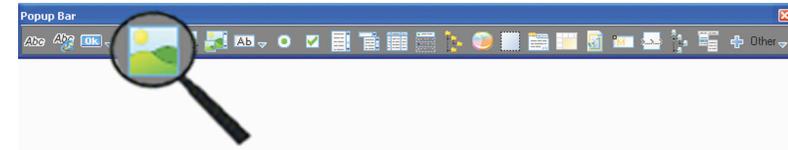


In the browser

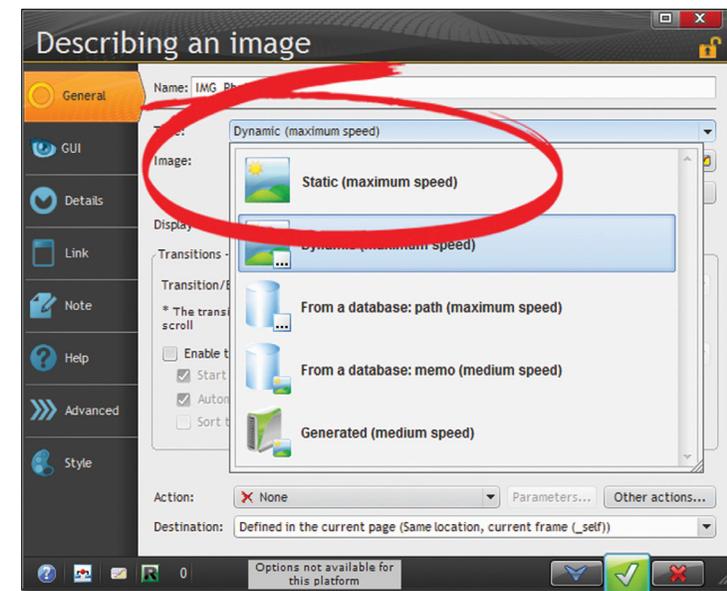
**Note:** Overlaying controls does not work on older browser (older than Internet Explorer version 4 or Netscape 4.7).

# Type of image: dynamic, static, generated, ... ?

The images are often used to illustrate the pages of the sites.

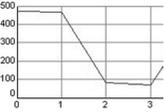


Different types of images can be used in WebDev. Each type corresponds to a specific use of the image.



Here is a summary of the characteristics of these images and how they should be used.

**Advice:** Always favor an homothetic display for images coming from a database.

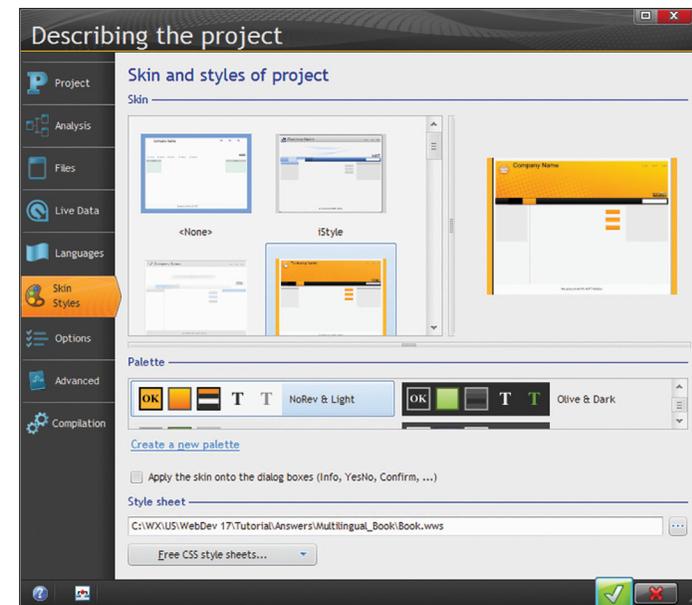
Type of the image	Characteristics
<p><b>Static</b></p>  <p>Image not modified when using the site (company logo, and so on ...) Image modified in browser code only (image selected by an Upload control).</p>	<p>The image associated with a static image must be selected in the WebDev editor. This image can be changed by programming in browser code only. The image formats recognized by WebDev are allowed. When saving the page, the image is automatically converted into GIF, JPEG or PNG format and saved in the &lt;Project Name&gt;_WEB directory (except if the image is already in GIF or JPEG format). When the WebDev site is deployed:</p> <ul style="list-style-type: none"> <li>the file associated with the image must be found on the server.</li> <li>if the image is changed in browser code, the new image must be accessible and in a format recognized by the browser.</li> </ul>
<p><b>Dynamic</b></p>  <p>Image modified when using the site (image of a product in a form, photo in a phone book, and so on) This type of image is the most commonly used.</p>	<p>A dynamic image can be changed by programming in server code or in browser code. The image associated with the control by programming must be:</p> <ul style="list-style-type: none"> <li>found in the &lt;ProjectName&gt;_WEB directory.</li> <li>In a format recognized by the browser (GIF, JPEG or other)</li> </ul> <p>Note: no binary memo can be used.</p>
<p><b>Generated</b></p>  <p>Image drawn when using the site: - charts (grXXX functions) - drawings (dXXX functions) A background image can be used.</p>	<p>A generated image is built while using the dynamic WebDev site. This type of image should not be used to display existing "image" files: the performance of your site might be reduced. When the browser asks to display the image (when displaying or refreshing the page for instance), a temporary image of the current drawing is built on the server and saved in JPG format. This image is displayed in the browser.</p>
<p><b>From a database</b></p> <p>Images of products displayed in a loop</p>	<p>These images are saved in binary memos (memo) or in text controls (path). We recommend that you use a centered homothetic image: all the displayed images will be proportionally resized in order to be displayed in the Image control.</p>

# Customizing a site's look: the skin elements

The graphic look of a site is a major element. With WebDev, it is very easy to get a good looking site with a professional look at first try. To do so, various tools are available to you: preset skin templates, palettes and page templates.

## Skin

A project's skin can be selected in the project property window ("Project .. Project description"). The skins are configured in the "Skin, Styles" tab. A large number of skins are provided with WebDev.



A skin includes:

- a style sheet containing WebDev styles.
- an image directory.
- a default color palette.
- a set of reports used as base to build the new reports of the project.

The skins can be switched at any time to change the graphic look of a site. When you change skin, WebDev automatically adapts the look of all the controls of the site.

The skin of a page can be defined at three different levels (in descending order of priority):

- In the project.
- In the page template from which the page inherits the properties.
- In the page itself.

The general skin must be defined at project level. If a set of pages must use a specific skin (the "Promotion" pages of a site for instance), you have the ability to replace the project skin by redefining it for a page template. If a particular page must have a different skin, this skin can be chosen for the page itself.

### Color palettes

The color palettes define a set of colors that are proposed in all the color pickers of the project. The color palettes are an easy way to customize the visual aspect of a site. These colors are also applied to the styles defined in the skin of the project in order to easily change the visual aspect of a project.

The color palette of a page can be defined at three different levels (by descending order of priority):

- In the project.
- In the page template from which the page inherits the properties.
- In the page itself.

The general palette must be defined at the project level. If a set of pages must use a specific palette (the "Management" pages of a site for instance), you have the ability to replace the project palette by redefining it for a page template. Finally, if a specific page must have a different palette, this palette can be defined for the page itself.

### The preset page templates

The preset page templates provide page structures corresponding to a large variety of sites and favor reusability and graphic harmony among projects. Several templates are supplied with WebDev.

The same preset template can be added several times into a project so that it can be used with different skins or different color palettes for example.

Once it has been added to the project, the preset template can be opened in the editor and modified like any other page template.

A preset page template can also be used as base template for creating another page template.

# Choosing a type of button/link

Several types of button/link are available depending on the operation that must be performed on the page controls. The type of button/link is defined in the control description ("General" tab).

Operation on controls:

- Send the value of controls to the server (submit)
- Reinitializing the page controls (reset)
- None

Let's see some examples for "standard" use of buttons/links in a page:

- **Sending the current page's data to the server for processing**  
Example: Registration form: the link is used to check the validity of the information entered and to display a confirmation page.  
To perform this type of process, use a "Send the value of controls to the server" link. The page data is sent to the server in order to be processed.



- **Performing a process on the server, without retrieving the data of the current page**  
Example: Menu page made of buttons: each button is used to start a page of the site.  
To perform this type of process, use a "None" button.



- **Performing a process in browser code**  
Example: Performing a calculation locally, an input check, ...  
To perform this type of process, use a "None" button.

Country \*:   
 Phone Number:   
 Email \*:



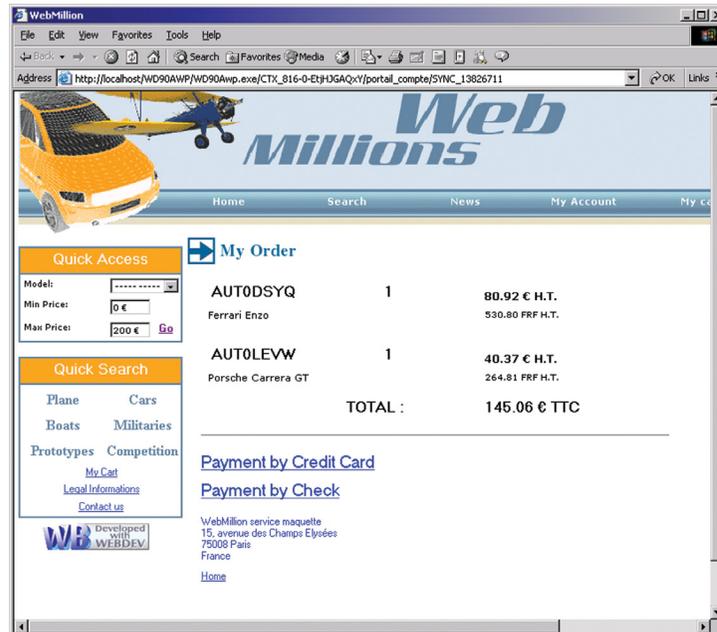
- **Displaying a page external to the WebDev site**  
Example: Starting a search engine via a button.  
To perform this type of process, use a "None" button.



- **Reinitializing the page's controls**  
Example: Link used to clear the controls found in the current page.  
To perform this type of process, use a "Reset page controls" link.

# Preventing going back to a page

The "Back" button of the browser allows the Web user to go back to a page that was already visited. However, it is not always a good idea to go back to a page. For example, in a business site, if the Web user goes back to the page for order validation, he might validate his order twice, and therefore place 2 orders.



To prevent from going back to a page via the "Back" button of the browser:

1. Display the page in the WebDev page editor.
2. Select the "Page .. Description" option. The description window of the page is displayed.
3. Display the "Details" tab.
4. Check "Prevent from using the browser "Back" key to go back to this page".
5. Validate.

# Security management: managing the "Back" button

A browser is used to browse the different pages of a site via "Next" ( → ) and "Previous" ( ← Précédente ) buttons in its menu.

A page context is created on the server whenever a new dynamic page is displayed in the browser.

When the same dynamic page is used to display different data (page with browsing table, loopers or "Form with browse" page), the page context found on the server changes according to the data displayed.

**When the Web user clicks the "Back" button, the browser does not inform the site or the server. The browser goes back to the previous page without the corresponding context.**

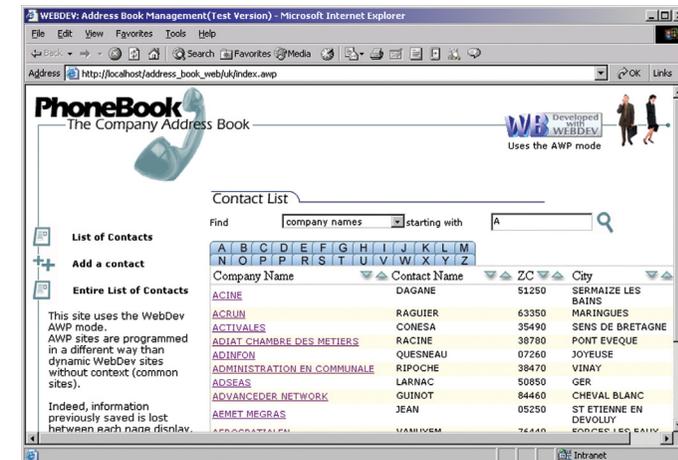
No specific management is required when a "Back" operation is performed between different pages.

A specific management is required when the same dynamic page displays different data.

## Why manage the "Back" button? A simple example ...

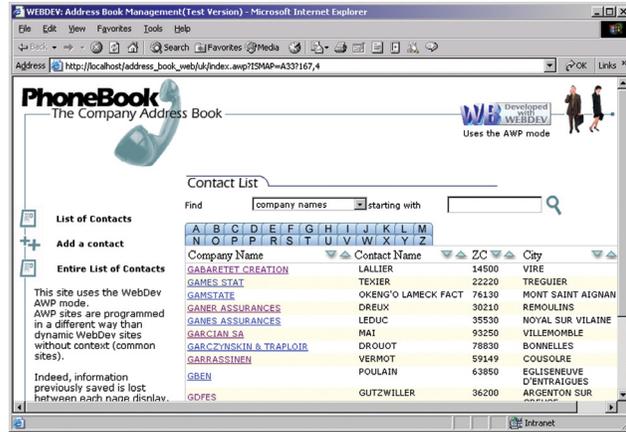
To understand why "Back" needs managing, let's take a look at a simple example: a dynamic page used to display the list of available DVD players.

### Step 1: Displaying the list of contacts



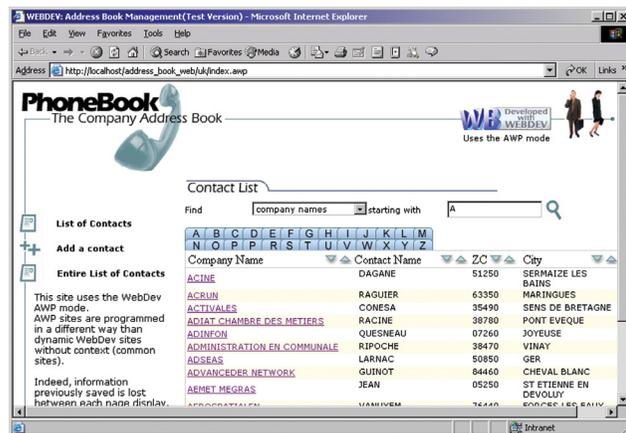
The context contains the list of elements displayed in this table: contacts whose name starts with "A".

### Step 2: Clicking the "G" button.



Contacts whose name starts with "G" are displayed. The page context is updated on the server: it contains the list of elements displayed in this table (contacts whose name starts with "G").

### Step 3: Clicking the "Back" button of the browser.



The dynamic page displayed in the browser corresponds to the page displayed in step 1. However, the browser did not inform the server that the Back button was clicked by the Web user. The page context on the server is still the one corresponding to the contacts whose name starts with "G".

If the Web user selects an element in the table (contacts whose name starts with "A"), the selected element will be the corresponding element in the context: a contacts whose name starts with "G". **The dynamic page viewed is out of sync with the page context found in the server. This desynchronization is detected by the WebDev engine.**

**This is the reason why the Back button must be managed.**

### How to manage the "Back" button?

To manage the "Back" button, you can use:

- the **synchronization code of the dynamic page**. This code is called when desynchronization occurs.
- a **hidden edit control** (invisible edit control) to identify the record displayed in the browser.
- the **"Disable the mechanism for page synchronization for this control" option** on your buttons and/or links ("Advanced" tab of the button description).

You can choose and combine one or another of these options depending on the type of action that can be performed in the dynamic page.

### Detailed example of Back management

A dynamic page is used to browse the different products of a site. The buttons found in the page are as follows:

- **Return to main menu**  
This action requires no specific management of the "Back" button. The button does not handle the data files: desynchronization between the context and the page displayed has no effect.
- **Delete the current product**  
The "Delete the current product" action handles a data file. The deleted record must correspond to the record displayed by the Web user. To manage the "Back" button, we recommend that you use:
  - a hidden control in the dynamic page: this control will contain the identifier of the record actually displayed in the browser.
  - the synchronization code: this code is used to update the context (search for the record viewed by the web user).

The actions performed in the synchronization code will be:

1. Re-read the record corresponding to the identifier stored in the hidden control.
2. If the record is not found: an error page is displayed and the code of the "Delete" button is not run.
3. If the record is found, the page context is automatically modified on the server: the code of "Delete" button is run.

Total security!

The "Address Book" example, supplied with WebDev, illustrates the case presented in this page.

# Managing the Back button in practice

## 1 Overview

The "Back" button of the browser allows the Web users to display the pages that have already been visited.

In a WebDev site, each HTML page displayed on the browser is associated with a page context, found on the server. Each action performed in a page displayed by the browser must trigger an identical action on the corresponding page context found on the server.

However, the browser "Back" button is used to perform an action on the browser only: the page displayed in the browser and its context on the server can become "out-of-synch" if the browser "Back" button is used.

### 1.1 There are two ways you can manage the browser "Back" button

To avoid any out-of-sync problems between the pages displayed on the browser and the corresponding contexts found on the server, WebDev proposes two methods for managing the browser "Back" key:

- **Solution 1:** preventing the return to this page with the browser "Back" button.  
If the browser "Back" button is used to display the previous page, this action will have no effect.
- **Solution 2:** Synchronization management (default solution)  
For each action performed in a page from the browser, a synchronization test is automatically performed between the HTML page and its context.  
Two modes are available for managing the synchronization:
  - default synchronization (mode used by default when creating a new page).
  - programmed synchronization.

### 1.2 Example of desynchronization

Site example:

- A browser page contains a browsing table linked to the ITEM file and a "Next" link.
- The ITEM file only contains a single item, each record is made of a letter in the alphabet.
- The page is used to display 6 table rows, the "Next" link is used to display the next 6 rows.

When opening the page, the table displays the 6 first records of the file (from 'A' to 'F'). Sequence of actions performed by the user:

#### 1. Click the "Next" link

**Result:** the server points to the next 6 records of the ITEM file and returns their contents to the browser. The browser displays the next page of the table with the 6 new contents ('G' to 'L').

#### 2. Click "Back" in the browser

**Result:** the browser displays the page preceding the first action. The table displayed contains the letters 'A' to 'F'. The server was not contacted, therefore it is still positioned on the records 'G' to 'L'.

#### 3. Click "Next"

**Result:** the server points to the next 6 records in ITEM (that is 'M' to 'R'). The browser synchronizes with the server and displays the same elements: the web user has the feeling that some information is not displayed.

This behavior can lead to unexpected consequences when modifying a file record (modification of another record than the one viewed by the user for example).

**Reminder:** each action in the browser must lead to an action on the server: the server then sends an answer to the browser. However the click of the "Back" button is a browser action independent of your WebDev site, therefore the second condition may not be performed.

## 2 Disabling the "Back" button

If the browser "Back" button is used to display the previous page, this action will have no effect.

### 2.1 Implementation

Disabling the "Previous page" feature of the browser triggers the insertion of the following Javascript code into the generated HTML page:

```
<SCRIPT LANGUAGE="JavaScript">
  history.forward()
</SCRIPT>
```

When viewing this page in a browser window, it will not be possible to go back to this page when clicking the "Back" button of the browser.

#### Notes:

- Using the "Back" button of the browser can make the page blink.
- This mechanism can fail if the [STOP] button of the browser is clicked before the forward() statement is run by the browser.

### 2.2 Implementation

To disable the "Back" button of the browser for a specific page:

1. Display the "Details" tab found in the description window of the page ("Description" from the popup menu of the page).
2. Check "Prevent from using the browser "Back" key to go back to this page."
3. Validate.

To disable the "Back" button of the browser for a frameset:

1. Display the "Details" tab of the description window of the frameset ("Description" from the popup menu of the frame).
2. Check "Prevent from using the browser "Back" key to go back to this frameset".
3. Validate.

To disable the "Back" key of the browser for all the pages of the project:

1. Display the description of the project ("Project .. Project description").
2. Click the "Options" tab.
3. Check "Prevent from using the browser "Back" key to go back to the page".
4. Validate. This option will be automatically taken into account for all the new pages of the site.

## 3 Managing the synchronization

### 3.1 Overview

For each action performed in a page, the mechanism for page synchronization automatically checks the synchronization. This test is designed to check whether the page displayed in the browser corresponds to the page context found on the server.

Two modes are available for managing the synchronization:

1. Managing the default synchronization.
2. Managing the synchronization by programming, in the synchronization code of the page.

### 3.2 Default synchronization

The synchronization mechanism is triggered only if "Use the mechanism for synchronizing pages" is selected for the page.

If a desynchronization occurs, a warning message informs the Web user that the requested action has not been performed. The page corresponding to the context found on the server is redisplayed. The site can continue to operate.

To implement the management of the synchronization in a page:

1. In the "Details" tab of the window description ("Description" option of the popup menu), check "Use the page synchronization mechanism".
2. Validate. This page will be automatically included in the history of browser pages: to go back to this page, use the "Back" key of the browser.

To implement the management of synchronization in all the project pages:

1. In the project description ("Project .. Project description", "Options" tab), select the "Use the mechanism for synchronizing pages" option.
2. Validate. The project pages will be automatically included in the history of browser pages: it will be possible to go back to these pages using the browser "Back" button.

Notes:

- This management mode requires no specific WLanguage code.
- The synchronization mechanism can be disabled for the page controls that do not require synchronization management ("Close" button for example: simply check "Disable the mechanism for page synchronization for this control" in the "Advanced" tab of the control's description window.
- The warning message can be customized (see next paragraph).

## 4 Synchronization by programming

To manage the synchronization by programming:

1. In the "Details" tab of the window description ("Description" option of the popup menu), check "Use the mechanism for synchronizing pages".
2. Configure (if necessary) the page controls for which the synchronization must not be managed.

For each control that triggers an action on the server, you can specify whether the page synchronization must be managed (default option) or ignored during this action.

To ignore the synchronization management, check "Disable the mechanism for page synchronization for this control" in the "Advanced" tab of the control description.

3. Enter the code required for custom management of the synchronization in the synchronization code of the page. Use **ChangeAction** in the synchronization code of the page. This function is used to define the action that will be performed in case of page desynchronization.

Notes:

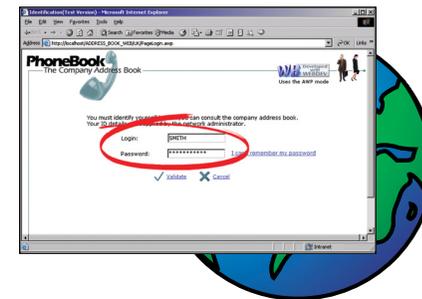
- The function named **ChangeAction** is initialized with the "Do nothing" action if a WLanguage function that displays or refreshes a page is used in the synchronization code of the page.
- To customize the desynchronization message, all you have to do is enter in the synchronization code of the page:
  1. the custom message.
  2. the code for refreshing the current page on the server (with **PageRefresh** for example).
- To perform a synchronization from the information found on the computer of the Web user, we recommend that you:
  1. Use a hidden control to store the identifier for the record displayed and selected.
  2. In the synchronization code, find the current record on the browser. This search is performed from its identifier found in the hidden control.
  3. Refresh the page.

# Cookies: information stored on the computer of the user

A cookie is a file saved by the WebDev site on the computer of the Web user. This file can be read at any time from the server or by the browser. The cookies are used to store persistent information on the computer of the Web user.

For example, information is requested when a Web user connects to a site for the first time: name, login ... and it is stored on his computer. At the next logon, the WebDev site will read this information in the cookie created during the first visit and it will be able to greet the Web user personally on the first page.

### 1st connection



The new user must enter his name and his password

The WebDev site saves the information given by the Web user on his computer.



### Next connections

The WebDev site reads data previously saved on the computer of the Web user.



By default, the WebDev site displays the information read in the cookie

To create cookies in a WebDev site, use the **CookieWrite** WLanguage function. To read the cookies in a WebDev site, use the **CookieRead** WLanguage function.

# Protecting the access to the site: passwords

Managing the access to a site by password is required in two cases:

1. Protecting the personal details
2. Restricting the access to a site

## Protecting the personal details

To access his personal details, the Web user can specify a password. In a business site, a password defined by the Web user allows him to access the details of his previous orders.

In this case, the management of site access requires:

- identification: when he first connects, the Web user identifies himself and enters his password. In subsequent connections, the user provides his ID and accesses the features reserved to him (his history, ...).
- the ability for a web user to retrieve a forgotten password.

**WebDev is supplied with all the resources required to manage personal passwords.**

## Restricting the access to a site

By default, an Internet site can be accessed by any Web user. But a site is not always intended for the general public: some features may be reserved to some types of Web users (site administrator, and so on).

In this case, the site manager assigns a password to each Web user. This password cannot be modified. This password gives access to specific features.

For example, in a business site:

- a specific password will be assigned to salespeople: they will be able to access the sales statistics, ...
- a specific password will be assigned to the members of the marketing team: they will be able to online new products.

**WebDev allows you to easily manage these different types of access to a site.**

# TLS/SSL: encrypting information on the web

By default, the data exchanged between the computer of the Web user and the Web server is not encrypted. The authenticity of the server identity, the confidentiality and the integrity of the data exchanged are not guaranteed.

