

25

DEVELOP 10 TIMES FASTER

## How to discover WEBDEV?

WEBDEV is a powerful web-focused Integrated Development Environment that includes all the tools required for developing and maintaining websites.

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

- 1** Read the “Concepts” (online or on physical documentation)  
This guide presents the main concepts required to create a powerful WEBDEV site. Some concepts are followed by a “Practical” section that presents some features of the editor.
- 2** “Tutorial” (online tutorial + exercises)  
The tutorial provides a first “hands-on” approach to WEBDEV. It allows you to become familiar with the main editors of WEBDEV.
- 3** Test of examples  
Try the different examples provide with WEBDEV in the fields that interest you most (e-commerce, directory, scheduling, etc.).

Online help, accessible on <http://doc.windev.com> or installed with WEBDEV, allows you to quickly find the syntax of a WLanguage function and to get help about the interface, ... For each programming theme, you will find a description of the associated feature and the list of the corresponding WLanguage functions.

Remark: If there is a difference between the tutorials and the online help, follow the instructions in the online help.

**We hope you enjoy getting started with WEBDEV.**

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).

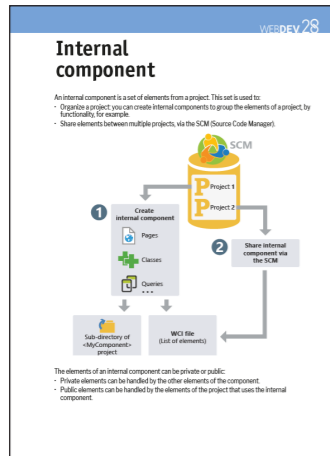
This documentation is not contractually binding. Modifications may have been made to the software since this guide was published. Please check the **online help**.

All product names or other trademarks mentioned in this publication are registered trademarks of their respective owners.  
© PC SOFT 2023: This publication may not be reproduced in part or in full without the express consent of PC SOFT.

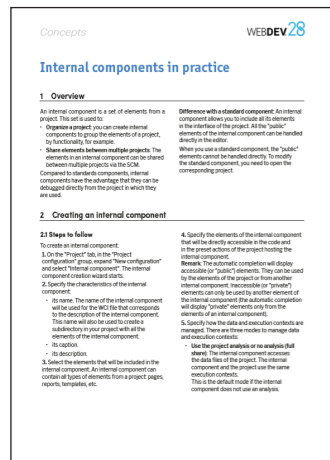
# Guide organization

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

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



Concept page



Implementation page

# Table of contents

## PART 1 - INTERNET: BASIC CONCEPTS

- What is a website? .....11
- Static or dynamic site? .....12
- Types of dynamic sites .....13
- Types of pages .....14
- How to display a dynamic WEBDEV site? .....18
- How to display an AWP site? .....20
- AJAX technology .....24
- If you are already familiar with WINDEV .....27
- Hardware and software requirements .....28

## PART 2 - DEVELOPING A SITE

- Project and analysis .....31
- Site development life cycle .....34
- I want to create a page in the editor .....36
- The following files are automatically created .....37
- Editing a page: zoning mode .....38
- "Mobile friendly" site: Dynamic serving .....39
- "Mobile friendly" site: Responsive Web Design .....40
- Internal page .....41
- Page templates .....42
- Popup page .....43
- Types of standard controls .....50
- Repeating a group of controls: loopers .....60
- Types of code .....61
- WLanguage: a simple and powerful language .....64
- Order of execution of the code of buttons/links .....65
- Upload: Sending files to the server .....67
- Control templates .....68
- Reports: concepts .....69
- Report templates .....70
- Print modes .....71
- Displaying and linking pages .....72
- Performing a process on several successive pages .....76
- Style sheets: to simplify the layout .....77
- Center or anchor a site horizontally .....78
- Site SEO .....79