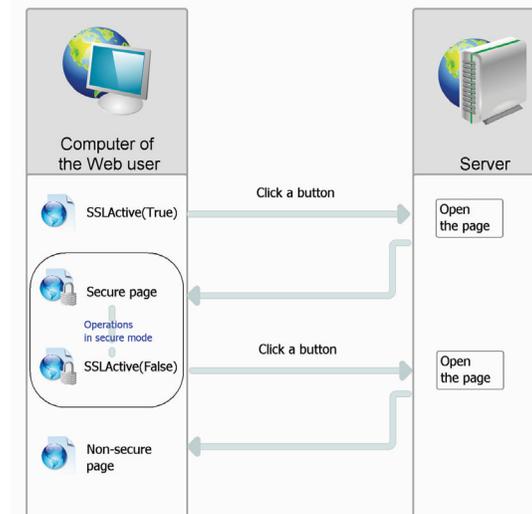
To insure confidentiality, the most used method is the TLS/SSL protocol (Transport Layer Security, formerly Secure Socket Layer). This protocol guarantees the identity of the accessed server and encrypts the data exchanged between the server and the browser.

## Implementing secured transactions via the TLS/SSL protocol

To implement secured transactions via the TLS/SSL protocol, you must comply with all the requirements linked to this protocol (purchase of a certified SSL key from a specific organization or generation of a self-signed key for private use or for test). See the online help of WebDev for more details.

## Transactions secured by TLS\SSL in a WebDev site

Usually, only the transfer of sensitive data must be secure: transfer of a credit card number for instance. Indeed, the secured mode is slower. However an increasing number of sites operate totally in TLS/SSL mode thanks to more powerful machines, which insures a better confidentiality for the user operations.



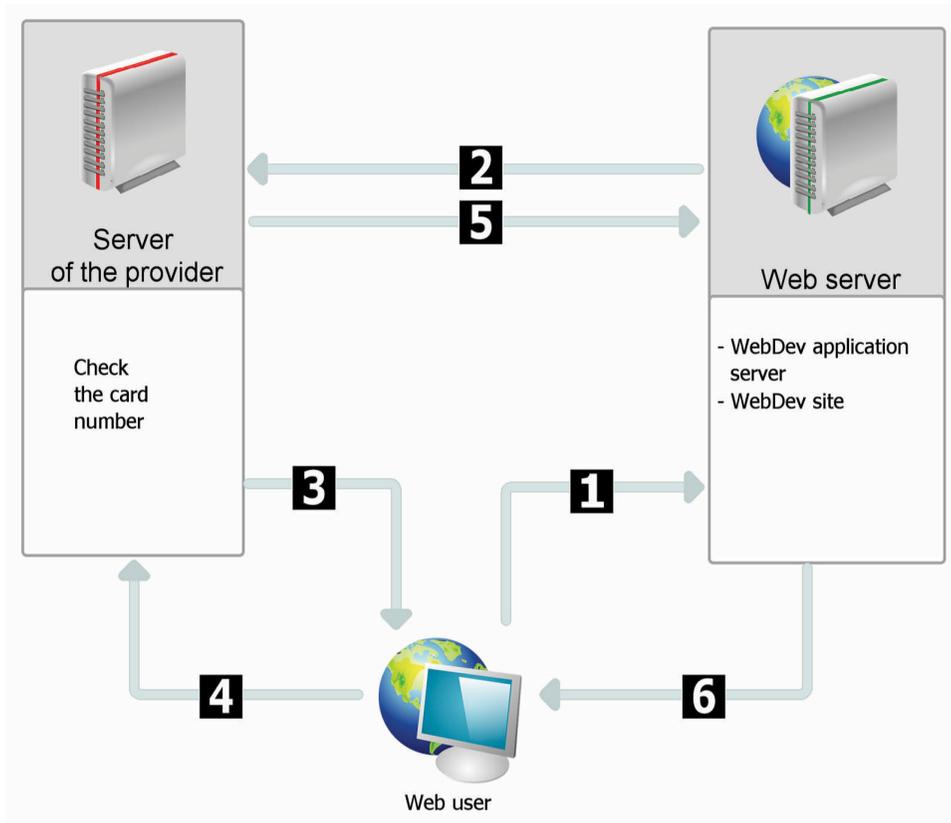
The secured mode is implemented when opening a new page via a button (or a link). To do so, use **SSLAActive** in the browser code of the button (or link). As soon as the secured page is opened, all the actions will be performed in secured mode (which means encrypted).

To go back to the non secure mode, simply use **SSLAActive(False)** in the browser code of a button (or link) that opens this new page.

# Secure payment with provider

Several providers allow you to implement systems for secure payment. The provider validates all the credit card payments with the card center.

The payment steps are as follows:



1. The customer validates his order.
2. The WebDev site requests debit from the provider.
3. The provider asks for the card number.
4. The customer enters his card number.
5. After checking it, the provider sends an authorization number.
6. The order is accepted.

# Sending emails

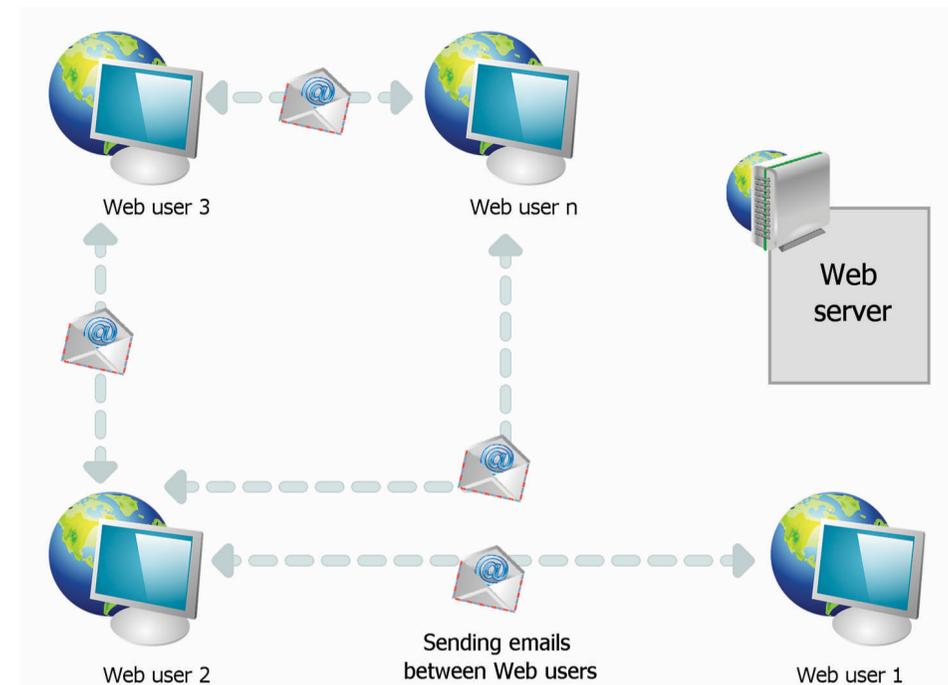
In a web site, emails allow:

- the web users to communicate among themselves: send emails to the site manager, mailshot, ...
- to validate an order: in a business site, an email is sent to the web users to validate their order.
- to transfer data from one computer to another: the new orders entered in a business site are transferred by email to the company hq, ...

Two methods are available to send emails from a site.

## 1 Sending the email from the computer of the Web user

Use this method when the Web user must send an email to a specific address: author of the site, technical support, sales department, ... The *EmailOpenMail* browser function is used to open the email software of the Web user. Some parameters can be filled by default: email address of the recipient, ...



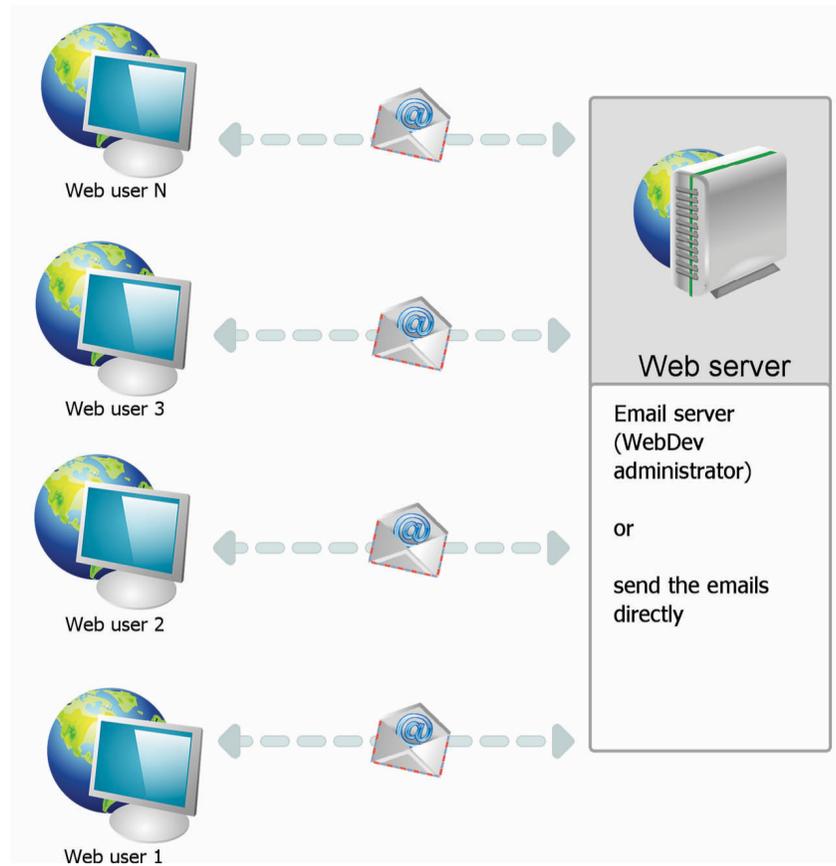
This method uses the email software of the Web user: no specific process must be performed in the WebDev site.

## 2 Sending the email from the server (dynamic pages only)

In this case, the server manages the emails.

The emails can be sent:

- in the dynamic WebDev site directly (to transfer data by email or to validate an order by email for instance);
- by the email spooler (supplied with WebDev).



Use this method for a custom management of emails:

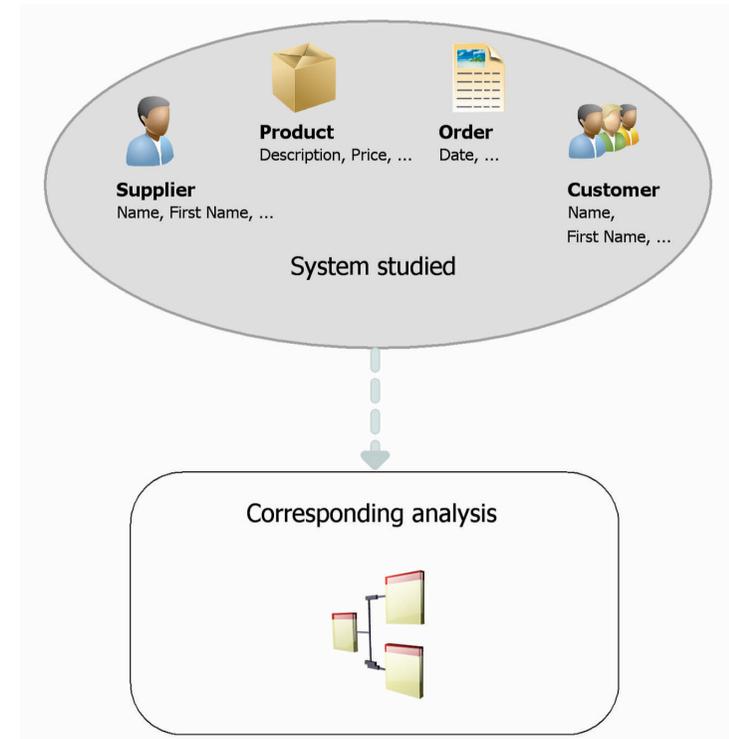
- checking the content of the emails sent by the web users ("mailing-list" site),
- checking the validity of the recipient address,
- encrypting the information sent by email,
- validating a process, ...



# Analysis: Structure of the database

When a WebDev, WinDev or WinDev Mobile project uses data files, this project must be associated with an analysis. An analysis allows you to describe the structures of the data (files, items, ...) used in your project.

The data model editor allows you to easily create an analysis.



The analysis of a WinDev project corresponds to the LDM (Logical Data Model). The entire structure and data organization are described: the data is grouped by file. Each file contains several data called item.

In the analysis, the description of a data file can be linked to a type of data file (HyperFileSQL, Oracle, ...).

# The analysis in practice

## 1 Overview

When a project uses data files, this project must be associated with an analysis. An analysis allows you to describe the structures of the data (files, items, ...) used in your project.

Two methods are available to describe an analysis:

- **1st method:** Creating the analysis directly (which means a Logical Data Model or LDM). This operation is presented in details in "Creating

a LDM (Logical Data Model)", page 166.

- **2nd method:** Creating the Conceptual Data Model (CDM) and creating the analysis from the CDM. These operations are presented in details in "Creating a CDM", page 171 and "Generating the LDM from the CDM", page 173.

This chapter presents both methods.

## 2 Creating a LDM (Logical Data Model)

In WebDev, the terms "LDM" and "Analysis" are interchangeably used to define the structure of the database associated with a project.

### 2.1 Creating an analysis (or LDM)

To create a LDM:

1. Select "File .. New .. Analysis". The wizard for analysis creation starts.

2. Specify:

- **the name and directory of the analysis.** The analysis corresponds to a ".WDA" file. By default, this file will be created in the directory of the project analysis (<Project Name>.ANA directory). This directory must be accessible in read/write.
- **the caption of the analysis** that briefly describes the purpose of the analysis.
- **whether the analysis must be associated with the current project.**
- the types of databases handled by the project.

3. The creation of the first data file is automatically proposed.

4. Create all the elements (data files, items and links) of your analysis.

### 2.2 Adding a data file to the analysis

The analysis is used to define the structure of the data file used by the project. A data file found in the analysis can be:

- A new data file.
- A preset data file, provided with WebDev.
- A data file imported from an existing database (using a specific file format for instance).

The following paragraphs explain how to create a data file.

To create a new data file:

1. Select "Insert .. Data file". The wizard starts.
2. Check "Create a new description of data file".
3. Specify:

- **the logical name of the data file.** This name will be used to handle the file.
- **the data file's caption**, which briefly summarizes the object of the data file.
- **the representation of a record** in the data file. This representation improves the meaning of the questions asked when describing the links. This option must be preceded by an indefinite article (A or AN).
- whether the data file includes an **"Automatic identifier" item**. The value of this item is unique for each record and it is automatically calculated by WebDev.

- **the type of the database for which the data file will be created.** Depending on the selected type, this data file will be handled by the HyperFileSQL engine, by an OLE DB driver or by one of the WebDev native accesses (SQL Server, Oracle, ...).

- **whether the size of the file can exceed 2 GB.** This option can only be used if both the file system of the server computer and the file system of the computers where the data files are installed by WebDev support the NTFS format (Windows 2000, NT or XP).

**Caution:** If this option is checked, this data file cannot be opened in Windows 95, 98 or Me.

- **whether the data file must support the replication.** This option is used to manage the automatic updates of identical and remote databases. See the online help for more details.

4. The created data file becomes the current data file. The description window of the items of the data file is automatically opened. It enables you to describe the items of the data file.

To import a preset data file description:

1. Select "Insert .. Data file".
2. Check "Select a description among the preset data files".
3. Choose the preset data file. This data file will be imported into the current analysis. This data file can be modified later.
4. Select the file items to keep. These items can be modified later.
5. Specify whether the links must be automatically sought. If this option is checked, the items with the same name will be linked.
6. The imported data file is automatically inserted into the current analysis.

**Note:** To modify a data file or the items of a data file, select the data file and enable "Structure of files .. Description of data files" or "Structure of files .. Items".

To import the existing description of a data file, there are two solutions:

**Solution 1: from the data model editor**

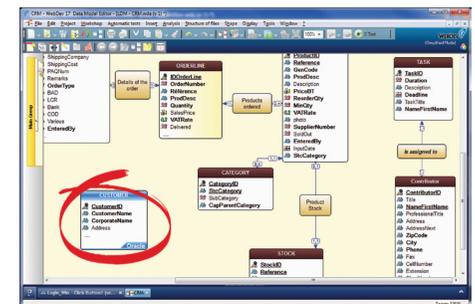
1. Select "Insert .. Data file".
2. Check "Use the data files of an existing database" then select the type of the database.
3. Specify the data format used by your project. You can keep the current format or migrate the files to HyperFileSQL format.
4. If required, specify the WebDev analysis where the description must be imported (existing or new analysis).
5. Specify the source database containing the descriptions to import and the type of this source database. Depending on the selected type, specify the requested information.
6. Select the tables or files whose description must be imported and validate.
7. The imported data file is automatically inserted into the current analysis.

**Solution 2: From Windows explorer**

Directly drop the data file description from Windows explorer into the data model editor. For example:



Here: an Oracle database on the server. All you have to do is drag the name of the database ...



... to transfer its description into the data model editor.

### 2.3 Creating an item

To create an item:

1. Double click the data file in which the item must be created. The description window for the items of the data file is opened.
2. Click the first empty row in the table of items.
3. Specify the name, caption and type of the item in the table.
4. In the right section of the screen, specify the details about the new item (type, size, default value, sort direction ...).
5. In the bottom section of the screen, specify the details about the shared information.
6. Validate the description of the item.

**Note:** you also have the ability to create an item from the meta-types proposed by WebDev. To do so, click the "+M" button on the right of the table. The list of available meta-types is displayed.

### 2.4 Creating a link

Different types of links can be created between data files. See for more details "Characteristics of the links defined in an analysis", page 168.

**To create a link:**

1. Select "Insert .. Link". The mouse cursor turns into a pen.
2. Select the two data files to link. The description window of the link is automatically opened.

3. To define the cardinalities:

- select the cardinalities among the proposed ones (0,1; 1,1; 0,N; 1,N).
- answer the questions asked. The cardinalities will be automatically updated.

4. To describe the advanced cardinalities, check "Display the advanced cardinalities" and answer the questions asked.

5. Enter the caption of the link by briefly describing the purpose of the link.

6. Specify the keys to link.

7. Specify the integrity rules. These rules are used to ensure the data integrity when one of the relation keys is modified or deleted.

8. Validate. The link is automatically created.

## 3 Characteristics of the links defined in an analysis

### 3.1 Owner file and member file

When a link is defined between two data files, an **owner** file and a **member** file are found:

- the owner file is the owner of the key.
- the member file is a member of the analysis files containing a copy of the key.

To manage the link between two data files, the key of the owner file is copied into the member file.

For example, the key of the Supplier file is copied into each record of the Product file. Several records of the Product file can have the same key in the Supplier file:

- the owner file is the Supplier file,
- the member file is the Product file.

This type of link is represented as follows:



### 3.2 The cardinalities

The cardinalities are used to count the links between the data files.

The cardinality is defined according to the answers to the two following questions:

1. For each record in the data file, what is the minimum number of records in the other data file it is linked to? The answer provides the first part of the cardinality (minimum cardinality):

- if the answer is "none", the cardinality is 0,X.

- if the answer is "a single one", the cardinality is 1,X.

2. For each record in the data file, what is the maximum number of records in the other data file it is linked to? This answer provides the second part of the cardinality (maximum cardinality):

- if the answer is "a single one", the cardinality is X,1.
- if the answer is "several", the cardinality is X,N.

The answer to these two questions defines the cardinality that can be: 0,1; 0,N; 1,1; 1,N

The description of cardinalities is fundamental: it provides the basis for referential integrity of the database.

To be clearer, these two questions can be broken down into four questions. For example, to describe a link between the Supplier file and the Product file:

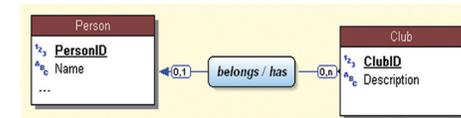


- each "supplier" has at least one "product": Yes/No?

- each "supplier" can have several "products": Yes/No?
- each "product" belongs to at least one "supplier": Yes/No?
- each "product" can belong to several "suppliers": Yes/No?

**Example of cardinalities:**

This example presents two different cardinalities:



- **Cardinality 0,1:** A person can be a member of only one sport club. This person is not required to belong to a sport club.
- **Cardinality 0,N:** A club can have no member or several members.

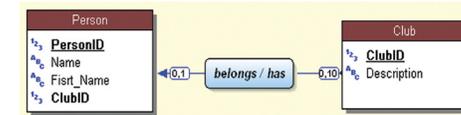
### 3.3 The advanced cardinalities

The cardinalities can be defined more precisely. We talk about advanced cardinalities.

These cardinalities are used to exactly specify the minimum cardinality and the maximum cardinality. To define the advanced cardinalities, check "Display the advanced cardinalities" in the description of the link ("Insert .. Link").

**Example of advanced cardinalities:**

This example presents two different cardinalities:



- **Cardinality 0,1:** A person can be a member of only one sport club. This person is not required to belong to a sport club.
- **Cardinality 0,10:** A club can have up to 10 members.

### 3.4 The referential integrity

The referential integrity of a database corresponds to the respect of the constraints implied by the links between the data files.

The referential integrity consists in checking that:

- if a record is deleted from the owner file, the corresponding records are also deleted from the member files,
- if a record is added into a member file, a corresponding record exists in the owner file,

- if a record is modified in the owner file, the unique key is not modified, ...

The check of the referential integrity depends on the nature of the link between the data files.

See the online help for more details.

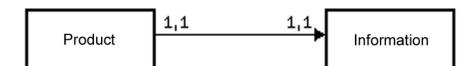
### 3.5 The different types of links

Several types of links can exist between the data files:

- parallel
- optional
- complement
- shared
- complex

#### Parallel link

For a parallel link, **each record** found in a data file (Product) is linked to **a record** found in another data file (Information), and **conversely**



To manage a parallel link, the identifier of the Product file is copied into the Information file. This identifier is also a unique key in the Information file. The owner file is Product, the member file is Information.

**Note:** This type of link is rare because the two files can be combined into a single file.

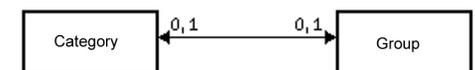
The parallelism of the records in the file is respected if the following operations are run simultaneously on the two files:

- creating a record
- deleting a record
- reindexing with compression

#### Optional link

For an optional link:

- **Each record** found in a data file (Category) **is associated with no record or with a single record** in another data file (Group).
- **Each record** of the other data file ("Group") **is associated with no record or with a single record** in the first data file ("Category").



An optional link is performed by copying the identifier of each data file into the other data file.

### Complement link

For a complement link:

- Each record of a data file (Product) is linked to no record or to a single record in another data file (Details).
- Each record of the other data file (Details) is necessarily associated with a record in the first data file (Product).



The complement links are quite common. They are used when a record can have additional optional information.

To manage a complement link, the key of the Product file is copied into the Details file. To ensure the maximum cardinality of 1, it remains a unique key. This key's uniqueness forbids inserting more than one record in the Detail file for a record in the Product file.

The owner file is Product, the member file is Details.

### Shared link

For a shared link, the same record in a data file (Supplier) can be shared by several records in another data file (Product).

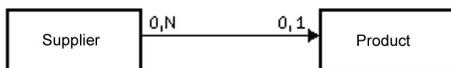
To manage a shared link, the key of the Supplier file is copied into the Product file. It becomes a multiple key to increase the speed of integrity check.

The owner file is Supplier, the member file is Product.

Depending on the cardinality, we can distinguish between four types of shared links:

- Shared link with a 0,n - 0,1 cardinality
- Shared link with a 0,n - 1,1 cardinality
- Shared link with a 1,n - 0,1 cardinality
- Shared link with a 1,N - 1,1 cardinality

#### Shared link with a 0,N - 0,1 cardinality



With this type of link:

- an owner may have no member (a supplier does not necessarily have a product).
- a member may have no owner (a product does not necessarily have a supplier).

#### Shared link with a 0,N - 1,1 cardinality



With this type of link:

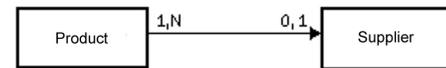
- an owner may have no member (a supplier may have no product).
- each member has a single owner (each product has a single supplier).

**Note:** This type of link is common. WebDev lets you automatically create the pages allowing you to manage the data files linked by a 0,N - 1,1 type link.

#### Shared link with a 1,N - 0,1 cardinality

With this type of link:

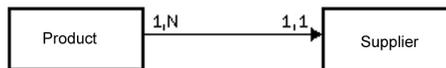
- each owner has at least one member (a supplier has at least one product).
- a member may have no owner (a product may have no supplier).



#### Shared link with a 1,N - 1,1 cardinality

With this type of link:

- each owner has at least one member (each supplier has at least one product).
- each member has a single owner (each product has a single supplier).



### Complex link

For a "complex" link, you must manage a link file, called *relationship file*

The relation file will be automatically created. It will contain a unique key containing the two keys of the linked files.

The relation file can also contain information specific to the link.

A complex link is made of two shared links.

#### Example of complex link

An order (Orders file) can contain one or more products. A product (Product file) can be used in several orders.

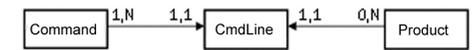
In this case, a link file is required (OrderLine file).

The OrderLine file contains:

- a unique key containing the keys of Product and Orders.

- the number of products ordered.

The links between the data files are as follows:



## 4 Operations available on an analysis

WebDev enables you to perform the following operations on an analysis:

- Duplicating/Copying a LDM: Duplicating a LDM allows you to have two identical LDMs with different names.
- Deleting a LDM.
- Renaming a LDM.
- Associating a LDM with a project.
- Enlarging or reducing the display of a LDM in the editor.
- Moving the display of the LDM in the editor.

WebDev lets you perform the following operations on a data file found in the data model editor:

- Duplicating/copying a data file.
- Deleting a data file.
- Renaming a data file.

WebDev allows you to perform the following operations on an item of a data file found in the data model editor:

- Duplicating/Copying an item.
- Deleting an item.
- Rename an item.

See the online help for more details.

## 5 Creating a CDM (Conceptual Data Model)

In the MERISE methodology, the conceptual data model defines the objects and the actions that must be taken into account in the database. This model only represents the structures of information and the exchanges of information for the system used, independently of the physical constraints linked to the nature of the database used.

WebDev automatically offers to create the first entity.

4. Create all the elements (entities, associations, properties and relationships) of you CDM.

### 5.2 Creating an entity

To create an entity:

1. Select "Entity .. New entity". The description of the entity is displayed.
2. Specify:

- **the name of the entity.** This name will be the logical name of the file created when the LDM is generated from the CDM.
- **the caption of the entity** that summarizes the subject of the entity.
- **whether an "Automatic identifier" property** must be defined in the entity. The value of this property is unique for each record and it is automatically calculated by WebDev.
- **the notes** presenting the operating mode of the entity ("Various" tab).

3. The entity becomes the current entity. WebDev automatically offers to describe the properties of this entity.

### 5.1 Creating a CDM

To create a CDM:

1. Select "File .. New .. Data .. CDM". The wizard for creating a CDM starts.
2. Specify:

- **the name and directory of the CDM.** The CDM corresponds to a ".MCD" file. By default, this file will be created in the directory of the project analysis (<Project Name>.ANA directory). This directory must be accessible in read/write.
- **the caption of the CDM,** which briefly describes the object of the CDM.
- **whether the CDM must be directly associated with the current analysis.** This analysis can be an existing analysis or a new analysis that will be created.

3. The created CDM becomes the current CDM.

### 5.3 Creating an association

#### To create an association:

1. Select "Association .. New association". The description of the association is displayed.

2. Specify:

- **the name of the association.** When the LDM is generated from the CDM, this name will be the logical name of the created file if this association contains at least one property or if it belongs to a complex link.

- **The caption of the association,** which briefly summarizes the subject of the association.

- **The notes** presenting the operating mode of the association.

3. The association becomes the current association. WebDev automatically proposes to describe the properties of this association.

### 5.4 Creating a property

#### To describe a property:

1. Double-click the entity or the association in which the property must be created. The window for describing the properties is displayed.

2. Click the first empty row in the table of proper-

ties or click "+".

3. Specify the name, the caption, the type, the size and the notes of the property.

4. For an entity property, check "Identifier" if the property is the identifier of the entity.

5. Validate the description of the property.

### 5.5 Creating a relationship

#### To create a relationship:

1. Select "Entity .. Draw a relationship". The mouse cursor turns into a pen.

2. Select the two entities to link or the entity and the association to link.

3. Define the cardinalities of the relationship.

### 5.6 Defining the cardinalities

#### To define the cardinalities:

1. Select the relationship whose cardinalities must be defined.

2. Select "Entity .. Information about the relationship".

3. Choose the cardinality among the proposed ones.

4. Validate the description of cardinalities.

## 6 Main operations that can be performed on a CDM

WebDev enables you to perform the following operations on a CDM:

- Deleting a CDM.
- Renaming a CDM.
- Specifying a zoom factor to reduce or enlarge the display of the current CDM.
- Moving the display of CDM.

WebDev enables you to perform the following operations on a CDM entity:

- Duplicating/copying an entity.
- Deleting an entity.
- Renaming an entity.

WebDev enables you to perform the following operations on a CDM association:

- Duplicating/Copying an association.
- Deleting an association.
- Renaming an association.

WebDev enables you to perform the following operations on a CDM property:

- Deleting a property.
- Renaming a property.

See the online help for more details.

## 7 Generating the LDM from the CDM

To associate a CDM to a project, you must generate the associated LDM.

Reminder: It is the LDM that is associated with the project (not the CDM).

The generation of the CDM is used to automatically create the corresponding LDM once the validity of the CDM has been checked.

#### To generate the LDM from the CDM:

1. Select "Analysis .. Generate the logical model".

2. Define the parameters for generating the LDM from the CDM:

- **the name and path of the generated LDM.** The LDM corresponds to a ".WDA" file. By default, this file will be created in the directory of the project analysis (<Project Name>.ANA directory). This directory must be accessible in read/write.

- **whether the captions of the associations without property** (of the CDM) are stored to identify the links (of the LDM).

- **the type of access to the data file.** Depending on the selected type, the data files will be handled by the WebDev engine, by an OLEDB driver or by one of the native accesses of WinDev (SQL Server, Oracle, ...).

- **the type of the database** used.

- **the format of the characters** used in the analysis.

3. Validate the generation. The LDM corresponding to the CDM is automatically created.

#### Special case: Updating an existing LDM

If the name and path of the LDM specified when generating the CDM correspond to an existing LDM, the existing LDM will be updated with the new description of the CDM.

When updating an existing LDM, the generation checks:

1. The existence of the entity in the LMD:

- if the entity is new, the corresponding data file is

created.

- if the entity already exists, its properties are updated.

2. The existence of the properties in the LDM:

- if the property is a new one, the corresponding item is created.

- if the property is already found, with the same description, no action is performed.

- if the property already exists, with a different description, a conflict occurs. The data model editor proposes:

- to ignore the new description of the property and to perform no modification.

- to update the existing item with the new description of the property.

- to create a new description synonym with the new description of the property.

3. The existence of relationships and associations. When the association requires that one or more relationship files be created, the LDM data files and items are updated as described below. When a relationship has been modified, the link of the LDM is recreated or updated:

- if the identifying properties of one of the entities have been modified, the cardinality of the existing links are updated and new links are created.

- if no identifying properties of any entity has been modified, cardinalities are simply updated.

#### Notes:

- The deletion of entities, relationships or associations is not taken into account when a CDM is generated into a LDM.

- The deletion of the entity properties can be taken into account by unchecking "Keep the LDM modifications" in the options for generating the LDM from the CDM. In this case, the files of the LDM only contain the items of the corresponding entity in the CDM after the conversion.

## 8 Generating the analysis (LDM)

The generation of the analysis is performed before the programming phase and after the description of the data files. This generation is used to:

- validate the modifications made to the analysis (LDM).
- create the modules required for programming.
- automatically update the data files if necessary.

As long as the analysis (the LDM) has not been generated, the description of the analysis (i.e. the

data files) cannot be used in the project.

The generation ("Analysis .. Generation") includes three steps:

- checking the modifications made and generating the data files of the analysis description.
- automatic modification of the accessible data files (files found in the "EXE" directory of the project).
- Synchronizing the project.

## 9 Managing the analysis versions

WebDev enables you to manage the different versions of the analysis ("Analysis .. Manage the versions").

You have the ability to work with the current analysis or with a specific analysis.

**Note:** when an analysis is damaged, you can directly select the damaged analysis to restore one of the previous versions for instance.

All the versions of the analysis are viewed in a graph. The yellow rectangle indicates the generation number of the analysis; the caption on the right indicates the generation date.

The "small rectangles" displayed between two versions indicate:

- one rectangle: few modifications have been made between the two versions.
- several rectangles: several modifications have been made between the two versions.

**To see the details of the modifications performed between two versions:** Double-click the line containing the "small rectangles" or click the [Modifications] button.

The options available in the version manager are:

- Restoring an analysis.
- Canceling the last generation.
- Resetting the version number to 1.

### 9.1 Restoring an analysis

To restore an analysis from the version manager:

1. Select the version to restore.
2. You can:
  - restore the version into the specified directory. An independent analysis corresponding to the selected version is created. This option is selected by default.
  - overwrite the current analysis.
3. Validate. The restore operation is performed.

**Caution:** you will not be able to open data files corresponding to an analysis with a version number greater than the restored version. In this case, you must also restore the files corresponding to the version of the restored analysis or delete the existing files in order to re-create them.

### 9.2 Canceling the last generation

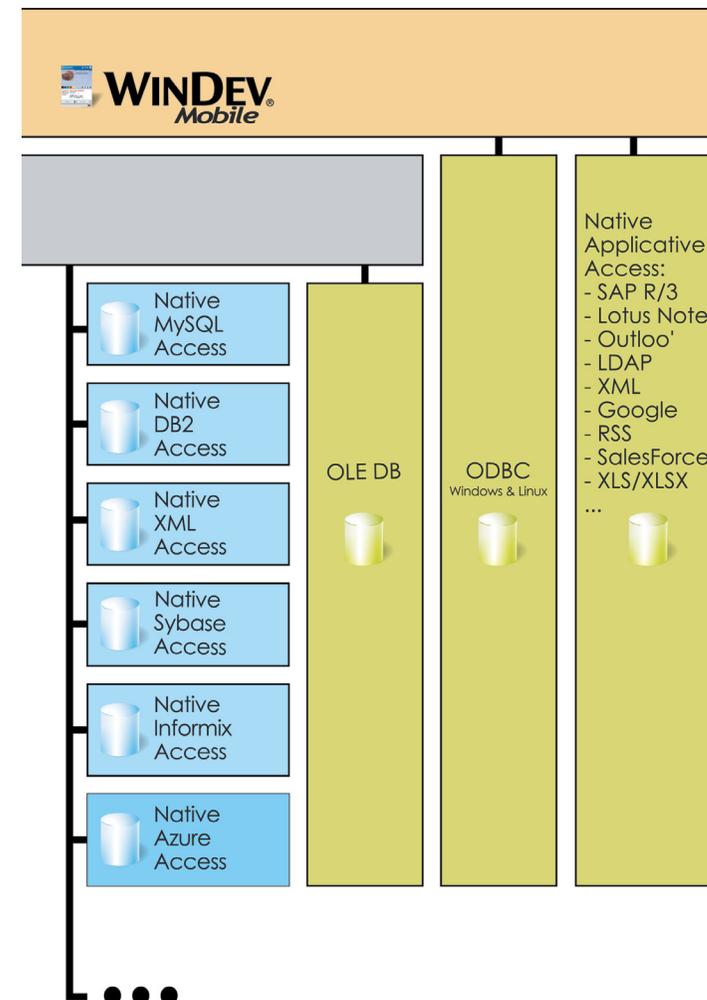
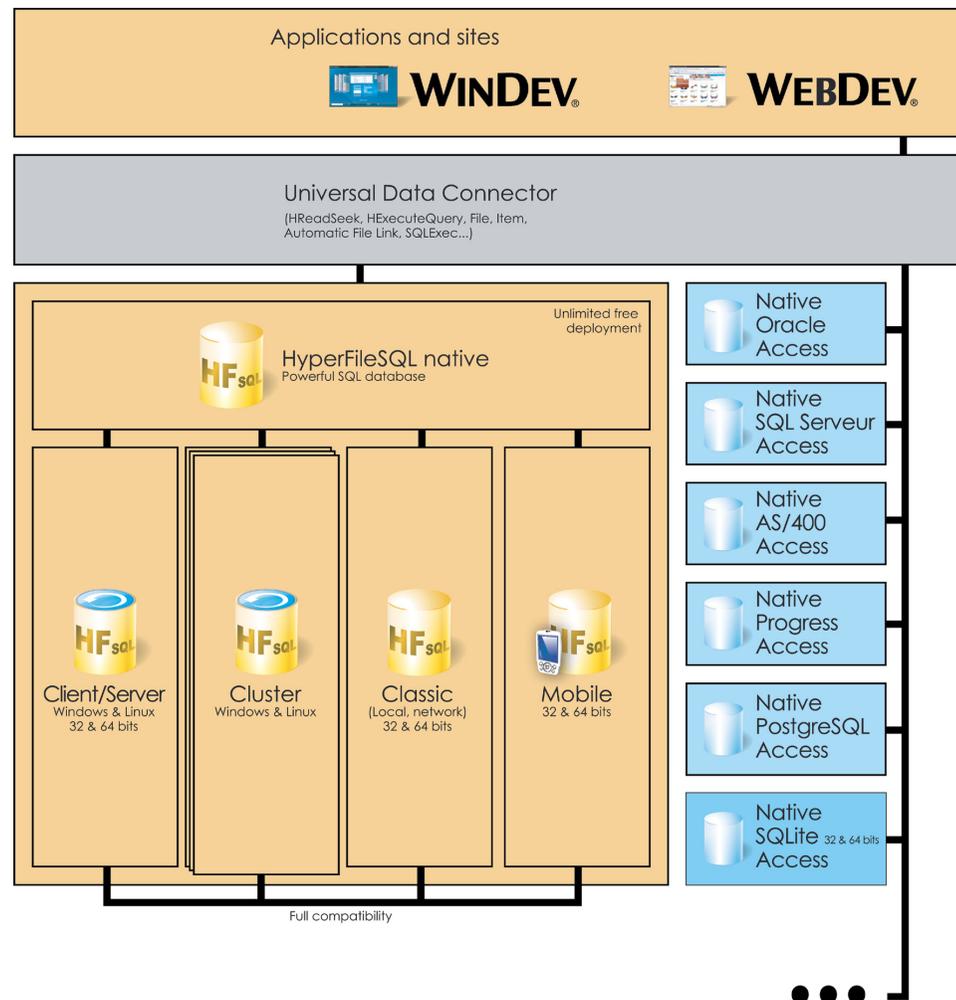
Canceling the last generation of the analysis is used to restore the analysis to its status before the last generation. The modifications performed since are not applied.

### 9.3 Resetting the version number to 1

The generation number of the analysis can be reset to "1" ("Analysis .. Reset the analysis version to 1"). In this case, the version number of the analysis is reset to one. No specific action is performed on the data files.

# The different types of accessible files

WebDev, WinDev and WinDev Mobile propose a simple access to most of the databases on the market.

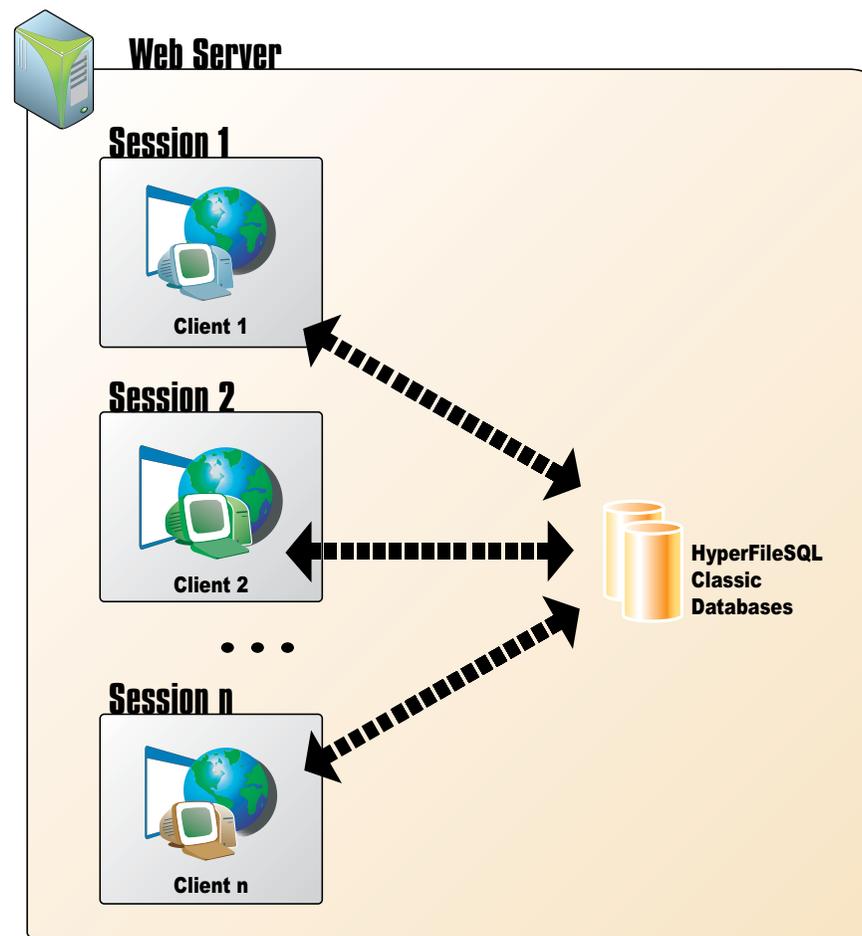


# HyperFileSQL Classic

A WebDev HyperFileSQL application can operate in Classic mode (called HyperFileSQL Classic) or in Client/Server mode.

The characteristics of the Classic mode are as follows:

- A site that uses HyperFileSQL Classic is run on different browsers. A session is started on the server for each site run.
- The data files are found in a directory that can be accessed by the sessions of the WebDev sites (on the Web server or on another computer). Each session physically accesses the data files.
- The processes (query, read/add into a data file, and so on) are performed by each session.

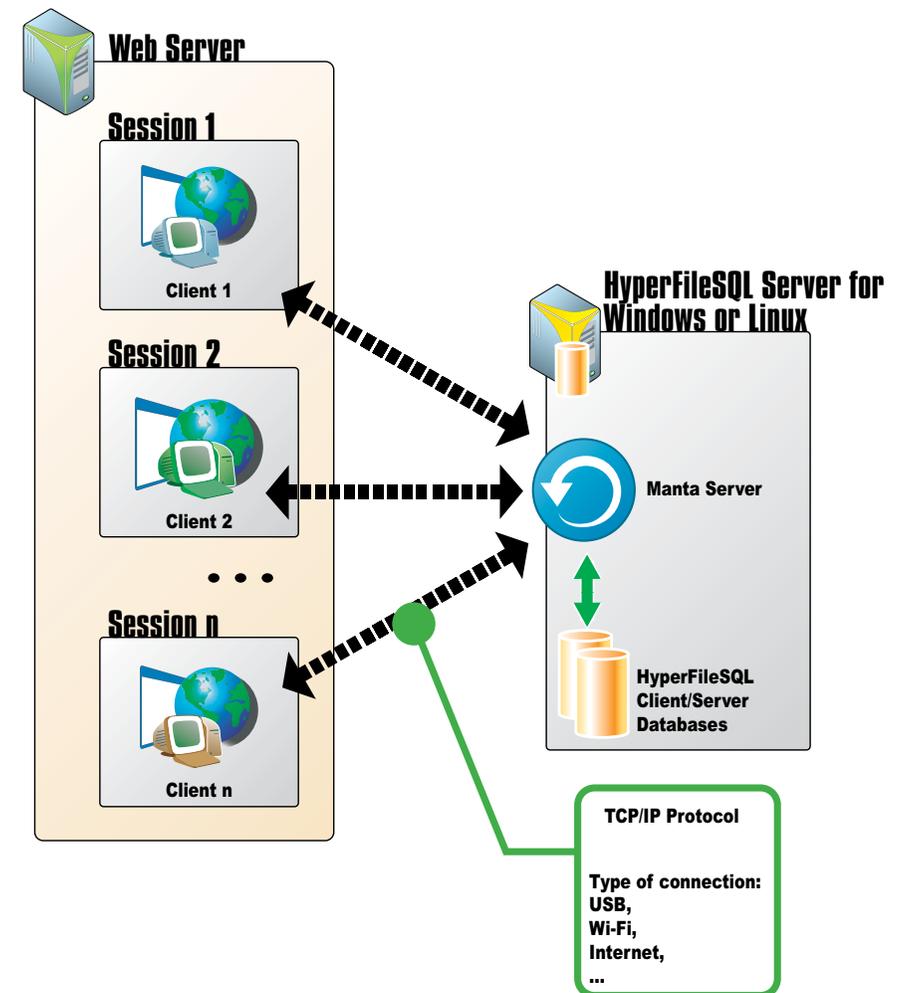


# HyperFileSQL Client/Server

HyperFileSQL Client/Server is a powerful system for managing relational databases (RDBMS) in Client/Server mode.

The characteristics of the Client/Server mode are as follows:

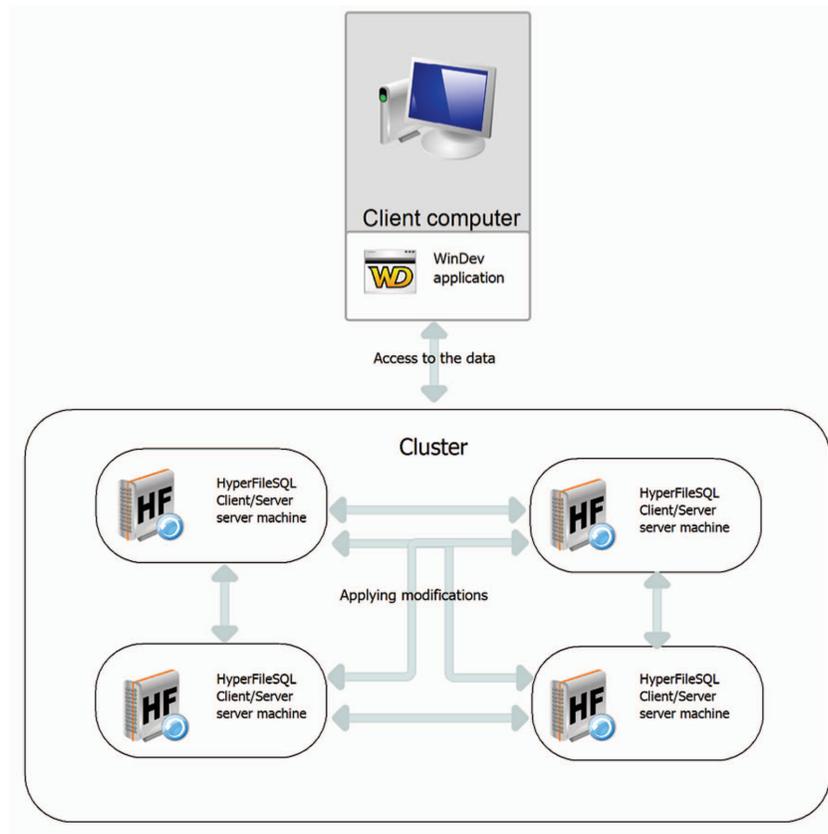
- A HyperFileSQL Client/Server site is run on different user computers. A session is started on the Web server for each site run.
- The data files are found on a server (HyperFileSQL server). Only the server physically accesses the data files.
- All the processes (query, read/write operation in a data file, ...) are performed on the server.



# HyperFileSQL Client/Server clusters

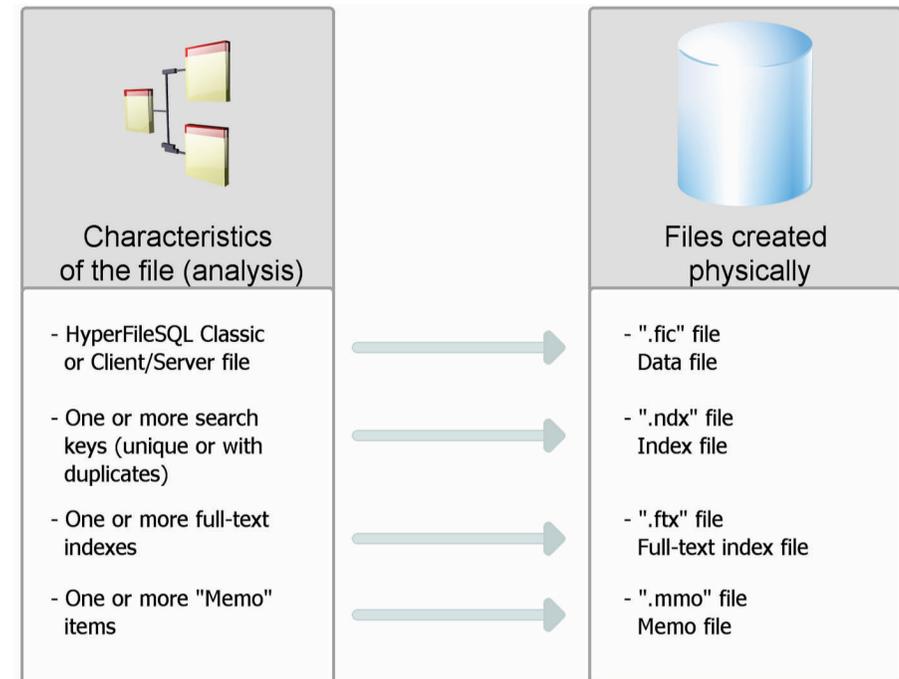
HyperFileSQL Cluster is an extension of the HyperFileSQL Client/Server database. In a database cluster, all the different HyperFileSQL servers contain a copy of the databases and they are synchronized in real time.

- The read load can be balanced among the different servers.
- The physical configuration can evolve without any interruption for the client computers.
- If one of the servers crashes, the client is automatically redirected to an operating server.



# HyperFileSQL: the files physically created

The data model editor is used to describe the structure of the data files. Different files are physically created according to the information entered in the data model editor.



Note: This diagram presents the main created files only. Some specific files can be created if the data file uses logs, transactions or replication.

# Associating the controls with the data

A page can display information coming from:

- a database: the controls are directly linked to the items found in the files or queries available in the database.
- variables found in the code of the application (variables global to the page or to the project or parameters passed to the page).