**PART 3 - DEVELOPMENT ENVIRONMENT**

WEBDEV editors.....85  
 Project dashboard .....89  
 WEBDEV, WINDEV, WINDEV Mobile: 100% compatible .....90  
 Project configuration .....91  
 Multiple generation .....92  
 Source Code Manager (SCM) .....93  
 Internal component.....100  
 External component .....103  
 Generation modes .....110  
 Project Monitoring Center .....113  
 Managing requirements .....114  
 Managing tasks .....115  
 Managing business rules.....116

**PART 4 - ADVANCED CONCEPTS**

RAD (Rapid Application Development) .....119  
 UML models .....120  
 User Groupware .....126  
 Multilingual sites .....133  
 10 tips about ergonomics .....137  
 Overlaying controls .....141  
 Customizing the aspect of a site: skin elements.....142  
 Choosing a type of button/link.....144  
 Preventing from going back to a page .....145  
 Security benefit: managing the "Back" button .....146  
 Cookies: information stored on the end user's computer .....152  
 Protecting access to the site: passwords.....153  
 TLS/SSL: Encrypting information on the Web .....154  
 Secure payment .....155  
 Sending emails.....156

**PART 5 - DATABASES**

Analysis: Database structure.....161  
 Types of accessible files.....170  
 HFSQL Classic.....172  
 HFSQL Client/Server .....173  
 HFSQL Client/Server clusters .....174  
 HFSQL: files physically created .....175  
 Binding controls to data.....176  
 Queries: concepts .....178

Embedded queries.....179  
 The Table/Looper control.....180  
 Retrieving data from a site.....182  
 Universal replication .....187  
 3-Tier architecture .....188

**PART 6 - TESTING A WEB SITE**

Testing a site: Elements to be tested .....191  
 How to test a site? .....192  
 Debugger: principle .....194  
 When to use the debugger? .....195

**PART 7 - DEPLOYING A WEB SITE**

Deploying a static site.....205  
 Deploying a dynamic WEBDEV site .....207  
 Activating a dynamic WEBDEV site.....211

**PART 8 - HOSTING WEBDEV SITES**

What is the WEBDEV administrator used for .....221  
 Configuring the server.....228  
 Dedicated server or shared server? .....229  
 Which type of server to choose? .....231  
 Hosting Control Center .....232  
 Dynamic site on specific configurations .....233  
 Traffic statistics of dynamic sites .....237  
 Monitor your sites, servers, etc. ....238

**PART 9 - APPENDICES**

WLanguage functions specific to WEBDEV 28.....241  
 Examples and components provided with WEBDEV .....252

**PART 1**

**Internet:  
Basic concepts**

28

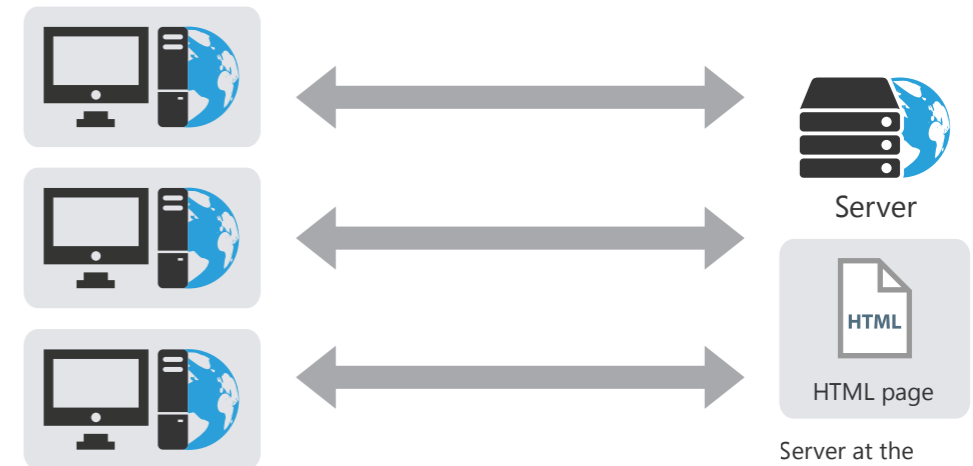
DEVELOP 10 TIMES FASTER

# What is a website?

A website 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, etc.).