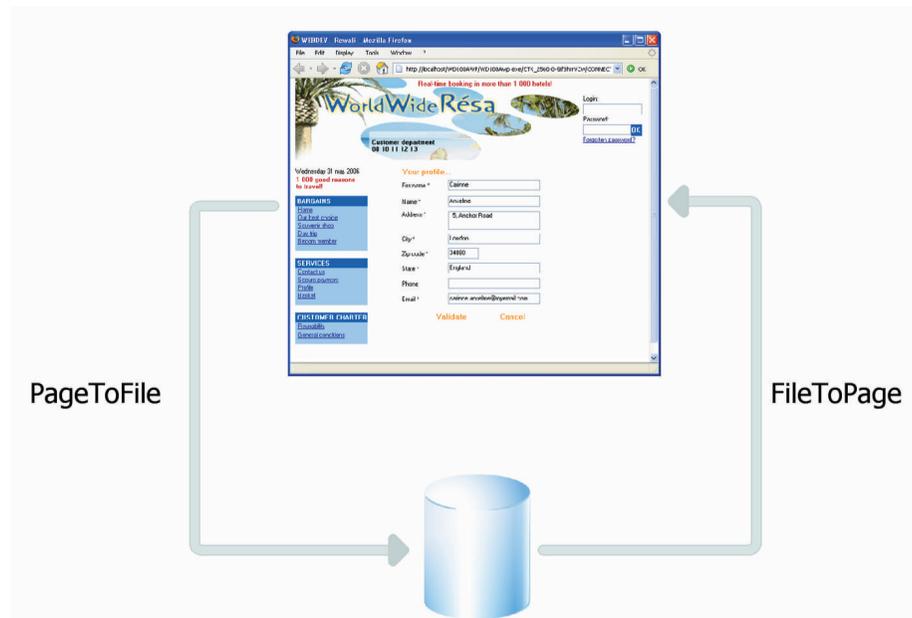
To display this information in a page, the controls of this page must be linked to:

- the different items of the database.
- the available WLanguage variables.

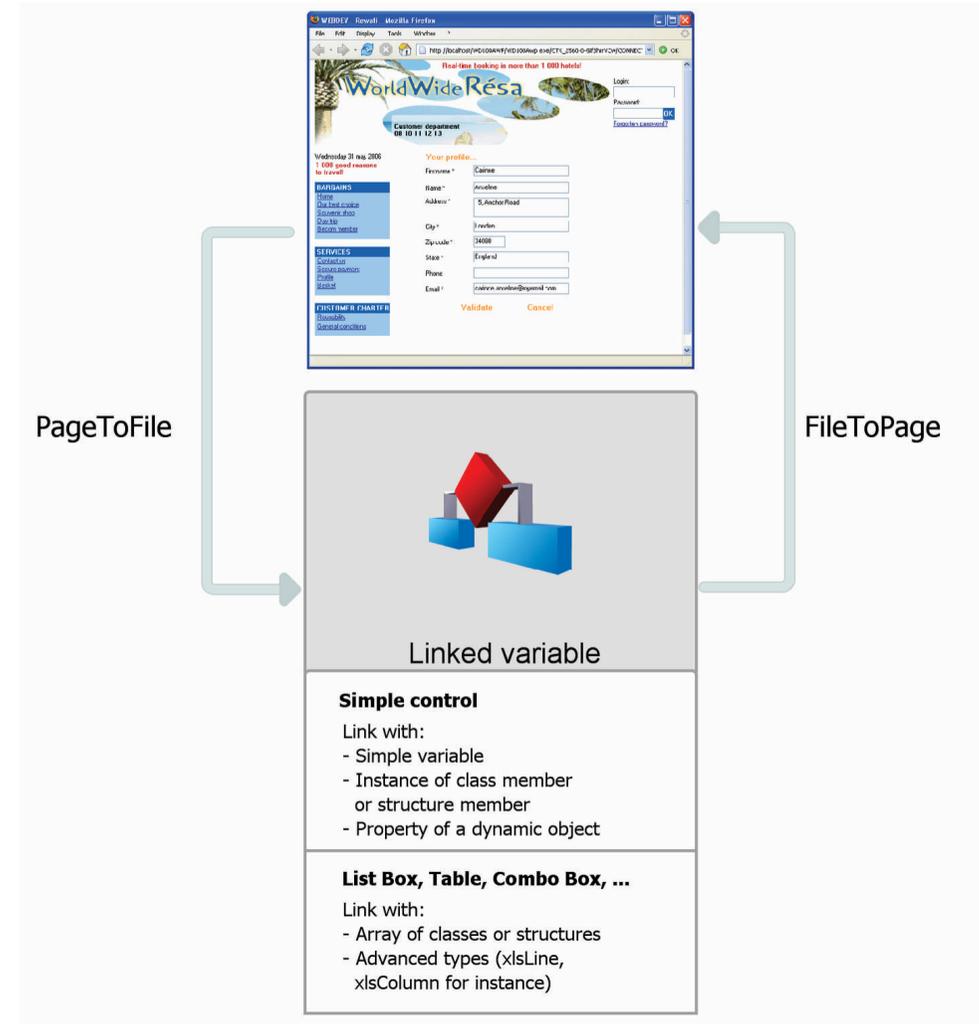
The method for displaying and retrieving the information is straightforward:

- The link between a control and an item or variable is defined in the page editor during the control description ("Link" tab).
- **PageToFile** is used to update the record or the variable with the data found in the page.
- **FileToPage** is used to update the data displayed in the page with the information saved in the data file or with the information saved in the variable.

## Link between control and item



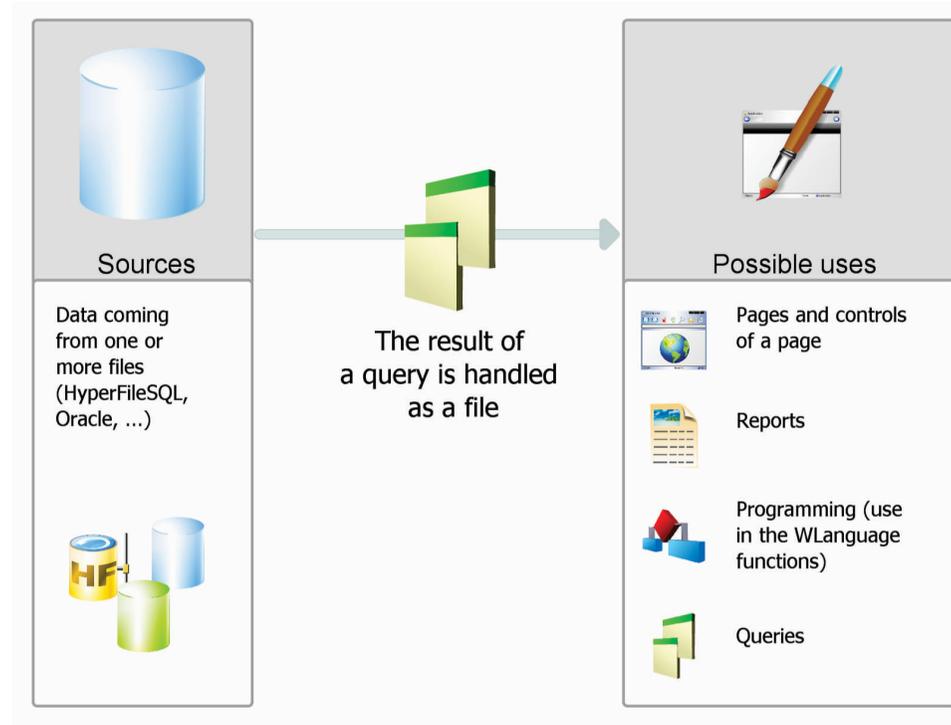
## Link between control and variable



# The queries

A query is used to interrogate a database in order to view, insert, modify or delete data. The structure of the query defines the data used.

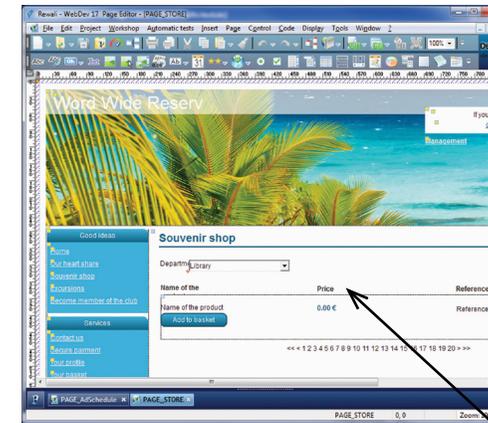
A query can interrogate one or more data files.  
The query editor enables you to easily create queries without programming.



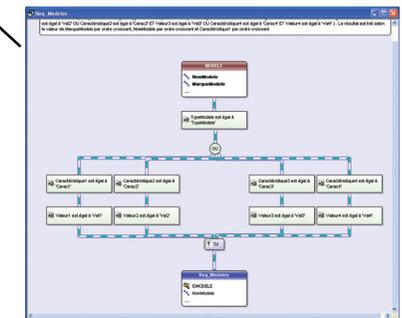
**Note:** In programming, a query can be handled like a data file. Especially, it can be associated with a display control (a table for example) that will present the data returned by the query.

# Embedded queries

The controls found in a page can be linked to a data file or to an existing query, ... These controls can also be linked to a query created when the control was designed.



Looper linked to an embedded query



Embedded query: MyPage\_1\$query

In this case, the query is included into the page. It is found in the WWH file corresponding to the page. If the WWH file is copied (into another project for instance), the embedded queries used by this page will also be copied.

# The Table/Looper control

The Table/Looper controls can be used to display a set of information (the content of a data file for example). The content of these controls can come from three different sources:

- "Direct access file" table/looper
- "Memory" table/looper
- "File loaded in memory" table/looper

**Note:** These three fill modes will be presented in details for the Table control. The same concepts apply to the Looper control.

## "Direct access file" table

A browsing table with direct access is used to directly display the data coming from a data file or from a query. The structure of the file was described in the data model editor and the data was entered in the site.

Reading the data file lets you display the data in the table. For each line displayed, the data file is read: the record read is displayed in a table row.



The number of records displayed in the control can be limited by a filter (*HFilter* used in the initialization code of the control).

Several WLanguage functions can be used to handle the browsing tables. These functions start with "Table".

**To add a record to the table**, add the record to the corresponding HyperFileSQL data file (*HAdd*) and redisplay the table with *TableDisplay*.

**To delete a record from the table**, delete the record from the corresponding HyperFileSQL data file (*HDelete*), then redisplay the table with *TableDisplay*.

## Memory table

A memory table is used to directly display the data loaded in memory. The data is added into the table by programming (by *TableAddLine* for example).



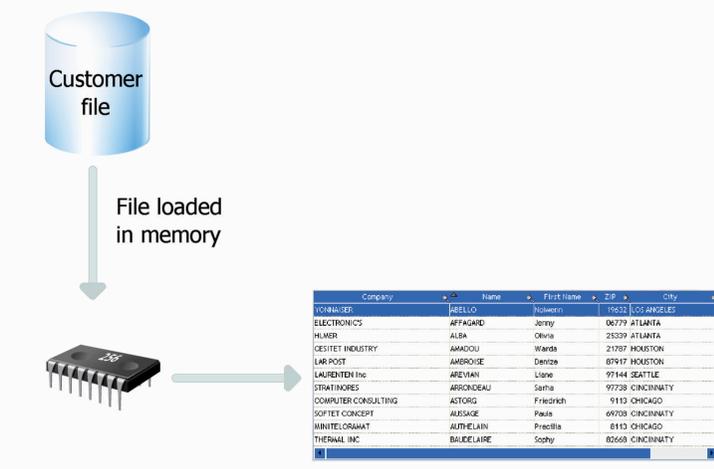
## Browsing table loaded in memory

The browsing tables loaded in memory combine the benefits of the browsing tables and the benefits of the memory tables.

The table is linked to the data file but the content of the data file is entirely loaded in memory. The sort and the search are available on all the columns.

The data not linked to the data file is kept when handling the scrollbar (Check Box column for example).

The records found in the data file being loaded in memory, this type of table is recommended for data files containing less than 10 000 records (to avoid memory overflow).



**Note:** The different fill modes (memory, browsing, browsing loaded in memory) are available for the list boxes, the combo boxes, the tables, the loopers, ...

# Retrieving data from a site

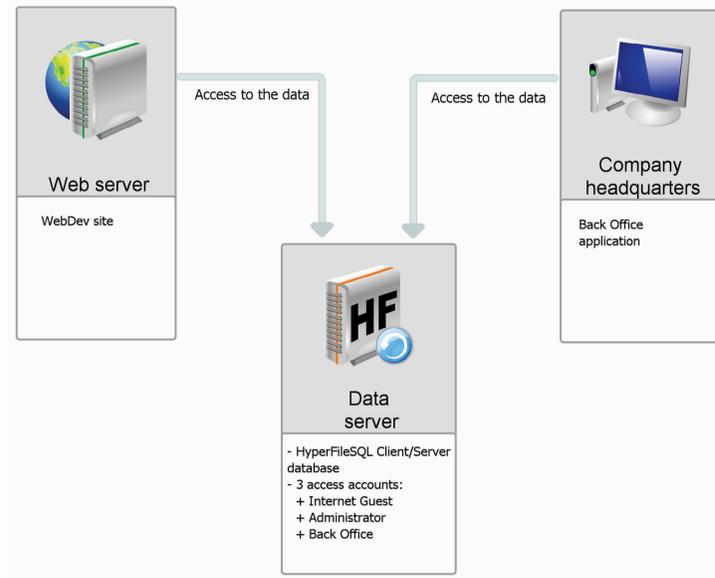
A business site allows the Web users to place some orders, to make some reservations, ... Several methods can be used to retrieve this information.

**Tip:** Process the data retrieved in a "Back Office" application developed in WinDev.

## 1 Sharing data

The data of a WebDev site can be directly shared with the data of the Back Office application used to process the operations performed on the site. All you have to do is use a HyperFileSQL Client/Server database.

The principle is as follows:



**Benefit:** the site's data can be accessed in real time.

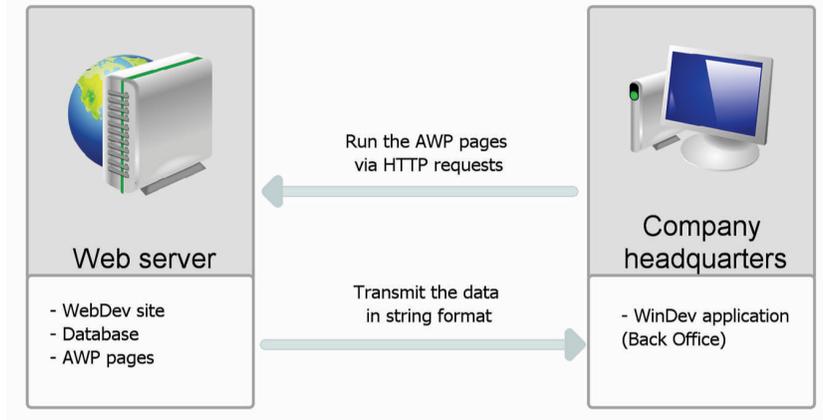
**Drawback:** You need to manage security in an optimized way; use a secure connection to access the data, open a specific port to access the data (port 4900 by default), configure the firewall, ...

*This solution is recommended for a site installed on a dedicated server.*

## 2 Retrieving the data from a site

### 2.1 Via an interrogation AWP page

WebDev enables you to create AWP pages. The AWP pages are independent pages that can run processes from the parameters passed in command line to the page. The AWP pages can be used to retrieve the data entered on the site.



The AWP pages found on the Web server are interrogated on a regular basis by the HTTP requests run by the Back Office application. The data is returned in string format and processed by the WinDev application.

**Benefits:**

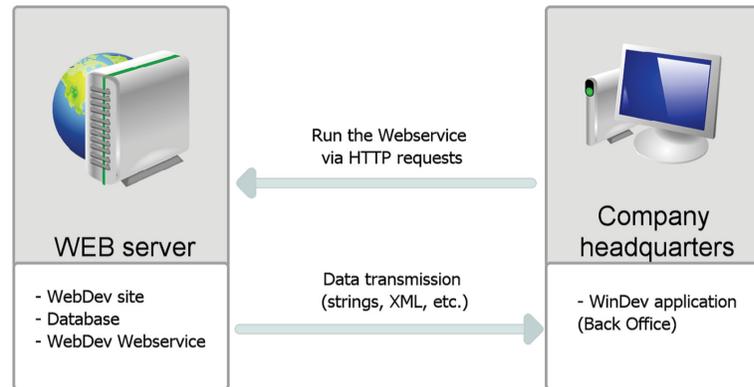
- Solution that can be easily implemented.
- Equivalent to a custom Web service.
- No additional executable is required on the server. No additional session is run on the server.

**Drawback:** One-way data retrieval: no data update toward the server.

*This solution can be used on a shared server.*

### 2.2 Via a Webservice

WebDev enables you to create Webservices. The Webservice is installed on the Web server and the WebDev application server allows you to use it. The Webservice can be used to retrieve the data entered in the site.



The Webservice found on the Web server is interrogated on a regular basis by the HTTP requests run by the Back Office application. The data is returned in XML format, as strings (or other) and processed by the WinDev application.

**Benefits:**

- Solution that can be easily implemented.
- No additional executable is required on the server. No additional session is run on the server.

**Drawback:** One-way data retrieval: no data update toward the server.

*This solution can be used on a shared server.*

### 2.3 By FTP:

WebDev allows you to regularly save the data files of the WebDev site. Then, this backup can be transferred by FTP to the company headquarters. Then, all you have to do is:

- retrieve the compressed backup of the data files
- decompress the files (via a WinDev application)
- process the data found in the files.

**Benefits:**

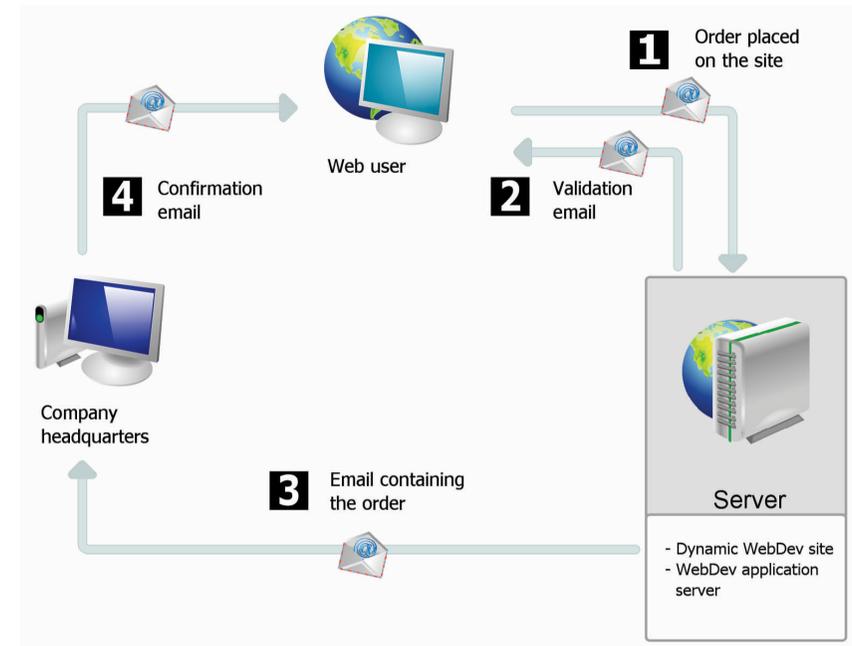
- Solution that can be easily implemented.
- No additional executable is required at the hosting company.

**Drawback:** All the data files of the application are retrieved.

*This solution can be used on a shared server.*

### 2.4 By emails:

The principle:



1. The Web user places an order on the site.
2. The WebDev site informs the Web user that his order was successfully saved. The Web user is informed that a confirmation email will be sent to him.
3. The WebDev site sends an email to the company headquarters. This email contains the order placed by the Web user.
4. An email is sent by the company to the Web user to inform him that his order will be processed.

**Benefits:**

- Solution that can be easily implemented.
- No additional executable is required on the server.

**Drawback:** One-way data retrieval: no data update toward the server.

*This solution can be used on a shared server.*

### 3 Performing a replication

The replication is used to keep the remote databases of identical structure updated. A replication can be performed between the database of a WebDev site and the database of the WinDev application at the company headquarters.

For example, a database used to enter orders and to manage stocks is used both on a WebDev site (online orders) and at the company headquarters (orders by phone or by mail).

The replication is used to:

- transmit the online orders to the company headquarters (one-way replication)
- transmit the online orders to the company headquarters and transmit the stock update to the Internet site (two-way replication)

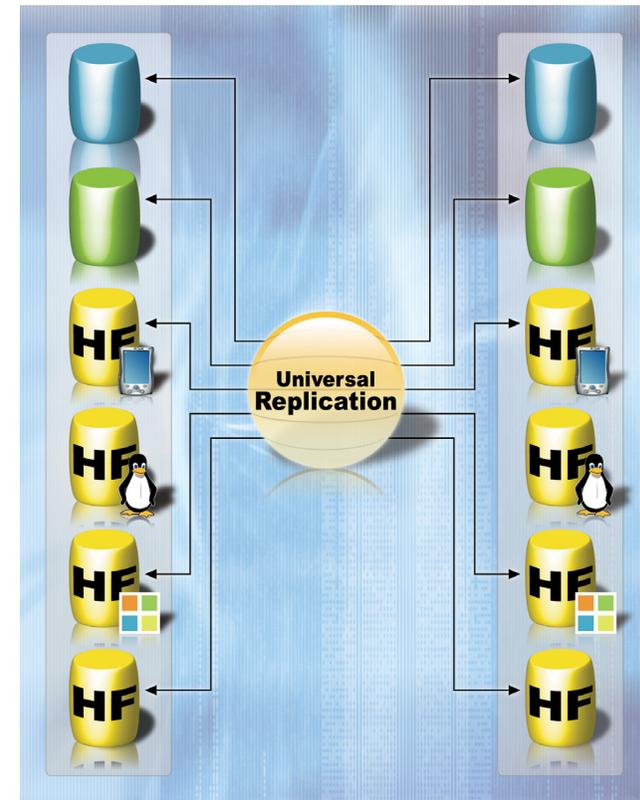
**Caution:** The replication constraints must be taken into account during the project creation. Indeed, several rules must be followed when describing the analysis and the file items.

The replication can be performed by email or via the assisted universal replication. See the online help for more details.

# Universal replication

The universal replication is used to update databases of same format or databases of different formats (HyperFileSQL Classic, Oracle, SQL Server, ...). You can for example perform a synchronization between a HyperFileSQL database and an Oracle database.

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 synchronization can be adapted to special cases. For example, you have the ability to retrieve the records concerning a specified product or the records created on a specific date, manage the conflicts, ...

These adaptations must be done by programming by *HRplFilterProcedure*

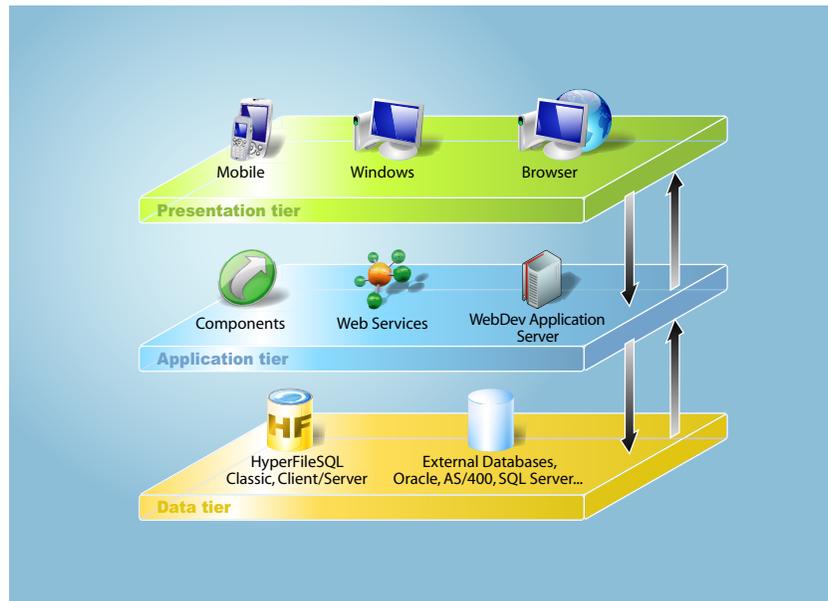
# 3-tier architecture

The purpose of 3-tier programming is to separate the "3 standard layers" of an application: UI, process and data.

Therefore, an application will include 3 separate tiers:

- a presentation layer,
- an application layer,
- a data access layer.

The reason for separating the layers is to facilitate the maintenance and the future evolutions of the application. 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.



## PART 6

# Running the test of a Web site



# Running the test of a site: The elements to test

Running the test of a site is a main step when developing a site. The test is used to check the operating mode and the usability of the site.

The main points to check in a WebDev site are as follows:

- **the operating mode of the site:** sequence of pages, execution of the code, ...
- **the style of the site:** use different browsers (mainly Internet Explorer and Netscape Navigator), use different screen resolutions, resize the browser, ...
- **the specific Web features:** print, cookies, browser "Back" key, ...
- **the access to the site by different Web users:** management of logins and passwords, management of concurrent accesses to the data files, ...
- **the non-regression of a site during an update:** validating the modifications performed in a site and checking whether the unmodified features are still accessible.
- **the stress test** (for a site installed on the server): validating the maximum number of Web users who can access the site.

# How do I run the test of a site?

Several tools are available for running the test of a WebDev site:

- the main editor of WebDev.
- the WebDev administrator. This module is required to run a dynamic WebDev site. On the development computer, the test of the dynamic sites can be run from a test page created by the administrator.
- WDTTestSite, tool for running stress tests.

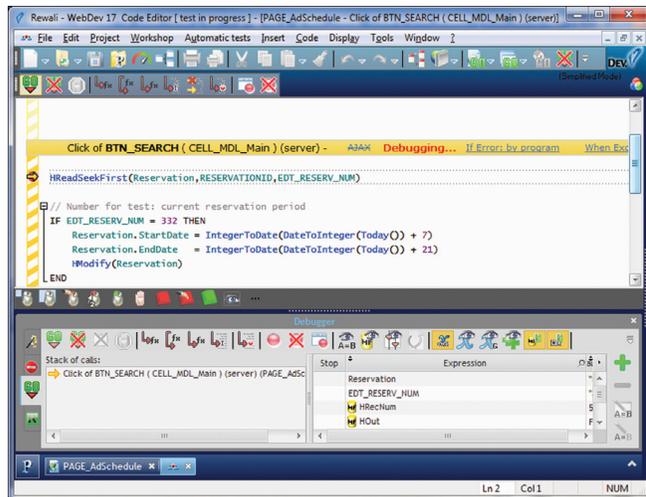
The table below presents how and when these test modes must be used.

Test mode	Type of test	Run the test
Editor	<p>Running the test from the editor allows you to test:</p> <ul style="list-style-type: none"> <li>• the features of the site,</li> <li>• the code used in the different processes. A powerful debugger allows you to monitor the execution of the different processes.</li> <li>• the use of the site with different browsers (installed on the development computer). This type of test allows you to use the debugger.</li> </ul>	<p>"Project .. Test mode .. Trace the project" or "Project .. Test mode .. Debug the project from the home page" from the editor.</p> <p>Note: The rights granted to the Web user (access rights, write rights and read rights) correspond to the rights granted to the developer.</p>
WebDev administrator  (dynamic WebDev site only)	<p>Running the test from the administrator allows you to test:</p> <ul style="list-style-type: none"> <li>• the features of the site,</li> <li>• the specific Web features (cookies, ...),</li> <li>• the use of the site in real conditions.</li> </ul>	<p>Start the WebDev administrator ("Start" menu) and click the "Test page" button ("Advanced" tab). This test allows you to use the WebDev site in conditions similar to the ones of a Web user.</p> <p>The rights granted to the Web user (access, write and read) correspond to the rights granted to the default Web user defined in the Web server used.</p>

WDTTestSite (dynamic WebDev site only)	<p>WDTTestSite is used to run stress tests.</p> <p>WDTTestSite is used to start several simultaneous connections to a WebDev site. Each connection performs a set of actions in the WebDev site (preset scenario)</p> <p>This test must be run on a WebDev site installed on a deployment computer that is ready to operate.</p>	<ol style="list-style-type: none"> <li>1. Create a test scenario (with WDTTestSite).</li> <li>2. Install WDTTestSite and the scenario on the different computers to simulate an important number of accesses.</li> <li>3. Specify the number of connections to establish.</li> <li>4. Run the test scenario on each computer.</li> </ol> <p>Note: Each computer establishes a defined number of connections. Each connection performs the selected scenario.</p>
Automatic tests	<p>The automatic tests are used to run the test of the procedures and classes found in a site at different levels of development. The test scenarios are automatically generated in WLanguage from the procedure test or from the class test. The scenarios can be modified in the code editor.</p>	<ol style="list-style-type: none"> <li>1. Display the procedure or the class in the code editor.</li> <li>2. Select "Automatic tests .. Save a new test".</li> </ol>

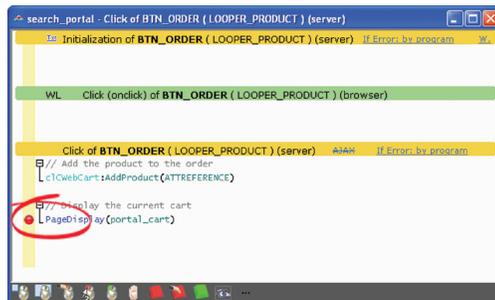
# The debugger

To help you optimize your code, a debugger is supplied with WebDev. The debugger allows you to run your dynamic sites step by step, by viewing the WLanguage code run, the content of the variables, ...



To start the debugger, use:

- a *breakpoint directly inserted into the code editor*: the debugger will automatically start when the line preceded by a breakpoint is run. To insert a breakpoint, all you have to do is click in front of the code line: a red bullet appears.



- the *WLanguage server keyword STOP directly in your code*: the debugger will be automatically started when this code line is run.
- the *"Project .. Test Mode .. Trace Project"*: the debugger is started at the same time as the dynamic site's test.

# When can I use the debugger?

The debugger can always be used when a test is run from the editor. To do so, breakpoints must be added to the requested locations in order to be able to debug the site step by step.

The different types of tests available from the editor are as follows:

- **"Go" on a page:**  
The test of the current page in the editor can be run via "Code .. Run the page test".
- **"Go" on the project:**  
The test of the project on the development computer can be run via "Project .. Test mode .. Trace the project".
- **The deployed project:**  
You have the ability to debug a deployed project. This feature is very useful to reproduce a problem that occurs in specific conditions. To start the deployed project, select "Project .. Test mode .. Connect to a site and debug the connection".  
**Note:** This option can be used with a development server or with a deployment server. The server must be configured to allow the remote debugging.
- **The deployed project being used:** The deployed project can be debugged while it is being used. This feature is very useful to reproduce a problem that occurs in specific conditions. Most common case: a problem occurs while you are using the site: to start the debugger, all you have to do is add a breakpoint into the code of your project. To use a session of the deployed project, select "Project .. Test mode .. Debug an existing connection".  
**Note:** This option can be used with a development server or with a deployment server. The server must be configured to allow the remote debugging.

# Test of a site in practice

## 1 Overview

WebDev proposes several methods for running the test of your sites:

- test of the entire project,
- test of a single page,
- test of a single query (see the "Reports and Queries" guide for more details),
- test of a single report (see the "Reports and Queries" guide for more details),
- running the project step by step,
- test of the performance of your site,
- regression test and automatic test,
- stress test.

**The test of the entire project** is used to simulate the startup of the site. This enables you to run the test of the entire site, even if its development is not finished yet. As soon as a problem occurs, you have the ability to start the debugger to find out and correct the problem.

**The test of a single page** is used to run the current page. This enables you to run the test of your project from a given page or to check the operating

mode of a page as soon as it is developed. Like for the project test, the debugger can be started as soon as a problem occurs.

**The test of a single query** is used to run the current query. This enables you to check the operating mode of a query as soon as it is developed.

**The test of a single report** is used to run the current report. This enables you to check the operating mode of a report as soon as it is developed. Like for the project test, the debugger can be started as soon as a problem occurs.

**Running the project step by step** is used to start the debugger when the site is started. This solution allows you to closely monitor how the site runs.

**The performance test** of your site is used to check and optimize the execution time of your site.

**The regression test (or automatic test)** is based on the execution of scripts. It enables you to check that during the execution of your site, ... the existing features are still supported.

**The stress test** is used to start several simultaneous connections to the same dynamic WebDev site.

## 2 Running the test of a WebDev project

### 2.1 Running the test of the project from the editor

Running the test from the editor allows you to test:

- the features of the site,
- the use of the site with different browsers.

The project test can be run regardless of the current element in the editor.

#### Different types of test

**To run the test of a static site from the editor**, select "Project .. Test mode .. Debug the project from the home page" (or ).

The editor is automatically minimized, the browser specified in the WebDev options is opened ("Tools .. Options .. General options of WebDev .. Web") and

the home page of the site is displayed.

You can also connect to a site to debug the connection or to debug the current connection to a site.

**To run the test of a dynamic site from the editor**, select "Project .. Test mode .. Debug the project" (or  or [Ctrl] + [F9]).

The editor is automatically minimized, the browser specified in the WebDev options is opened ("Tools .. Options .. General options of WebDev .. Web") and the first page of the site is displayed.

**To run the test of a static + dynamic site from the editor:**

- **to run the test of the static section of the site:** perform the operations corresponding to the test of a static site.

- **to run the test of the dynamic section of the site:** perform the operations corresponding to the test of a dynamic site.

**Note:** To run the test of a site with a specific browser, select "Project .. Test browser" and select the browser you want. The default browser is defined in the options of WebDev ("Tools .. Options .. General options of WebDev" "Web" tab).

#### Dynamic site: Starting

The following modules are automatically started when running the test of a dynamic WebDev site:

- The Web server installed on the computer and configured for WebDev when installing WebDev. The test cannot be run if the Web server is not started.
- The WebDev administrator (WD170ADMIN.EXE). The administrator is used to manage the connections to the Web server and to configure the WebDev sites. **Note:** the test of a project can be run from the administrator's test page ("Advanced" tab of WD170ADMIN, "Test page" button).
- The WebDev engine (WD170AWP.EXE). The WebDev engine is used to manage the requests made by the Web users from their browser and to return the corresponding dynamic HTML page. **Note:** The WebDev engine is started only if the project contains dynamic pages.
- The Internet browser. The Internet browser is used to display the HTML pages of the WebDev site.

### 2.2 Running the test of the project from the WebDev administrator

Running the test from the WebDev administrator (WD170Admin) is used to test:

- the features of the site.
- the specific Web features (use of cookies, ...).

**Note:** The WebDev administrator can only be used to run the test of the dynamic sites or the test of the dynamic section of the static + dynamic sites.

Running the test from the WebDev administrator is equivalent to starting the site from a remote computer.

Before deploying a WebDev site, we recommend that you run the test of this site at least once from the WebDev administrator.

#### To run the test from the WebDev administrator:

1. Start the WebDev administrator ("Tools .. WDAAdmin - WebDev administrator").
2. Click the "Test page" button ("Advanced" tab).

**To stop the test**, display the WebDev administrator (click the icon 17 found in the taskbar) and click the "Disconnect" button ("Connections" tab).

**Note:** The WebDev administrator also allows you to run a test of the project that is equivalent to the test of the project from the editor:

1. Start the WebDev administrator ("Tools .. WDAAdmin - WebDev administrator").
2. Click the "Run the test" button ("Connections" tab).

### 2.3 Stress test/Regression test

WDTTestSite is used to perform stress tests: WDTTestSite is used to establish several simultaneous connections to a dynamic WebDev site.

Each connection performs a set of actions in the WebDev site (preset scenario).

See the online help for more details.

### 3 Running the test of a remote WebDev site

WebDev proposed several methods to run the test of a site on the development computer and to debug it. However in some cases, you may have to debug the site directly on the end-user computers. From your office in London, you have the ability to debug a site running on a Web server in Taiwan. The debug operation is done without having to go anywhere, on the final configuration directly.

Two features are available:

- Starting and debugging the site on a remote application server.
- Debugging a site currently used on a remote application server.

For these two features, a specific configuration is required for the remote computer.

### 4 Running the test of a page

#### 4.1 Running the test of the page from the editor

To run the test of a page from the editor:

1. Open the page whose test must be run.
2. Select "Code .. Test page" ( , or [F9]).
3. The editor is automatically minimized and the page is run.

During the test, all the features of the page will be run. You will have the ability to open other pages for example.

#### 4.2 Stopping the test of a page

Several methods can be used to stop the test:

- **1st method:** close the page being tested. WebDev displays the editor that was used at the beginning of the test.
- **2nd method:**
  - Go back to the editor with the taskbar or with [Alt]+[Tab].
  - Confirm that the test must be stopped. WebDev displays the editor that was used at the beginning of the test.

### 5 Tracing a project

#### 5.1 Principles for debugging

Debugging an application consists in:

- checking the operating mode of a process,
- understanding the operating mode of an existing program,
- checking the value of the variables,
- checking the operating mode of special cases in an application or in a site.

The debugger is used to perform these operations.

**Note:** WebDev also includes several trace tools (trace window, information box, ...). See "Debugging without the debugger", page 205 for more details.

#### 5.2 Overview of the debugger

The debugger is used to trace the WLanguage programs in order to help you improve these programs. The source code run is viewed on the screen. The processes run are sorted in hierarchical order in the "Debugger" pane.

The value of the variables can be viewed:

- individually in the rollover tooltip of each variable.
- in the "Debugger" pane.

### 5.3 Features of the debugger

The debugger enables you to:

- find out the call stack
- view the content of the variables or expressions
- run the code step by step with ability to skip blocks.
- use conditional breakpoints
- modify the code while continuing the execution,
- ...

### 5.4 Debugging without the debugger

In some cases, running a program with or without the debugger may be different.

Indeed, the debugger introduces pauses in the execution of the process during which several tasks are performed by WinDev.

Therefore, the debugger cannot be used in a procedure called by a timer or in the code of a scrollbar.

**Note:** To find out the limits of the debugger, see the online help.

To debug these applications, you may want to follow the evolution of a value, how different procedures are called, ...

This information can be:

- displayed on the screen
- stored in a trace file.

**Caution:** If the information is displayed on the screen, it must only be displayed during the tests of the application.

#### Displaying the information

Two tools are used to display the information:

- **the information boxes:** *Info* function of WLanguage. **Caution:** The display of an information box is a locking operation.
- **the trace window:** WLanguage function *named Trace*. The trace window is displayed in the top left corner of the screen, without interrupting the execution of the program.

#### Managing the display of the debug information

Displaying the debug information on the screen is useful in test mode only.

Any unsuitable display must be deleted before distributing an application.

To avoid any oversight, we advise you to manage the display of the debug information via a global procedure.

For example:

```
PROCEDURE MyTrace (StringToTrace)
IF InTestMode () THEN
    Trace (StringToTrace)
END
```

In this code, depending on the result of *InTestMode*, the trace window is only displayed during the test of the application.

Such procedure enables you to leave the call to the trace windows in the code of the application without any risk of displaying it on the end-user computers.

The call to the trace procedure is similar to the use of the *Trace* function

```
MyTrace ("Customer: "+...
Customer.CustomerNum)
```

#### Creating a trace file

For the routines that take a long time to run (batch processes, ...), to check the operating mode of the program, you must keep a physical trace of the processes run (via a text file for example).

The following procedure is used to manage the display of the trace:

- On the screen (/DEBUG parameter in command line).
- in a text file (default mode).

```
PROCEDURE MyTrace (StringToTrace)
FilePath is int
FilePath = ...
fDataDirUser ()+...
    ProjectInfo (piProjectName)+".txt"
FileNum is int
DebugMode is boolean = False
IF Position (CommandLine (), ...
"/DEBUG") > 0 THEN
    DebugMode = True
END
IF DebugMode THEN
    Trace (StringToTrace)
ELSE
    FileNum = fOpen (...
    FilePath, ...
    foCreateIfNotExist+...
    foWrite + foAdd)
```

```

If FileNum <> -1 THEN
  DateTimeTrace is DateTime
  DateTimeTrace = SysDateTime()
  DateTrace is Date
  DateTrace = MyDate..Date
  TimeTrace is Time
  TimeTrace = MyDate..Time
  fWriteLine (FileNum, ...
  DateToString (DateTrace)+...
  " - "+TimeToString (TimeTrace))
  fWriteLine (FileNum, ...
  StringToTrace)
  fWriteLine (FileNum, " ")
  fClose (FileNum)
END
END
    
```

## 6 Performance test

### 6.1 Overview

The performance profiler enables you to check and optimize the execution time of your site.

Its principle is straightforward:

You run the test of your site. During this test, the performance profiler keeps track of all the actions performed and the corresponding processes run.

At the end of the test, the performance profiler displays:

- the 10 most time consuming operations.
- all the actions performed in the site whose test was run, sorted by duration (from the longest one to the shortest one).

You can select a process in order to analyze the reasons for its processing time in order to optimize it.

### 6.2 Starting the performance profiler

To start the performance profiler, select "Project .. Performance profiler .. Analyze the performance".

The project is automatically run in test mode. The process to optimize can be run in your site.

To go back to the editor, all you have to do is close your application or your site.

Then, the performance profiler displays the result of the analysis.

**Note:** We recommend you use the profiler to optimize your application or site (before deployment for instance).

#### Notes:

- The trace file is created by default in the data directory of the user. This file is named like the project. This file contains the information to trace during the execution of the program. The information is completed by the date and time of each "Trace". This enables you to detect a potential problem during the process.

- Sample content of the trace file:

```

01/12/2001 - 10:53:25:20
Customer name: Martin
    
```

### 6.3 Reading the result of the performance profiler

The performance profiler presents the result of the analysis in two tabs:

- the "Synthesis" tab presents the 20 longest processes.
- the "Details" tab presents all the processes run during the test of the application (from the slowest one to the fastest one).

The following information is displayed for each process:

Function	Function, process or procedure run.
Total Time	Execution time of the function.
Nb of calls	Number of calls made to the function (procedure or process).
Time 1 call	Execution time of a call to the function (procedure or process).
% code	Percentage of code run outside the call to a WLanguage function or outside the call to a custom function or procedure.
Parent	Process that called the function.

#### Note:

- The "Full execution" caption represents the total amount of time for running the test of the site with the performance profiler.
- The "Total Page XXX" caption represents the total amount of time for running the page XXX (from its opening to its closing).

### 6.4 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.

**Note:** If the process corresponds to a WLanguage function, it is fairly hard to optimize it.

## 7 Regression tests

### 7.1 Overview

Several test tools are available to guarantee the quality of your applications:

- The test mode (Go from a project or from a page) is used to immediately check a modification performed in your site.
- WDTTestSite that is used to create different tests for a WebDev site.

To automate these tests and to increase the quality of your applications, you have the ability to run **automatic unit tests**. These tests are used to easily check all the features proposed by your applications.

### 7.2 Automatic tests

Each test contains a scenario that can be directly edited in the interface of the product. This scenario is written in WLanguage and it can be modified at any time.

These tests can be run before each deployment in order to check the operating mode of a site after several modifications.

The following elements can be checked:

- the sets of procedures
- the classes

Each test is associated with a WLanguage code: the test scenario. This scenario can be viewed in the code editor. The code of the tests can be modified.

The tests and the associated code are not distributed to the end users. The number of tests for a site has no incidence on the size of the site supplied to the end users.

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

### 7.3 WDTTestSite

WDTTestSite is used to run different tests on a WebDev site.

The following tests can be performed by WDTTestSite:

- **Stress test:**  
The stress test consists in simulating the connection of several Web users to a WebDev site. Each Web user runs a set of operations (scenario) simultaneously.
- **Non-regression test:**  
The non-regression test consists in checking the operating mode of a WebDev site between two updates. The non-regression test consists in checking whether a scenario performed with an earlier version of the site still operates properly once the site was updated.
- **Test of a site in multi-user mode:**  
The test of a site in multi-user mode is used to check whether the concurrent accesses to the data files are managed properly. This test consists in simulating the simultaneous connection of several Web users to a WebDev site. Each Web user runs a set of operations (scenario) simultaneously.
- **Comparing different servers:**  
WDTTestSite is used compare the speed of different servers. To do so, run the same scenario on different servers and compare the execution time of this scenario.
- **Optimization of processes built using WLanguage:**  
WDTTestSite is used to compare the execution time of a scenario before and after the WLanguage code was optimized.

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



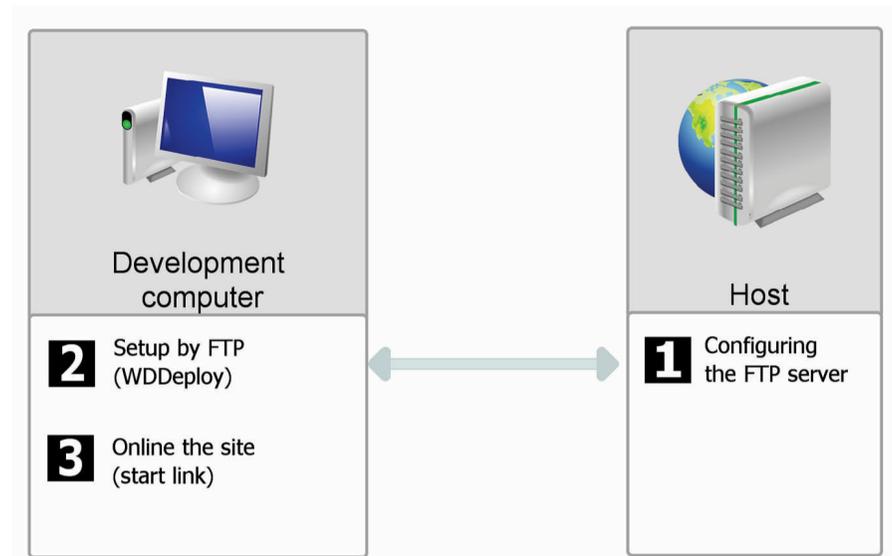
**PART 7**

**Deploying a Web site**

# Deploying a static site

## Principle

A static site is deployed via FTP:



## Information required for the deployment

No setup can be run without this information. Contact your hosting company to get this information.

**Before deploying via FTP**, the host must give the developer the following information:

- Name of the Web server where the setup must be performed (or its IP address).
- User name and password defined in the FTP server.

**When deploying via FTP**, the developer must specify in WDDeploy:

- the information given by the hosting company.
- the files to install.

# Deploying a static site, a semi-dynamic site or a PHP site in practice

## 1 Overview

When developing a static site, a semi-dynamic site or a PHP site, one of the important phases consists in deploying this site on a Web server (at a hosting company or in Intranet).

- **network deployment:** recommended when the server is easily accessible.

WDDeploy is used to simplify the deployment of your static sites.

Two deployment methods are available:

- **remote deployment (via FTP):** recommended when the server is not directly available.

**Note:** To deploy a dynamic site, see "Deploying a dynamic site", page 218.

## 2 Creating the setup

To deploy your static site:

1. Select "Workshop .. Deploy the site .. Deploy the site". WDDeploy starts.

2. Create the profile corresponding to your site. This profile contains the following information:

- Location of the local files ("**<ProjectName>\_WEB**" sub-directory of your site).
- Address of the WebDev site.
- Location of the files of the deployed site (network or FTP server).

- Characteristics of the FTP server used to update the site. These characteristics are provided by the hosting company, ...

3. Click the "Prepare" button. WDDeploy prepares the list of files to install. For an update, WDDeploy compares the files found on the development computer and the files already installed.

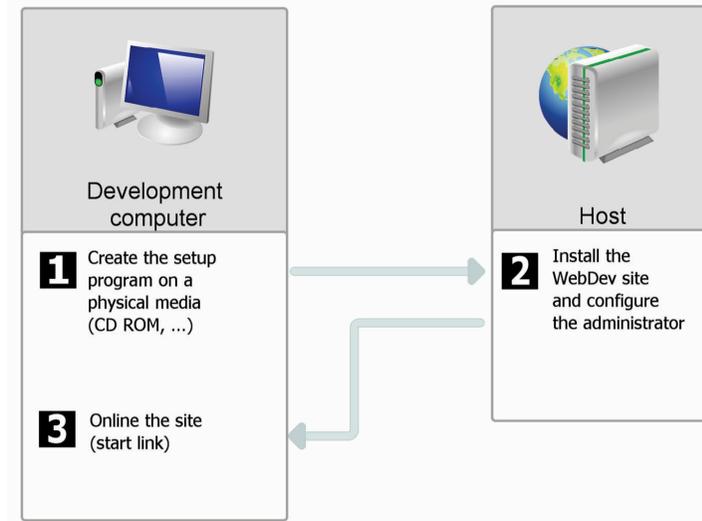
4. Click the "Synchronize" button. The files of your site are copied to the location specified in the profile. See the online help for more details.

# Deploying a dynamic WebDev site

## The different types of deployment

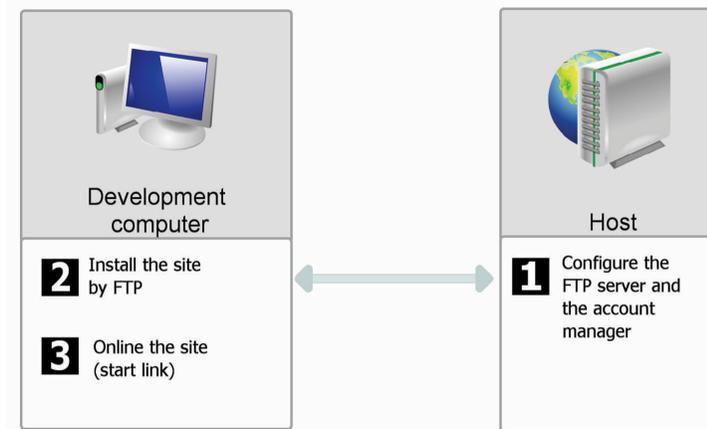
Three methods are available for deploying a dynamic site:

- **deployment by physical media (CD, ...)** of the WebDev site, with creation of a setup version deliverable on CD to the host.

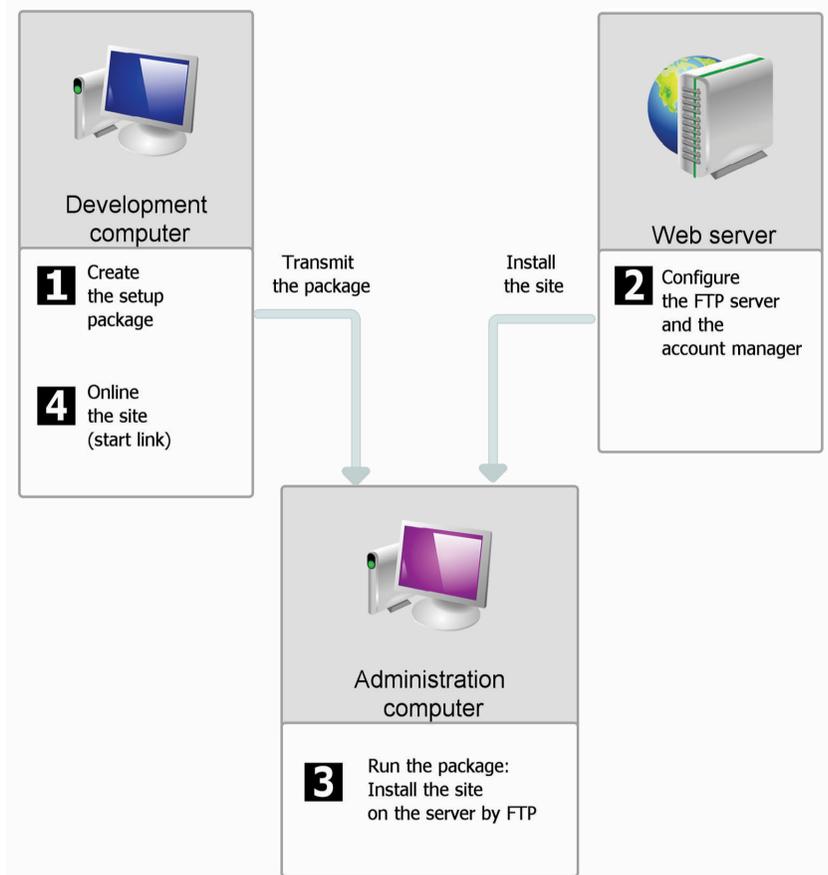


**Note:** a WebDev application server can be included in the setup supplied by physical media.

- **remote deployment from the development computer (file transfer by internet, FTP)**



- *remote deployment from an administration computer (file transfer by internet, FTP)*



## Choosing a deployment method

The choice of the deployment method mainly depends on the constraints imposed by the hosting company and on the location of the deployment server (remote or local).

**The setup via physical media is recommended** if the Web server can be easily accessed (the Web server and the development team are located in the same premises for instance).

**The remote setup (via FTP) from the development computer is recommended** if the deployment computer cannot be easily accessed. A standard FTP server must be found on the deployment computer.

Before performing a remote setup, the hosting company must configure the FTP server and the WebDev account manager.

**The remote setup (via FTP) from an administration computer** is recommended if the characteristics of the server are not known by the developer when creating the setup program. In this case, the setup program is called a "Package". If a site is intended for several clients, you have the ability to create a single package and to distribute it to all the relevant clients. The parameters of the server (address, FTP account, ...) are specified when the package is run on the administrator computer.

## Deployment by physical media: necessary information

When deploying by physical media, the information regarding the setup and the site's settings is supplied both in the setup program creation wizard and in the installation wizard.

**When preparing the setup**, you must provide:

- the setup directory,
- the files to install,
- the directory for generating the setup.

**During the setup**, you must provide:

- the directory of the data files,
- the directory of the site files.

**After the setup**, the characteristics of the site (connection time-out, number of connections, ...) must be configured in the WebDev administrator (deployment version).

## Deployment by FTP from the development computer: necessary information

No setup can be run without this information. Ask the provider to get this information.

**Before deploying via FTP directly from the development machine**, the host must give the developer the following information:

- Name of the Web server where the setup must be performed (or its IP address)
- User name and associated password defined in the WebDev account manager
- User name and password defined in the FTP server

**When deploying via FTP**, the developer must specify in the wizard:

- the information given by the hosting company
- the files to install
- the name of sub-directory for the data files
- the different parameters for connecting to the site

**Notes:**

- A delayed setup can be performed at the specified date and time.
- The setup and the update by FTP are secured by password and data encryption.

**Deployment by FTP from an administration computer: necessary information**

When creating the setup program, no specific information is required.

When running the site's setup program (from an administrator computer), the hosting company will have to specify:

- Name of the Web server where the setup must be performed (or its IP address)
- User name and associated password defined in the WebDev account manager
- User name and password defined in the FTP server

**Notes:**

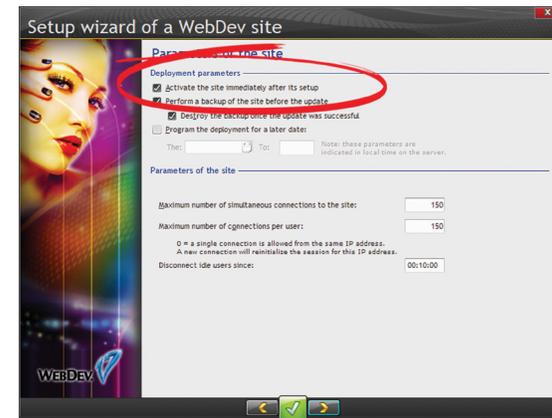
- A delayed setup can be performed at the specified date and time.
- The setup and the update by FTP are secured by password and data encryption.

# On-lining a dynamic WebDev site

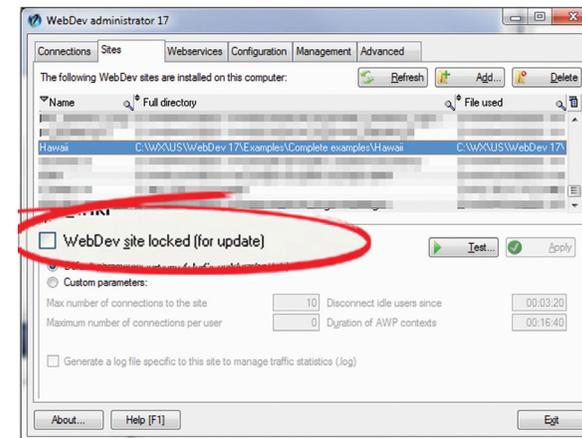
Your dynamic WebDev site can be accessed as soon as it is on-lined. On-lining a dynamic WebDev site consists in activating the site.

Several methods are available for activating a dynamic WebDev site:

- during the setup by FTP: a check box is used to automatically activate the site after its setup.



- with the remote administrator.
- with the WebDev administrator found on the deployment computer ("WebDev site locked" in the "Sites" tab).



## Deploying a dynamic site in practice

When developing a dynamic site, one of the important phases consists in deploying this site on a Web server (at a hosting company or on an Intranet).

Two deployment methods are available:

- **remote deployment (via FTP):** recommended when the server is not directly available.
- **deployment by physical media (CD, ...):** recommended when the server is easily accessible.
- **deployment by package:** recommended when the server is not easily accessible and if you do not know the characteristics of the server.

### 1 Implementing a library

Before performing a setup, all the objects found in your dynamic site must be included in a library. A library is a file that groups all the elements created during the development (description of the database, description of the pages, compiled source codes, ...). The HTML pages and the images are not included in the library. The library is automatically implemented when creating the setup procedure.

Once the library is created, the wizard for setup creation starts. You can choose to perform:

- **a setup via FTP (remote setup)** Before performing a remote setup, the host must have:
  1. Created and configured your FTP account on the server.
  2. Created and configured your WebDev account on the server. The host should provide you with a login and password for these accounts.
- **a deployment in PC SOFT Cloud:** PC SOFT proposes a hosting platform in the format of Cloud Computing: you deploy without having to worry about the hardware constraints and you are invoiced according to the real use of your applications. A PC SOFT Cloud account is required to perform this type of setup. If necessary, the wizard proposes to create this account.
- **a setup by FTP via the test hosting service of PC SOFT:** PC SOFT proposes to host your test sites.

#### Notes:

- For each Web server that hosts WebDev sites, the hosting company must own a license for the WebDev application server.
- The creation of the setup is performed by "Workshop .. Deploy the site".

To deploy a static site, see "Deploying a static site, a semi-dynamic site or a PHP site", page 212.

The setup is easily performed, directly from the environment. Once your site is developed, you have the ability to run its test directly.

A PC SOFT developer account is required to perform this type of setup. If necessary, the wizard proposes to create this account.

- **a deployment package:** A deployment package is an executable containing all the elements required to deploy a WebDev site. This package can be run from any computer to install the Web site on the server by FTP. To create this type of setup, there is no need to know the characteristics of the FTP account or the characteristics of the WebDev account found on the Web server.
- **a setup via physical media:** The wizard enables you to create a setup program for your site. This setup program must be supplied to your hosting company and it will allow your WebDev site to be installed on the Web Server. To install your WebDev site, the host must:
  1. Install the WebDev application server on the Web server (if not already done).
  2. Install your WebDev site by running the executable program *Install* supplied with your setup. **Note:** You also have the ability to create a setup via stand-alone physical media, containing the setup of a WebDev application server limited to 10 connections. This solution is ideal for proof-of-concepts presented on a laptop.

## 2 Remote deployment (by FTP)

### 2.1 The steps

The deployment steps are as follows:

1. Install the WebDev application server at the hosting company (optional step). See the documentation about the WebDev application server for more details.
2. Creation by the hosting company of a WebDev account (in the administrator) and creation of an FTP account (in the FTP server).
3. The hosting company supplies the names and associated passwords as well as the name (IP address) of the server.
4. Create the remote setup from the editor ("Workshop .. Deploy the site"). The necessary files are copied into the transfer directory.
5. On-line the WebDev site.

### 2.2 Creating the setup

You must define in the wizard:

1. **The information regarding the remote computer.** This information must be supplied by the host:
  - **Address of the server** (name of a computer accessible by network, IP address or Internet address)
  - **Characteristics of the WebDev account** (user name and password).
  - **Characteristics of the FTP account** (user name and password).

To easily identify the characteristics of the server for a forthcoming update, give a name to this server. This name will be displayed (followed by the server address) in the window for choosing the type of setup.

2. **The files to install.** By default, the wizard selects the library, the data files, the images and the HTML pages.

3. **Configuring the setup:** Chose the type of remote setup to perform:

- **Instant update:** The site files are compressed and immediately transferred onto the Web server (the Web server must be accessible from the setup computer). Then, the WebDev site is automatically installed on the Web server.

- **Scheduled update:**

The site files are compressed and immediately transferred onto the Web server (the Web server must be accessible from the setup computer). The WebDev site will be installed on the server at the specified date and time. This enables you to update your site during the night (between midnight and 3 AM for instance), when the number of Web users is less important.

Caution: the specified date and time are the ones of the server. Make sure that they are valid.

- **Enable the site immediately after setup:**

The site will be available after its setup: you will be able to connect to it either by using the proper address, or via a static page containing a link pointing to the site. If this option is not checked, the site can be enabled via the remote administrator.

4. **Configuring the site.** These options are used to configure the options for connecting to the site after its setup.

These parameters can be modified if necessary:

- by the deployment administrator found on the server.
- by the remote administrator, that can be used by the site manager.

5. If an analysis update was performed, select the **automatic modification of the data files** during the setup.

**Note:** if this option is chosen, the update will be proposed during the setup.

**Special case:** If the data files found on the server are in HyperFile 5.5 format, they must be migrated to HyperFileSQL Classic format. This migration must be performed during the first update of the site.

6. Specify whether your site uses **the Native AS/400 access.**

Reminder: if you site uses the Native AS/400 Access, the AS/400 DLLs must be installed on the Web server. If you are using a license for a limited number of computers, the number of current connections must be managed in the WebDev site (INI file or registry, ...).

Summary of the elements installed and their location:

Site server	Data server
Directory of the WebDev sites for user N	Data directory for user N
Directory defined by the hosting company. In this directory, a sub-directory will be created for each site of the user N.	Directory defined by the hosting company. In this directory, a sub-directory will be created for the data files of each site of the user N.
The following elements are installed in this site sub-directory: HTML pages, the library (".WDL"), the image directory.	The following elements are installed in this data sub-directory: the ".FIC" files, the ".NDX" files, the ".MMO" files.
Example: The directory of the sites for user N is "C:\UserN\WebDev sites". The user N installs his site named "Boat". This site will be installed in the "C:\UserN\WebDev sites\Boat" directory.	Example: The data directory of user N is "D:\UserN\WebDev data". The user N installs his site named "Boat". By default, the data files will be installed in the "D:\UserN\WebDev data\Boat" directory.

### 3 Deployment by deployment package

#### 3.1 The steps

The deployment steps are as follows:

1. Create the deployment package from the editor ("Workshop .. Deploy the site"). A setup file is created.
2. Send the deployment package (via Internet, a CD, ...) to the administrator of the Web server (hosting company for example)
3. Install the WebDev application server at the hosting company (optional step). See the documentation about the WebDev application server for more details.
4. Creation by the hosting company of a WebDev account (in the administrator) and creation of an FTP account (in the FTP server).
5. Run the deployment package on a computer and installing the WebDev site.
6. On-line the WebDev site.

#### 3.2 Creating the setup

You must define in the wizard:

1. **The files to install.** By default, the wizard selects the library, the data files, the images and the HTML pages.
2. If an analysis update was performed, select the **automatic modification of the data files** during the setup.  
**Note:** if this option is chosen, the update will be proposed during the setup.  
**Special case:** If the data files found on the server are in HyperFile 5.5 format, they must be migrated to HyperFileSQL Classic format. This migration must be performed during the first update of the site.
3. **Configuring the site.** These options are used to configure the options for connecting to the site after its setup. These parameters can be modified if necessary:
  - by the deployment administrator found on the server.
  - by the remote administrator, that can be used by the site manager.

#### 4. Specify whether your site uses the Native AS/400 access.

Reminder: if you site uses the Native AS/400 Access, the AS/400 DLLs must be installed on the Web server.

If you are using a license for a limited number of computers, the number of current connections must be managed in the WebDev site (INI file or registry, ...).

#### 5. The default profile of the Web server to use.

These parameters can be entered or modified by the person who installs the deployment package.

#### 6. Configuring the setup:

Chose the type of remote setup to perform:

- **Scheduled update:**  
The site files are compressed and immediately transferred onto the Web server (the Web server must be accessible from the setup computer).  
The WebDev site will be installed on the server at the specified date and time. This enables you to

update your site during the night (between midnight and 3 AM for instance), when the number of Web users is less important.

Caution: the specified date and time are the ones of the server. Make sure that they are valid.

- **Enable the site immediately after setup:**  
The site will be available after its setup: you will be able to connect to it either by using the proper address, or via a static page containing a link pointing to the site. If this option is not checked, the site can be enabled via the remote administrator.

### 3.3 Installing a dynamic site

To install a site, you must:

1. Run the deployment package.
2. Specify the characteristics of the Web server on which the WebDev site must be installed (characteristics of the FTP account for the server and characteristics of the WebDev account for the server).
3. Validate the setup of the site.

## 4 Deployment by physical media

This deployment is performed in three steps:

1. Creating the setup version.
2. Installing (if necessary) the WebDev application server on the Web server. See the documentation about the WebDev application server for more details.
3. Installing the site at the hosting company.

### 4.1 Creating the setup

You must define in the wizard:

1. **A title and a version name** for your setup. A title can be entered in each language that can be selected during the setup.
2. **A default directory.**
3. **A layout** for the setup window. The layout can be customized with the "Customize" button.
4. **The files of your site.**  
**Note:** the library, HTML, Java, JavaScript and image files are selected by default.
5. **The appendix files**
  - **The license** (file in txt format) will be displayed when starting the setup.  
**Note:** if the license is not validated by the person performing the setup, the setup will be canceled.
  - The **"readme"** file: this file can correspond to different formats (txt, htm, doc, hlp, pdf ...).

Regardless of the selected format, the user must have a software allowing him to view this file.

- **An executable** that must be run at the end of setup.  
**Caution:** Don't forget to include the files linked to this executable (DLLs, parameter files, other executables, ...).  
**Note:** A license file and a "ReadMe" file can be selected for each setup language.

6. If an analysis update has been performed, select the **automatic modification of the data files** during the setup.  
**Note:** if this option is chosen, the update will be proposed during the setup.

7. If programs not created with WebDev must access the HyperFileSQL files of your site, choose to install an ODBC driver.  
**Note:** if this option is chosen, the setup of the driver will be proposed during the setup.

8. Specify whether your site uses the **Native AS/400 access**.  
Reminder: if you site uses the Native AS/400 Access, the AS/400 DLLs must be installed on the Web server.  
If you are using a license for a limited number of computers, the number of current connections must be managed in the WebDev site (INI file or

registry, ...).

#### 9. Print the **deployment documentation**:

The deployment documentation specifies the WebDev license as well as all the modules installed during the deployment. It also specifies the registry keys generated for the WebDev site. When the site is installed on the server, these keys can be updated in the registry or saved into a file for further examination.

**The deployment documentation must be provided to the person in charge of installing your WebDev site** (host or Intranet manager).

#### 10. Modify (if necessary) the **directory for generating** the setup.

By default, the setup is generated in the "INST" sub-directory of your project.

#### 11. Generate the setup program:

An information window signals the possible presence of files with **long names**. If the setup is performed on a network drive, make sure that:

- the network drive supports the long names,
- the network drive does not modify the case (uppercase/lowercase characters) in the names of the files.

#### 12. Once the setup is created, WebDev proposes to:

- run the test of the created setup.
- open the Windows explorer in the directory for generating the setup.

#### 13. Perform a backup of your site.

We recommend that you keep a backup of the source code corresponding to an installed version of your site.

### 4.2 Installing the dynamic site on a Web server (physical media)

If you are installing your site via a hosting company, this one can propose two possibilities:

- **Install your site on a "shared" server.** This server manages the sites for several customers. This server is configured when installing the WebDev application server. The WebDev administrator (WD170ADMIN.EXE) lets you individually configure each WebDev site installed on this server. Caution: The memory resources of the server are shared among all the sites.

- **Install your site on a dedicated server** (recommended solution). You own this server. In this case you must:

- configure the Web server,
- manage the memory resources for your site.

**To install the WebDev site on a Web server (at a hosting company or in Intranet):**

#### 1. Install (if necessary) the **WebDev application server** on the Web server.

Caution: if the WebDev application server is not installed on the Web server, the dynamic WebDev site will not work properly.

**Note:** For each Web server hosting WebDev sites, the host must own a license for the WebDev application server.

#### 2. To install the WebDev site, **INSTALL.EXE** (supplied with the setup of the WebDev site) must be run on the Web server.

**Important:** To update an existing site, you must lock the access to the site via the WebDev administrator (check "WebDev site locked" in the "Sites" tab).

#### 3. Choose the **setup language**: a setup wizard starts. Go to the next plane.

4. The setup automatically detects whether one or more of the following **Internet servers** are installed on the computer:

- **IIS (Microsoft Internet Information Server) Version 2 to 7:** Microsoft server.
- **Apache version 1.3.x and 2.x:** free Web server provided by Apache Group.

Three cases may occur:

- **Case 1: your Web server is not listed:** select "Other server". See the online help for more details.
- **Case 2: your Web server is listed in gray:** your Web server was not detected. To select it, all you have to do is check "Display all the servers". Then, you must ask to generate a configuration file in order to be able to run the configuration later (if you want to install your server later for example).
- **Case 3: your Web server can be directly selected:** select your server. You can ask to generate a configuration file in order to run the configuration later. The configuration file can be modified before it is run. Depending on the selected server, the extension of the configuration file will be ".reg", ".conf" or something else.

5. Select the setup directory: <webroot> specifies the root directory of the Web server. Depending on the selected Web server, the root directory is detected and displayed. If the root directory of the Web server is not displayed, it can be entered in the gray area or it can be found via the "... " button.

**Caution:** if the area is already filled, avoid modifying it.

6. **Update the data files** (optional step, for site update only). To update the data files associated with the WebDev site, choose to run the automatic modification of the data files. Contact the provider of the site if you don't know whether you must perform this update or not.

**Caution:** The data files must not be in use. Use the WebDev administrator to check whether no current connection is in progress and to stop the sites (check "Locked site" in the "Sites" tab).

7. Install an **ODBC driver for HyperFileSQL** if programs not developed with WebDev must access the HyperFileSQL files of the site that will be installed (optional step).

8. The summary of the setup is displayed:

- Go to the next screen to start the setup.
- To modify some of the parameters in the previous windows, click "Previous".

9. The setup is completed:

- Click the address to run the test of your site locally. This option can be used only if the Web server was configured automatically.
- Click "Copy" to retrieve the proposed address. See for more details "On-lining the dynamic site", page 224.
- Click "Done" to end the setup.

#### 10. **Configuring the Web server:**

Two cases may occur if the Web server was not configured automatically:

- Case 1: Your server was detected and a configuration file was generated: you can view, modify (if necessary) and run this file.
- Case 2: Your server was not detected: see the online help for more details.

11. Use the WebDev administrator to unlock the locked sites and to configure the parameters of your dynamic site.

These parameters correspond to:

- the total number of authorized connections to the site,
- the number of authorized connections for a user,
- the authorized idle time.

Run the test of the WebDev site from the WebDev administrator via the [Test page] option ("Advanced" tab).

#### Notes:

- The WebDev administrator (WD170Admin.EXE) must be started and run as background task. It can be installed as service in Windows NT.
- See "The WebDev administrator in practice", page 228 for more details.

#### Important: The resources required per connected Web user

For each connected Web user, you must plan for:

- about 400 KB of RAM per connection, in addition to the memory required by the server.
- about 1 MB of disk space (virtual memory) per connection, in addition to the space required by the site.

For example, for 20 parallel connections, you must plan for:

- RAM: 67.8 MB (7.8 MB of RAM for this site, 60 for the NT server).
- Virtual memory: 84 MB (20 MB for the connections and 64 MB for the NT server).

## 5 On-lining the dynamic site

To enable the Web users to access your site installed at a hosting company, create a link in your home page allowing your dynamic WebDev site to be started.

If the setup was performed by physical media, the address to use was supplied at the end of the setup. Replace <localhost> by the address of your site.

**Example used to start a "OURAPP" site from the PC SOFT site:**

```
http://www.windev.com/WD170AWP/  
WD170AWP.EXE/CONNECT/OURAPP
```

To start the site by using its address (mysite.com for instance), you must:

1. Create an "Index.html" home page.
2. In the header of this page, use the HTML commands (META REFRESH) to perform a redirection to the WebDev site, which means for example to the following address:

```
"http://195.51.231.57/WD170AWP/  
WD170AWP.EXE/CONNECT/  
prettyeyes"
```

**Example of META tag that must be included in the <HEAD> section of an HTML page:**

```
<meta http-equiv="refresh" content="0  
;URL=http://195.51.231.57/WD170AWP/  
WD170AWP.EXE.EXE/CONNECT/  
jaidebeauxyeux">
```

**Special case: If a home page was defined in your dynamic site, the address for starting this home page is as follows:**

```
http://<ServerAddress>/  
<ProjectName>_WEB/
```



## PART 8

# Hosting the WebDev sites

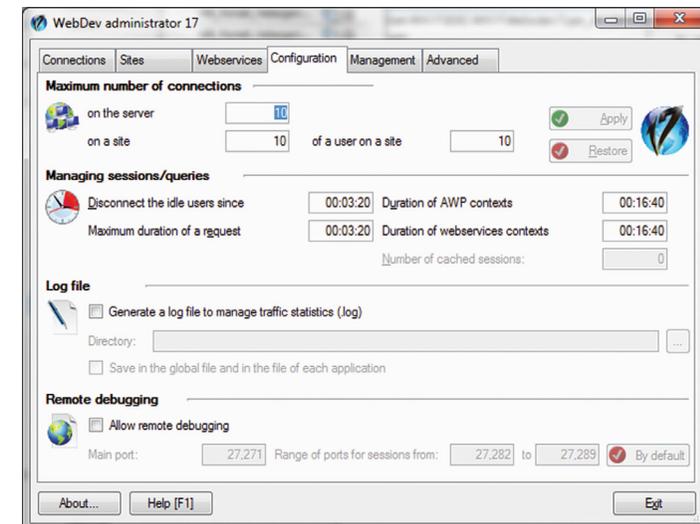


# What is the WebDev administrator intended for?

In development (for test) and in deployment, the WebDev administrator is required to run the dynamic WebDev sites.

When developing a dynamic WebDev site, the administrator is mainly used to:

- run the test of the dynamic sites via the test page ("Advanced" tab, "Test page" button)
- manage the webservices ("Webservices" tab)
- configure the connections to the dynamic sites for test: disconnection time-out, ... ("Configuration" tab)
- delete the current test connections ("Connections" tab)
- change the Web server used for your tests ("Advanced" tab, "Server" option)
- perform a diagnostic if a problem occurs when starting a dynamic WebDev site ("Advanced" tab, "Diagnostic" button)
- allow the remote debugging ("Configuration" tab).
- ...



In deployment, the WebDev administrator is used to:

- manage the WebDev accounts ("Management" tab)
- manage the configuration of the different dynamic sites installed on the server: number of authorized connections, ... ("Sites" tab)
- customize the error messages displayed in the different dynamic sites installed on the server ("Management" tab)
- ...

# The WebDev administrator in practice

## 1 Overview

The WebDev administrator is a module used to manage dynamic WebDev sites.

When developing a dynamic WebDev site, the administrator is mainly used to:

- run the test of the site via the test page ("Advanced" tab, "Test Page" option).
- define the parameters for connecting to the site for test: disconnection value, ... ("Configuration" tab).
- delete the current test connections ("Connections" tab).
- change the Web server used for your tests ("Advanced" tab, "Server" option).
- perform a diagnostic if a problem occurs when starting a WebDev site ("Advanced" tab, "Diagnostic" option).
- ...

In deployment, two types of WebDev administrators are available:

- **the WebDev Deployment administrator** that allows the hosting company to manage the dynamic WebDev sites installed on a Web server.

## 2 Interface of the WebDev administrator

The WebDev administrator is made of a window containing several tabs:

- "Connections" tab.
- "Sites" tab.
- "Webservices" tab.
- "Configuration" tab.
- "Management" tab.
- "Advanced" tab.

### 2.1 Characteristics of the current connections

The "Connections" tab gives various information about the current connections. This information can be consulted at any time on the server:

- the **number of current connections** indicates the

- **the remote WebDev administrator** that allows the site manager to manage his dynamic WebDev sites remotely from any computer.

This help only presents the WebDev Development administrator.

For more details about the deployment tools, see their help.

### Starting the WebDev administrator

To start the WebDev Development administrator, select:

- "Tools ..WDAdmin - WebDev administrator" from WebDev Development.
- "Programs .. WebDev 17 .. WebDev administrator" from the "Start" menu.

The WebDev administrator is automatically started during the first test of the site.

Then, the administrator can be run in background task.

By default, the WebDev administrator allows ten simultaneous connections to the same dynamic WebDev site (GO icon).

total number of connections to the dynamic WebDev sites managed by the administrator.

- The **Table of current connections** indicates for each connection:
  - the identifier of the connection,
  - the site affected by the connection,
  - the identity of the connected client. At run time, the client is identified by its IP address or by its Internet address,
  - the current time of the connection,
  - the idle time of the connection.
- **the refresh rate:** this parameter corresponds to the rate at which the table that displays the current connections is refreshed.

The "Connections" tab can also be used to:

- Run the test of the sites installed on the computer (in development version only): Select the site from the combo box and click "Test". This test mode is used to run the test of options specific to the management of cookies on the development computer.
- Stop an established connection: Select one of the connections and click "Disconnect".
- Stop all the current connections and close the administrator: click "Disconnect all".

### 2.2 Sites installed

The "Sites" tab gives information about the dynamic WebDev sites installed on the computer.

The "Sites" tab returns the list of dynamic WebDev sites installed on the computer and managed by the WebDev administrator. For each site, the following information is displayed in a table:

- the name of the site,
- the full setup directory of the site on the computer,
- the name of the corresponding "project" file.

The "Sites" tab can also be used to:

- **Delete** the selected site from the table. The site is deleted from the administrator: the WebDev site is still found on the disk.
- **Refresh** the list of sites found in the table. This refresh operation is used to check the presence of a site in the administrator once it has been installed.
- **Add** a site into the table.
- **Configure** each site found on the computer. The configuration used by default is the one defined in the "Configuration" tab. You also have the ability to create a specific configuration for the selected site. This configuration affects the number of authorized connections, the authorized idle time and whether the site must be locked during an update. See the "Configuration" tab for more details. To validate the new setting immediately, click the "Apply" button.

### 2.3 Webservices installed

The "Webservices" tab gives information about the dynamic WebDev Webservices installed on a computer.

The "Webservices" tab lists the dynamic WebDev webservices installed on the computer and managed by the WebDev administrator. For each webservice, the following information is displayed in a table:

- the name of the webservice,
- the full setup directory of the webservice on the computer,
- the name of the corresponding "project" file.

The "Webservices" tab can also be used to:

- **Delete** the selected webservice from the table. The webservice is deleted from the administrator tool: the WebDev webservice will still be found on the disk.
- **Refresh** the display of the webservices found in the table. This refresh operation is used to check the presence of a webservice in the administrator once it has been installed.
- **Add** a webservice into the table.
- **Configure** the webservices found on the computer individually. The configuration used by default is the one defined in the "Configuration" tab. You also have the ability to create a specific configuration for the selected site. This configuration affects the number of authorized connections and whether the webservice must be locked during an update. See the "Configuration" tab for more details. To validate the new setting immediately, click the "Apply" button.

### 2.4 General configuration

The general configuration corresponds to the default connection options for the WebDev sites found on the server and managed by the administrator.

The "Configuration" tab is used to:

- Configure the connections.
- Configure the sessions and the requests.
- Manage the log file to get some traffic statistics.
- Manage the remote debugging.

The connections can be configured using the following options:

- **Maximum number of connections on the server:** Maximum number of simultaneous connections allowed for all the sites managed by the administrator (a connection = a Web user).
- **Maximum number of connections on a site:** Maximum number of simultaneous connections allowed for each site managed by the administrator. This number can be modified for each site in the "Sites" tab.
- **Maximum number of connections for a site user:** Maximum number of simultaneous connections allowed for a given Web user to each site managed by the administrator. If this parameter is set to zero, a Web user trying to simultaneously connect to the same site will be automatically disconnected then reconnected. If this parameter is greater than zero, an error message will be automatically displayed after the x simultaneous connections. This number can be modified for each site ("Sites" tab).

The configuration of the sessions and requests is done via the following options:

- **Disconnect users that have been inactive for:** This is the maximum connection time allowed without any action performed by the Web user. When this time is exceeded, the Web user is automatically disconnected and an error message is displayed. This number can be modified for each site in the "Sites" tab.
- **Duration of AWP contexts:** Validity period of the AWP contexts. As soon as the specified duration is over and if no new request was performed, the context file is deleted.
- **Maximum duration of a request:** Maximum time-out allowed between the beginning of an action performed by the Web user (via a button or a link) and the display of a response. When this time is exceeded, an error page is displayed but the Web user is not disconnected.

The WebDev administrator is used to manage a specific log file allowing you to follow the traffic statistics.

- **Generate a log file to manage traffic statistics (.log):** This option enables you to generate a log file (".LOG" extension) used to view the traffic statistics of the WebDev sites via WDStatistic. See the help about WDStatistic for more details. The location of the log file can be specified in the "Directory" control. This option can be modified for each site in the "Sites" tab.
- **Save in the global file and in the file of each application:** This option is used to save the statistics both in the log file of the application and in the log file of the server. Indeed, if the log file is defined both at server level and at application level ("Sites" tab), some information may be "lost".

#### Managing the remote debugging

The WebDev administrator is used to specify whether the remote debugging of the dynamic WebDev sites is allowed or not.

If the remote debugging is allowed, the following parameters must be specified:

- Main port (27 271 by default).
- Range of session ports (between 27 282 and 27 289 by default).

The "Default" button is used to restore the default values.

#### 2.5 Management

The "Management" tab is used to:

- Configure the server for the site setups and updates.
- Configure the error messages displayed in the browser.
- Manage the WebDev accounts.

#### Setup/Update

These options are accessible in deployment version. They allow the hosting company to authorize and configure the site setups and updates remotely.

The "**Lock the server (for update)**" option is used to simplify the update of a site by forbidding any connection of a new user. Whenever attempting to connect to one of the sites found on the server, the Web user receives a message informing him that the site is temporarily unavailable and asking him to retry later. The Web users who are already connected can continue to use the current WebDev site.

#### Error message

The "Errors" button displays the different error messages that can be displayed in the browser of the Web user. You have the ability to customize the error message and the HTML page where the error message is displayed.

#### 2.6 Advanced options

The "Advanced" tab is used to:

- Choose one of the Web servers installed on the current computer to run the WebDev sites.
- Perform a diagnostic regarding the configuration of the current computer.
- Specify the name or IP address of the current computer.
- Display in the browser a page used to start all the WebDev sites installed on the current computer.
- Manage the emails in asynchronous mode.
- Manage the server sockets.
- Manage the prints.

#### Server used

The "Server" button is used to choose one of the Web servers installed on the current computer to run the WebDev sites. Caution: This server will be used for all the dynamic WebDev sites installed on this computer.

If the server used is not found in this list, choose "Other". You will have to manually configure the Web server used. See the online help for more details.

#### Diagnostic

The "Diagnostic" button is used to check the configuration of the current computer. This diagnostic is used to check:

- whether the FTP/IP protocol is installed
- whether a Web server (HTTP server) is started
- whether the manager of the AWP protocol is configured properly
- whether the necessary WebDev executables are found.

The dynamic WebDev sites will not operate if one of these conditions is not fulfilled. See the online help for more details.

#### Name or IP address of the current computer

The WebDev administrator allows you to specify a name or an IP address to identify a specific computer. You also have the ability to specify the port number. For example: localhost:8080.

This computer will be used:

- when clicking the "Diagnostic" button.
- when clicking the "Test page" button.
- when running the test of a site, page, report, ... from the WebDev editor ("GO" icon).

#### Test

The "Test page" button is used to display in the browser a page allowing you to start each one of the dynamic WebDev sites and webservices installed on this computer.

See "Test of a site in practice", page 202 for more details.

#### Email spooler

If this option is checked, your sites will be able to send emails without locking the execution of the different processes. The asynchronous mode must be enabled when starting the email session (by **EmailStartSMTPSession**).

If the asynchronous mode is enabled, all the outgoing emails will be redirected to a "Spooler". The emails are queued up before they are sent.

Note: the email spooler is cleared when the Web-Dev administrator is closed: the pending emails will not be sent and they will be removed from the spooler.

If "Disable the email spooler" is checked while emails are still found in the spooler, these emails will not be lost: the administrator continues to send them but no new email will be accepted by the spooler.

**Sockets**

If "Allow the server sockets" is checked, your sites will be able to handle the server sockets (via the Socketxxx functions of WLanguage).

**Printer used by default (Intranet sites only)**

The "Configure the prints" button enables you to select the printer used by default when printing on a local or network printer of the Web server.

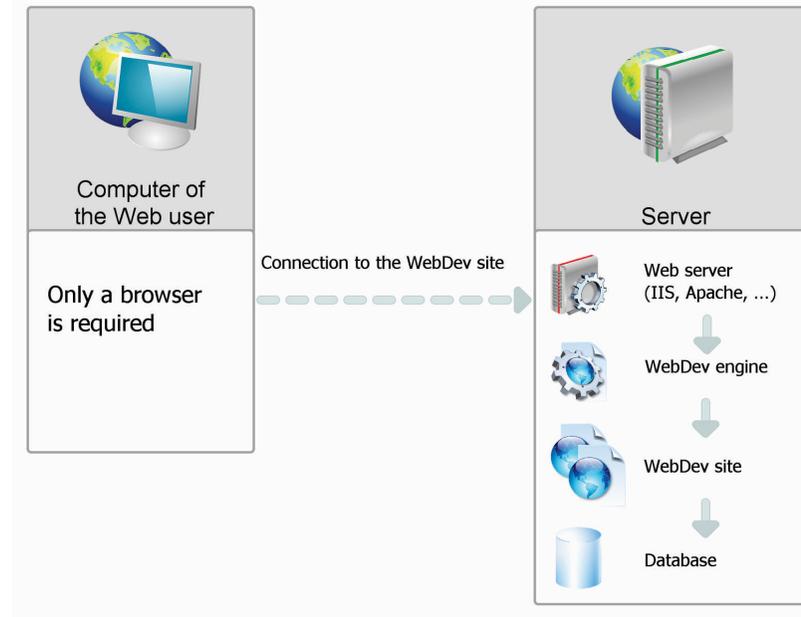
**Note:** if a network printer is selected, you need to define the print rights for this printer.

**Finding the expired pages**

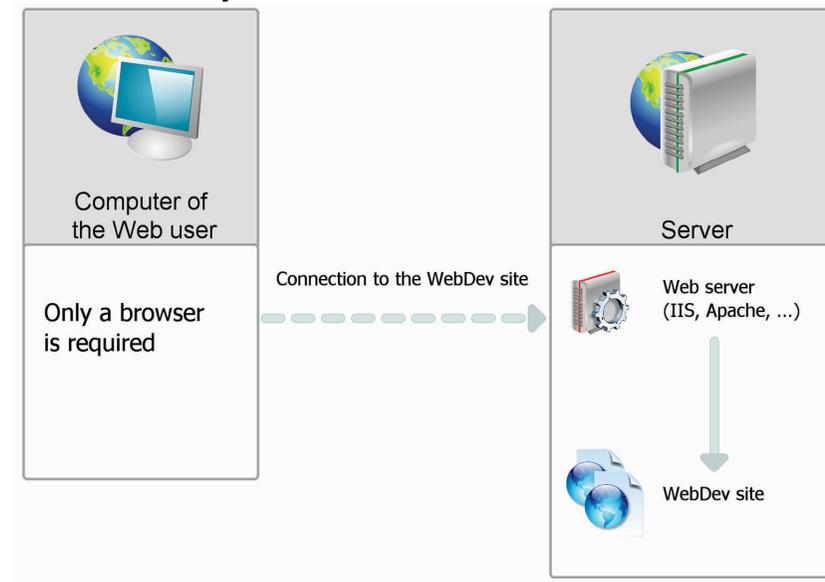
This option is used to enable and configure the search for expired pages on the WebDev sites managed by the WebDev application server.

# Configuring the server

**For a dynamic WebDev site:**



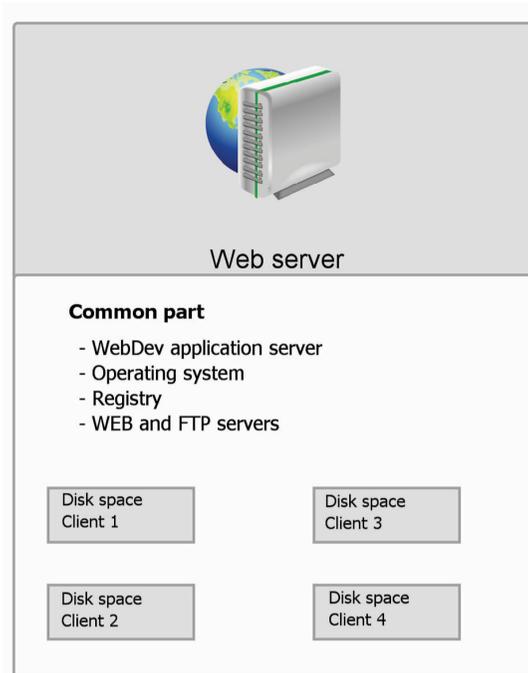
**For a static or semi-dynamic site:**



# Dedicated server or shared server?

The hosting companies propose two types of servers for installing your WebDev sites:

- **shared server:** server whose resources are shared by several clients (client = company hosting its site at a service provider)



**Characteristics:**

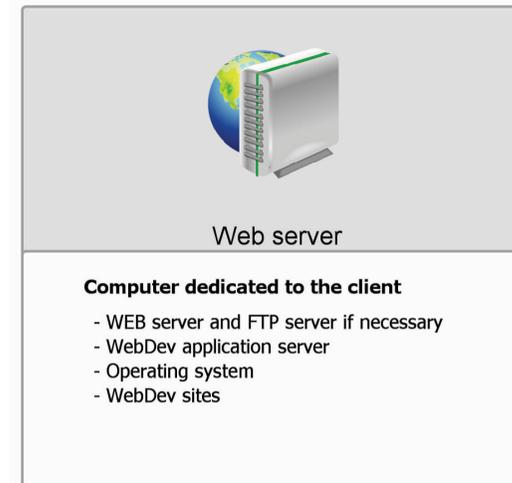
The hosting company provides a server to several clients.

The hosting company:

- configures the FTP server.
- defines the maximum number of connections for all the dynamic sites belonging to the same client (via the WebDev account manager).

The setup and the updates of a dynamic or static site can be performed by FTP.

- **dedicated server:** the server is reserved for one customer. The hosting company provides the Internet access only (IP address, cable, ...). The computer is either rented by the hosting company, or provided by the client.



**Characteristics:**

The hosting company provides the Internet access only (IP address, cable, ...).

The client must:

- provide the server (or rent it)
- configure the computer (system, ...)
- install the Web server
- install and configure the WebDev sites (by FTP or by CD-ROM)

# What type of server should I choose?

The table below presents the benefits and drawbacks of the different types of servers. Regardless of the server type, a deployment license is required for each server. Each server hosts one or more dynamic WebDev sites.

Server	Benefits/Drawbacks	When should I use this server?
<b>Shared</b>	<p>Main benefits:</p> <ul style="list-style-type: none"> <li>- Lower cost</li> <li>- Server managed by the hosting company</li> </ul> <p>Main drawbacks:</p> <ul style="list-style-type: none"> <li>- Memory resources shared by all the sites found on the server.</li> <li>- If a site is locked, all the sites found on the server are locked.</li> <li>- Stand-alone executables can rarely be used.</li> <li>- Non-secured access to the data: Unauthorized people can access the data: the host and possibly other clients if the server is not configured properly. Tip: encrypt the data files to avoid any unauthorized use.</li> </ul>	<p>A shared server is recommended for:</p> <ul style="list-style-type: none"> <li>- <b>the sites with few hits</b> (up to 20 or 30 simultaneous connections),</li> <li>- the sites requiring little memory (no large calculations performed on the server),</li> <li>- the sites in startup phase.</li> </ul>
<b>Dedicated</b>	<p>Main benefits:</p> <ul style="list-style-type: none"> <li>- The server resources are available for the sites installed on this computer.</li> <li>- Custom management of data download:               <ul style="list-style-type: none"> <li>- replication by emails, - WDREPLIC, - file copy by FTP.</li> </ul> </li> </ul> <p>Main drawbacks:</p> <ul style="list-style-type: none"> <li>- High cost</li> <li>- Server entirely managed by the client in most cases.</li> </ul>	<p>A dedicated server is recommended for:</p> <ul style="list-style-type: none"> <li>- <b>sites with many hits</b>,</li> <li>- the sites requiring a lot of disk space and memory,</li> <li>- the sites managing sensitive data,</li> <li>- the dynamic sites.</li> </ul>

# Hosting Control Center

Mainly intended for hosting companies and for Webmasters, the Hosting Control Center helps you easily host WebDev sites.

The control center manages the WebDev accounts as well as the Web server account and the rights for the operating system.

A default choice is proposed: it enables you to easily install a server, with no particular knowledge.

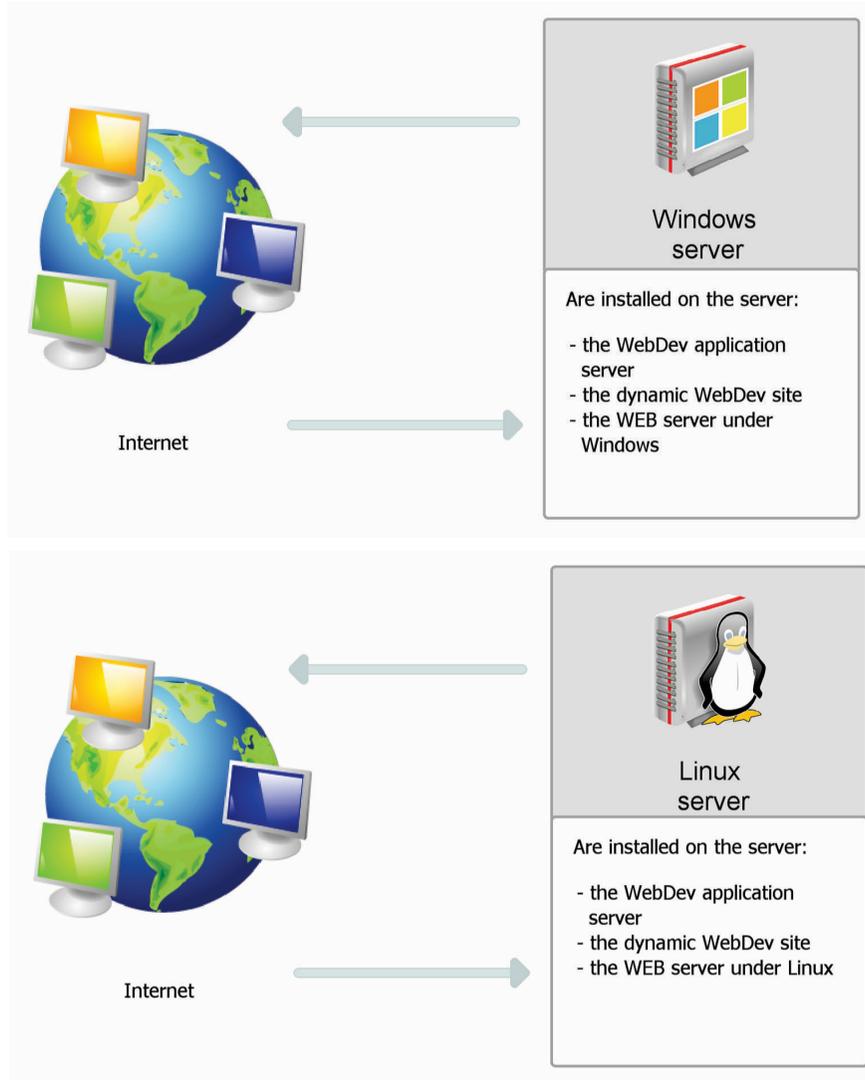
The setting of the following elements is centralized:

- Windows accounts,
- Groups of FTP users,
- Groups of WebDev clients,
- Home directory,
- FTP alias,
- WebDev accounts,
- Data directories,
- Virtual Web sites, ...

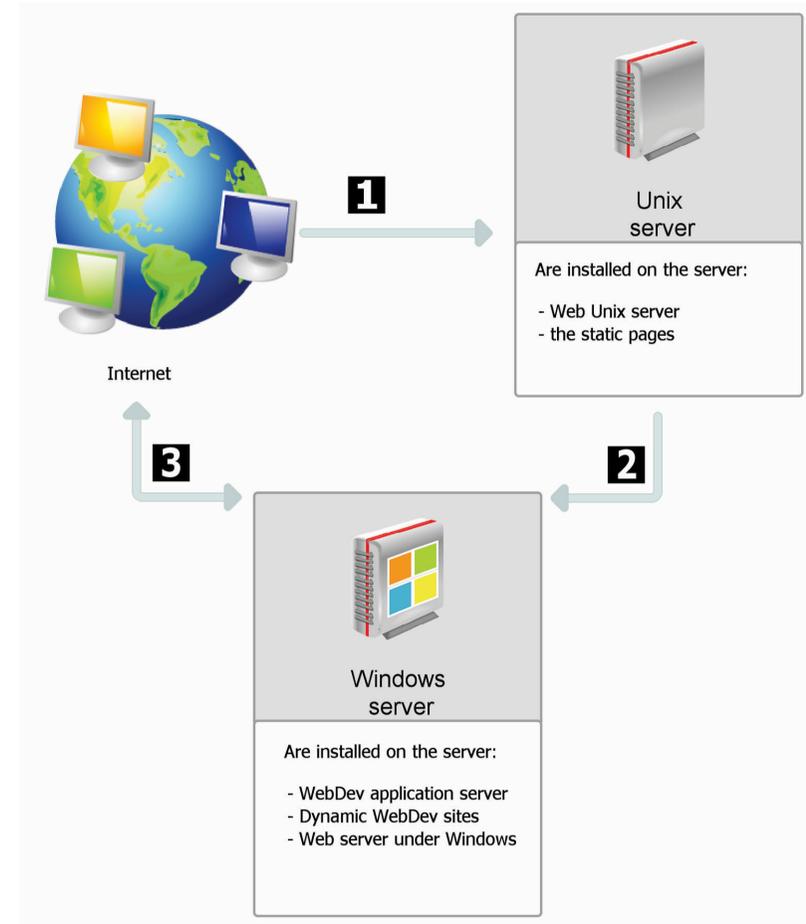
The WebDev Hosting Control Center operates in the Windows environment with the IIS Web server (Internet Information Server, all versions from version 5).

# Dynamic site on specific configurations

## 1. Setup on a Web server in Windows (2000 or later) or a Linux server



## 2. Setup on a Windows server, with an access to Internet by UNIX

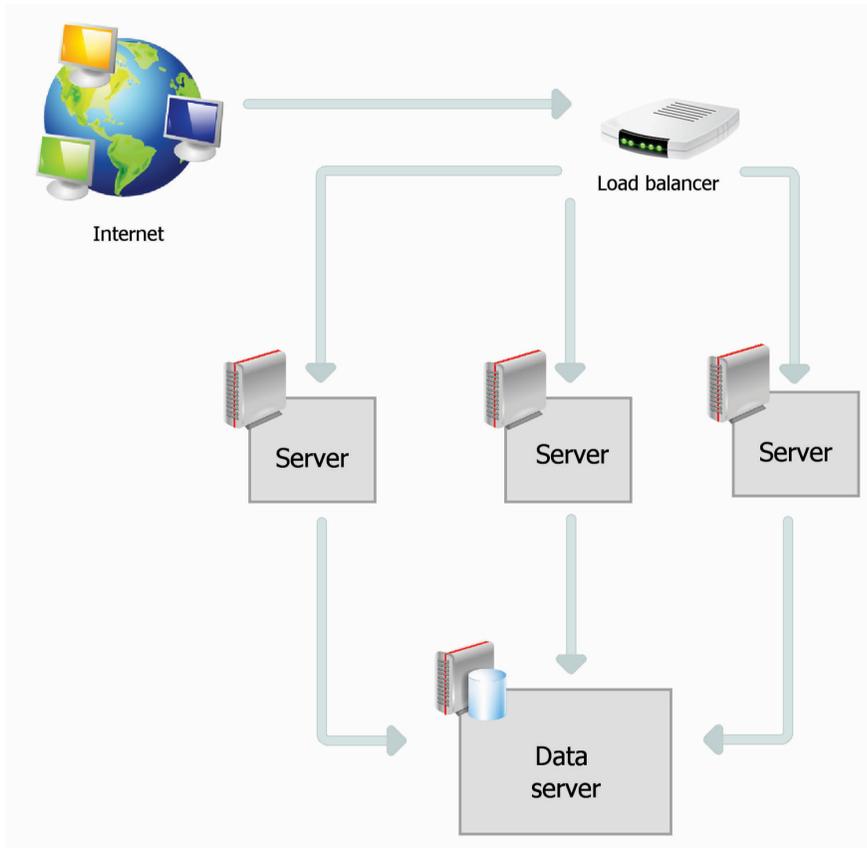


1. Opening a static page. The static page is found on the UNIX server.
2. Starting the dynamic WebDev site (found on the Windows server) via a link found in the static HTML page.
3. Standard operating mode of the WebDev site.

**Note:** The same configuration can be used with a Linux server.

### 3. Setup on a Windows or Linux server, with Load Balancer

The Load Balancer system is used to spread the number of connections to a dynamic WebDev site over several computers.



**Solution 1: Installing the dynamic WebDev site on the data server only.**

This solution consists in installing:

- the WebDev application server on each Web server
- the dynamic WebDev sites and the data files on the data server.

**Solution 2: Installing the dynamic WebDev site on all the Web servers.**

This solution consists in installing:

- the WebDev application server and the dynamic WebDev sites on each Web server.
- the data files of the dynamic WebDev sites on the data server.

# Traffic statistics of dynamic sites

It is often important to have precise statistics on site visits.

How do you find out who visited your site, which pages were looked up, which applications were used, which site was visited by the Web user?

WebDev is supplied with a tool used to perform statistical calculations for the dynamic pages: WStatistic. This tool can be installed on any computer.

Examples of statistics calculated by WStatistic: number of connections per day, visited pages, operating systems and browsers of the Web users, ...

Examples of statistics for the dynamic sites:

The screenshots show the WStatistic interface. The top screenshot displays 'Technical data' with a table of browser usage and a pie chart. The middle screenshot shows a 'Dashboard' with a line graph of traffic over time. The bottom screenshot shows 'Geographical data' with a world map and a table of visits by country.

Browser	Total	%
Internet Explorer 7.0	976	31,77%
Firefox 2.0	908	29,56%
Internet Explorer 6.0	654	21,29%
Other (random...)	245	8,63%
Firefox 1.5	146	4,75%
Yahoo Bot	32	1,04%
Firefox 1.0	32	1,04%
Safari	10	0,32%
Opera 9.2		
MSN Bot		
Yola Bot		

Country	Total	%
France	2.280	74,22%
Other country	209	6,67%
USA	140	4,56%
Belgium	70	2,28%
Canada	60	1,93%
Switzerland	60	1,93%
Algeria	42	1,37%
Turkey	42	1,37%
Bosnia and Herzegovina	25	0,81%
Romania	25	0,81%
Spain	24	0,76%

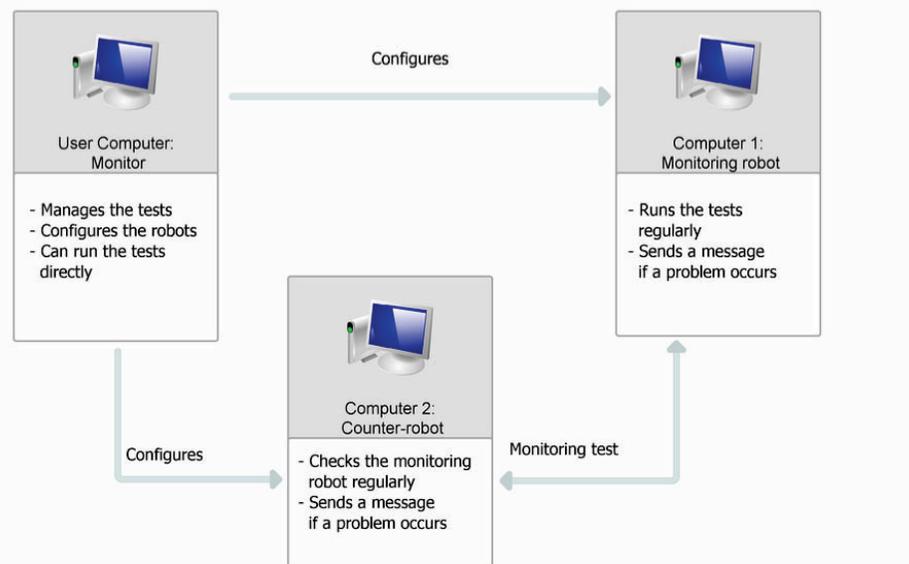
# Monitor your sites, servers, ...

A site depends on several external parameters: Web site, HyperFileSQL server, network, ... To optimize the management of incidents, WebDev proposes to use a monitoring robot. Made of three executables started on different computers, the monitoring robot is used to run different tests: Internet tests, network tests, ...

If problems occur when running a test, different methods can be used by the monitoring robot to inform you:

- Message sent in the PC SOFT messaging (WDBAL).
- Email
- Execution of another program.

Furthermore, a sound alert can be implemented on the monitor.



### Available types of tests

- |  |  |  |
|--|--|--|
|  Internet test (test of a Web site) |  Test of HyperFileSQL Client/Server servers |  Network test (SNMP)        |
|  Test of FTP server                 |  Test of news server                        |  Test of WinDev application |
|  Custom test (in WLanguage)         | ...  |  |

## PART 9

# Appendices



# WLanguage functions specific to WebDev 17

(Details in the online help of WLanguage)

AddFavorite	Adds an Internet address into the list of favorites for the Web user.
AJAXAsynchronousCallPending	Used to find out whether a server procedure called by AJAXExecuteAsynchronous is currently run.
AJAXAvailable	Used to find out whether the AJAX technology is supported by the current browser.
AJAXCancelAsynchronousCall	Cancels the automatic execution of the browser procedure called by AJAXExecuteAsynchronous.
AJAXExecute	Runs a server procedure without refreshing the page.
AJAXExecuteAsynchronous	Runs a server procedure without refreshing the page.
ASPDisplay	Calls an external ASP script and returns the result page in the current browser window.
ASPEXecute	Calls an external .asp script and returns the result in a string.
BrowserClose	Closes the current window of the browser and stops the execution of the current code.
BrowserIPAddress	Returns the IP address of the computer of the Web user connected to the WebDev site.
BrowserIsConnected	Indicates whether the browser is connected to the network.
BrowserName	Returns the name of the browser of the Web user.
BrowserOpen	Opens a new window of the browser.
BrowserRunApp	Starts the native Web browser of the application
CancelAWPContext	Deletes from the AWP context a variable added by DeclareAWPContext.
CaptchaDisplay	Displays a new captcha in a captcha control.
CaptchaVerify	Checks whether the value typed by the user corresponds to the string displayed in a captcha control.
CellCloseDialog	Hides a cell displayed by CellDisplayDialog.
CellDisplayDialog	Displays a cell in a page with a DDW effect (Dim Disabled Windows).
CertificateClientInfo	Returns information about the certificate used by the client computer.
ChangeAction	Used to specify the action to perform when the HTML page displayed in the browser is no longer synchronized with the page context on the server.
ChangeTarget	Changes the target frame of the page.
ColorPalette	Reads a color found in the current palette.
ConfigureAWPContext	Defines the operating mode of the AWP contexts.
ConnectionCount	Returns the number of instances of the WebDev site currently run on the server.
ContextClose	Closes a page context.
ContextExist	Used to find out whether a page context exists on the server.
ContextOpen	Opens a new page context without returning the information to the browser.
CookieRead	Retrieves the value of a cookie saved on the computer of the Web user.

CookieWrite	Writes a cookie on the computer of the Web user during the next display of a WebDev page in the browser.
CurrentPage	Returns the name of the page containing the WLanguage code currently run.
DeclareAWPContext	Declares a list of variables whose value will be persistent between the successive displays of AWP pages.
DeleteAWPContext	Deletes from the AWP context a variable added by DeclareAWPContext
DynamicSiteDisplay	Displays a dynamic site (created by WebDev) in the browser of the Web user from a dynamic or static WebDev page.
EmailOpenMail	Opens the default messaging software of the Web user.
EmailStatus	Returns the status of an email sent via an SMTP session opened in asynchronous mode.
FileDisplay	Returns a specific file to the browser.
FileToPage	Automatically initializes the controls of a page with the values of the associated items found in the current record (loaded in memory) of the HyperFileSQL file.
FolderData	Automatically initializes the memory value of the items found in a data file with the value of the page controls and/or the value of the WLanguage variables with the value of the page controls.
FolderWeb	Returns the path of the directory containing the images, the Javascript files, the Java applet files and the other files accessible from the browser ...
FramesetDisplay	Displays a WebDev frameset in the browser of the Web user.
FramesetRefresh	Refreshes a frameset displayed in the browser of the Web user from the context found on the server.
FramesetUse	Displays a WebDev frameset in the browser of the Web user and closes all the current page contexts and frameset contexts.
fWebDir	Returns the physical name of the directory containing the images, the Javascript files and the Java files of the WebDev site.
GadgetCloseFlyout	Closes the popup area of a Vista gadget.
GadgetDisplayFlyout	Displays a page of the Vista gadget in a popup area.
GadgetLoadParameter	Loads a persistent value in a Vista gadget.
GadgetSaveParameter	Saves a persistent value in a Vista gadget.
GaugeActivate	Enables the refresh of an Ajax indicator.
GaugeDeactivate	Stops the periodic refresh of an Ajax indicator.
GaugeExecute	Starts a long browser process and makes an Ajax indicator move forward according to the progress of this process.
GPSFollowMovement	Asks to be regularly notified about the current position of the device.
GPSGetPosition	Retrieves information about the current position of the device.
GPSLastPosition	Retrieves the information about the last known position of the device.
GPSStatus	Retrieves the activation status of the geo-localization provider or asks to be notified when the status changes.
gpwOpenConfiguration	Opens the page for configuring the user groupware.
grlImageSize	Defines the size of the image containing the chart.
IdentifierAWPContext	Returns the identifier of the AWP context.

iDirImageHTML	Used to select the directory of generated images during a print in HTML format.
ImageArea	Returns the number of the image area clicked by the Web user (clickable image only).
ImageFirst	Displays the first image.
ImageLast	Displays the last image.
ImageNext	Displays the next image.
ImageOccurrence	Used to find out the total number of images.
ImagePrevious	Displays the previous image.
ImageScrollingPosition	Returns the image displayed.
ImageStartScrolling	Starts the automatic scroll of images.
ImageStopScrolling	Stops the automatic scroll of images.
ImageXPos	Returns the horizontal position of the mouse cursor in relation to the relevant image control.
ImageYPos	Returns the vertical position of the mouse cursor in relation to the relevant image control.
JSEndEvent	Deletes the association between a WLanguage browser function and an event.
JSEvent	Associates a browser procedure with an event on an object in browser code.
JSInfoEvent	Used to handle the JavaScript properties of the browser event that triggered the execution of the code.
JSInterruptEvent	Interrupts the process of the current event.
JSMethod	Used to run a Javascript method on an element found in the current page.
JSONExecute	Calls a server URL of the same domain that returns data in JSON format (JavaScript Object Notation).
JSONExecuteExternal	Calls an external server URL that returns data in JSON format (JavaScript Object Notation). The data is processed in a specific procedure.
JSPProperty	Used to handle specific features on the objects found in the current page.
LocalStorageAdd	Adds a value to the local storage.
LocalStorageAvailable	Indicates whether the local storage is available.
LocalStorageDelete	Deletes a value from the local storage.
LocalStorageDeleteAll	Deletes all the values from the local storage.
LocalStorageGet	Retrieves a value from the local storage.
LocalStorageOccurrence	Returns the number of values for the local storage.
LocalStorageValueName	Returns the name of a value for the local storage.
MenuAddURLOption	Adds a new menu option at the end of a menu. This menu option is used to display the page corresponding to the specified URL.
PageActivateDDW	Enables or disables the effect applied by WebDev to the disabled pages when displaying a modal page.
PageAddress	Used to find out the Internet address of a WebDev page.
PageCloseDialog	Closes the current page. This page was opened by PageDisplayDialog.
PageDisplay	Displays a page in the browser of the Web user.
PageDisplayDialog	Displays a page in modal mode. Used to establish a dialog with the user.

PageExist	Checks whether the specified page is currently displayed in the browser of the Web user.
PageInitialization	Resets (or not) the controls found in the current page and runs the initialization processes of the controls.
PageParameter	Returns the value of a parameter passed to the current page.
PagePosition	Scrolls a page up to position a control in the visible part of the page (top) in the browser.
PageRateDDW	Used to define the rate of gray for the disabled pages.
PageRefresh	Refreshes a page displayed in the browser of the Web user from the context found on the server.
PageSubmit	Validates the specified page and starts the execution of a button.
PageToASP	Sends the data found in a page currently displayed in the browser to an ASP server.
PageToEmail	Emails the data found in a page currently displayed in the browser.
PageToFile	Automatically initializes the memory value of the file items with the value of the page controls.
PageToJSP	Sends the data found in a page currently displayed in the browser to a JSP server.
PageToPHP	Sends the data found in a page currently displayed in the browser to a PHP server.
PageToSource	Automatically initializes the memory value of the items found in a data file with the value of the page controls and/or the value of the WLanguage variables with the value of the page controls.
PageUse	Displays a WebDev page in the browser of the Web user and closes all the current page contexts.
PHPDisplay	Calls an external PHP script and returns the result page in the current browser window.
PHPExecute	Calls an external .php script and returns the result in a string. An http request is performed.
PopupClose	Closes a popup page.
PopupDisplay	Displays a popup page.
PreviousPage	Returns the name of the previous page.
rssDisplay	Builds a RSS stream and returns the content of the RSS stream to the client. The result is displayed in the browser of the Web user.
SaaSCheckService	Checks the access rights of the user to a service of the SaaS site.
SaaSClientConnexion	Returns the characteristics of the connection to the database of the client.
SaaSConnect	Authenticates a user of a SaaS site toward the SaaS webservice that manages the site.
SaaSDisconnect	Disconnect the user of a SaaS site toward the SaaS webservice that manages the site.
SaaSIsConnected	Defines whether the connection to the SaaS webservice is still established.
SaaSReadSiteParameter	Reads an information specific to the client for the current SaaS site.
SaaSWriteSiteParameter	Saves a specific information for a SaaS site in the configuration of a client account.

ScriptDisplay	Calls an external script (.php, .asp, .mhtml or .mht) and returns the result page in the current browser window.
ScriptExecute	Calls an external script (.asp or .php) and returns the result in a string. An HTTP request is run: you have the ability to use a POST method or a GET method.
SiteAddress	Returns the Internet address for connecting to a dynamic WebDev site found on the same server.
SiteMapPathAddLink	Adds a link into a site map path.
SiteMapPathDeleteAll	Deletes all the links from a site map path.
SiteMapPathDeleteLink	Deletes a link from a site map path.
SiteMapPathInsertLink	Inserts a link into the site map path.
SiteMapPathModifyLink	Modifies a link in a site map path.
SourceToPage	Automatically initializes the controls of a page with: - the values of the associated items in the current record, - the values of the associated WLanguage variables.
SSLActive	Used to enable or disable the secured SSL mode.
StringDisplay	Returns a specific string (or a buffer) to the client browser in response to a request.
SysVersion	Returns information about the PHP version used on the current server.
UploadCopyFile	Saves on the server a file "uploaded" by the Web user.
UploadCurrentFile	Indicates the file currently uploaded via the upload control. This function is useful for the multi-file Upload controls.
UploadCurrentFileSize	Returns the size (in bytes) already sent for the file currently uploaded via an Upload control.
UploadCurrentFileSizeSent	Returns the size (in bytes) already sent for the file currently uploaded via an Upload control.
UploadDelete	Deletes a file from the list of files to upload: the file will not be uploaded on the server.
UploadDeleteAll	Clears the list of files to upload: no file will be uploaded on the server.
UploadFileName	Returns the name of a file "uploaded" by the Web user.
UploadFileSize	Returns the total size (in bytes) of a file found in an Upload control.
UploadSize	Returns the total size (in bytes) of the files already sent by the current upload via an Upload control.
UploadSizeSent	Returns the total size (in bytes) of the files already sent by the current upload via an Upload control.
UploadStart	Starts sending the selected files in an upload control. At the end of the upload, the list of files to send is cleared on the browser.

# Components supplied with WebDev

WebDev is supplied with several components. These components are supplied with their source code and with an example.

The list of the main components supplied with WebDev is as follows:

Name of the component	Description
Basket	Used to easily manage a basket in a WebDev site.
Documentation request	Used to manage a request for documentation (brochure, catalog, ...) in a WebDev site.
Gantt chart	Used to display a GANTT chart in a memory table or in a page included in the component.
Login	Centralizes the management of users and passwords.
PayBox	Used to install a system for secured payment via PayBox. Caution: this is not a recommendation by PC SOFT for using the solution for secured payment implemented by PayBox.
Search engine	Used to index documents (text, video, sounds, ...) according to an identifier managed by the user (can come from a HyperFileSQL data file, ...) and keywords (text found in a document, file name, etc). Also used to perform some searches in the indexed documents and to generate some statistics (unused keywords, number of documents per keyword, etc).

Two sub-directories are specific to each one of these examples:

- the "<ComponentName>-Example" sub-directory contains an example of project that uses the component.
- the "<ComponentName>-Source" sub-directory contains the project of the component.

Additional components are supplied with the Technical Support Newsletter (LST) or can be downloaded from our site ([www.windev.com](http://www.windev.com)).

# Examples supplied with WebDev

The examples supplied with WebDev are intended to help you learn the features of WebDev.

Their source code is presented in details.

These examples are found in the "Examples" sub-directory of the WebDev setup directory and they can be opened directly from the "Wizards, Examples and Components" pane.

Different types of examples are supplied with WebDev:

- full examples: these examples correspond to complete sites and they can be used without any adaptation.
- educational examples: these examples illustrate a specific feature.

## Full examples

Name of the example	Description
After Sales Service	Management of dysfunction. Used to create, modify or delete suggestions and problems.
AfterSalesService_Php	Management of dysfunction in PHP. Used to create, modify or delete suggestions and problems.
Hawaii	Management of an online supermarket. The products can be purchased and stored in a virtual basket until payment.
Honolulu	Free Intranet portal: messaging, forum, blogs, ...
Internet_Catalog	Proposes a catalog site in AWP and a management site: <ul style="list-style-type: none"> <li>- Management of a basket</li> <li>- Management of promotions</li> <li>- Management of new products</li> <li>- Search for products</li> </ul>
NetAds	Management of classified ads. Used to perform multi-criteria searches, to add, modify and delete ads as well as to manage the users.
Rewali	Online travel booking.
WebDoc	Electronic Document Management (EDM). Uses the Search Engine component to perform searches by keywords.
WebDVD	Site for online DVD rental.
WebEstate	Site for selling and renting real estate.

WebFleet	Simulates the management of a computer fleet.
WebMillion	Uses a data file containing more than 1 million of records.
WW_AWPShowroom	Presents the use of a dynamic WebDev site associated with an AWP WebDev site.
WW_Blog_AWP	This example is a site for blog management, based on an AWP generation, allowing the site to be referenced by all the search engines. Furthermore, each blog can be exported in RSS.
WW_Blog_PHP	This example is a PHP site for managing blogs.
WW_CMS	This example is a CMS (Content Management System), typically a site for displaying articles.
WW_CommercialSite_php	This example of business site is a PHP site developed in WebDev.
WW_DrawLine	This example is used to draw charts.
WW_Forum_AWP	This example proposes the main features of a user forum (creation of forums, topics and messages, moderation by an administrator, ...).
WW_Forum_Classic	This example proposes the main features of a user forum (creation of forums, topics and messages, moderation by an administrator, ...).
WW_Forum_PHP	This example proposes the main features of a user forum (creation of forums, topics and messages, moderation by an administrator, ...).
WW_HTML_Editor	This example is developed for Microsoft Internet Explorer 5.5 (or later). It uses the HTML edit features of Internet Explorer.
WW_Images	This example is a simplified photo and image gallery.
WW_Images_php	This example is a PHP site. It is a simplified photo and image gallery.
WW_Loan	This example is used to simulate loans. The amortization tables can be edited in a file in PDF format.
WW_Pie	This example explains how to use the chart designer and how to associate actions with the mouse clicks performed on the chart sections.

WW_Schedule	This example is used to dynamically display the tasks assigned to the different contributors. You have the ability to add, modify or delete tasks by clicking the elements found in the schedule. All the operations are performed in AJAX.
WW_WebChat	Site for online chat. All the processes (sending messages, etc.) are managed in AJAX. Therefore, no refresh is performed in the pages.
WW_WebStockMarket	This example explains how to use the chart designer to draw stock charts.
WW_Zoom	This example explains how to handle images in browser code: zoom, move, manage the click on a map (a Michelin map in this case).

### Educational examples

Name of the example	Description
WW_Animated	Presents some animation and effect features proposed in WebDev and the WLanguage.
WW_AWPPager	This example explains how to manage a pager in an AWP page.
WW_ChangeCSS	This example explains how to modify the style sheet of a WebDev page by programming. This principle can also be used to change the interface of a site according to the preferences of the Web user.
WW_Click_on_Chart	Used to offset the sections of a Pie chart via a simple click.
WW_CloudTags	This example explains how to display a cloud of tags in a dynamic WebDev application.
WW_ConfigurableFlash	This example provides customizable Flash banners as supercontrols.
WW_CSSIntegration	Explains how to include special effects created with Javascript and CSS in a dynamic WebDev page.
WW_Drawing_HTML5	Drawing functions in browser code for the browsers that support the HTML5 standard.
WW_FadeInSlide	This example explains how to create a fade-in effect between 2 images.
WW_Flex	This example presents the possible interactions in browser code between a Flex application and a WebDev site.

WW_Frames	This example explains how to modify two different frames without redisplaying the entire frameset.
WW_FurtiveCells	This example explains how to create "furtive cells" in your WebDev sites
WW_Highlight	This example explains how the CSS can be used to frame and highlight words in an HTML static (or in an HTML control).
WW_iPhone	This example explains how to develop a mini WebDev site for iPhone.
WW_MapArea	This example presents the use of map areas in programming or via a preset action.
WW_Organizer	This example uses the Scheduler control in Organizer mode.
WW_PayPal_PHP	Explains how to propose a secured payment via the PayPal solution in your WebDev PHP sites.
WW_PHP_Controls	Example for using all types of controls available with WebDev during a PHP generation.
WW_RSS_Stream	This example explains how to read and display a RSS stream in a WebDev page via the RSS type.
WW_SAASClientSite	This example presents the functions for SaaS management found in WEBDEV.
WW_ScrollingText	This example proposes to use the ScrollingText supercontrol. This supercontrol is used to scroll text in a frame of your page.
WW_Server_Wait	Allows the Web user to start a server process while having the interface modified in order to be notified of the end of the server process
WW_Silverlight	Presents the possibilities of dialog between a dynamic WebDev site and Silverlight controls.
WW_StaticMap	Presents the interaction between a WinDev application and the Google Maps API service in order to display a geographical, political or road map of a given location.
WW_Upload	Explains how to use the upload control to: - transfer one or more files simultaneously, - display a progress bar during the transfer, - get information about the transferred files.
WW_WEB2	This example explains how to Drag/Drop controls into a Web page. It is used to move a product from a looper control into a basket (an internal page).

WW_WIKI	This example is a manager of WIKI documents. A WIKI document is a document that can be modified by all the persons who have access to this document.
WW_Wizard	This example includes a class and a page template that can be re-used to create and manage a Web wizard.
WW_XMLRestock	Used to manage the orders, modify the quantities ordered, validate and save the orders in XML format.
WWAjaxPHP	This example is an Ajax PHP site developed in WebDev. It mainly presents the Ajax tables and loopers.
WWBrowserDialog	This example presents a dialog between two distinct browsers.
WWConnect	This site is used to detect the closing of a client browser in order to save its disconnection or to perform specific operations.
WWDIALOG	This example presents the automatic dialog boxes as well as the custom dialog boxes.
WWDload	This site is used to propose files that can be downloaded by the users according to the group to which they belong.
WWEEnvironment	This site allows you to use the environment variables in order to retrieve information about the Web users.
WWFlash	This site is used to include a drop-down Flash menu that interacts with WebDev. Therefore, this site allows you to use Flash animations in a WebDev site. This site also presents the interaction with a Flash animation.
WWForm	This example explains how to dynamically change the appearance of a form found in a Web page.
WWKey	This program is used to detect the key pressed on the keyboard (OnKeyDown event) in Internet Explorer. This feature is implemented in a crossword.
WWLooper	This example presents the most common cases for using the loopers.
WWMenu	This example presents a menu developed in WebDev.
WWOrganizer	Explains how to retrieve the parameters passed, via PageParameter().
WWParam	This example explains how to pass and retrieve parameters to/from a site.

WWPocket	This example is a site that can be viewed on a Pocket PC. The site detects the resolution of the user and displays the pages consequently.
WWSlide	This site presents slides of Cantal (15).
WWTabControl	Presents the use of the Tab control.
WWTabulation	This site is used to intercept the use of the [TAB] key in order to insert a tabulation into an edit control.

Additional examples can be downloaded from our site ([www.windev.com](http://www.windev.com)).