A site is intended to be used by Internet users. Internet users use a simple browser to access the site.

WEBDEV allows you to easily create websites with or without data.



The Web user accesses the site and displays the HTML pages via a browser.

Server at the hosting company. The HTML pages are stored on this server.

# Static or dynamic site?

Several types of sites can be created:

- static sites,
- dynamic sites.

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

Static site	Dynamic site
<p>The content of the pages is fixed, it is defined only once.</p> <p>A static site cannot interact with data.</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>WEBDEV offers several types of dynamic sites:</p> <ul style="list-style-type: none"> <li>• WEBDEV site in Session mode.</li> <li>• WEBDEV site in AWP mode.</li> <li>• PHP site.</li> </ul>
<p>Static WEBDEV site: the WEBDEV Application Server is not required.</p>	<p>Dynamic WEBDEV site: the WEBDEV Application Server or the PHP language is required.</p>

**Remark:** A dynamic site can contain a static section (presentation of the company, etc.).

WEBDEV allows you to create:

- static pages,
- dynamic pages.

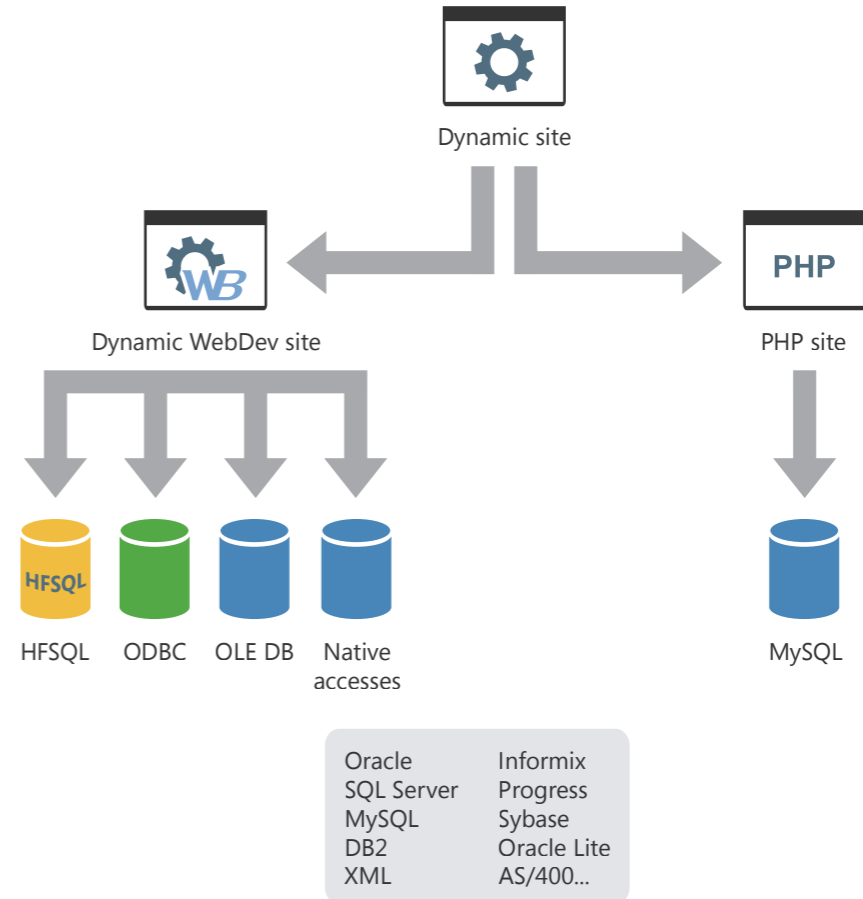
# Types of dynamic sites

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 bookings 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 (in Session or AWP mode).** 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.



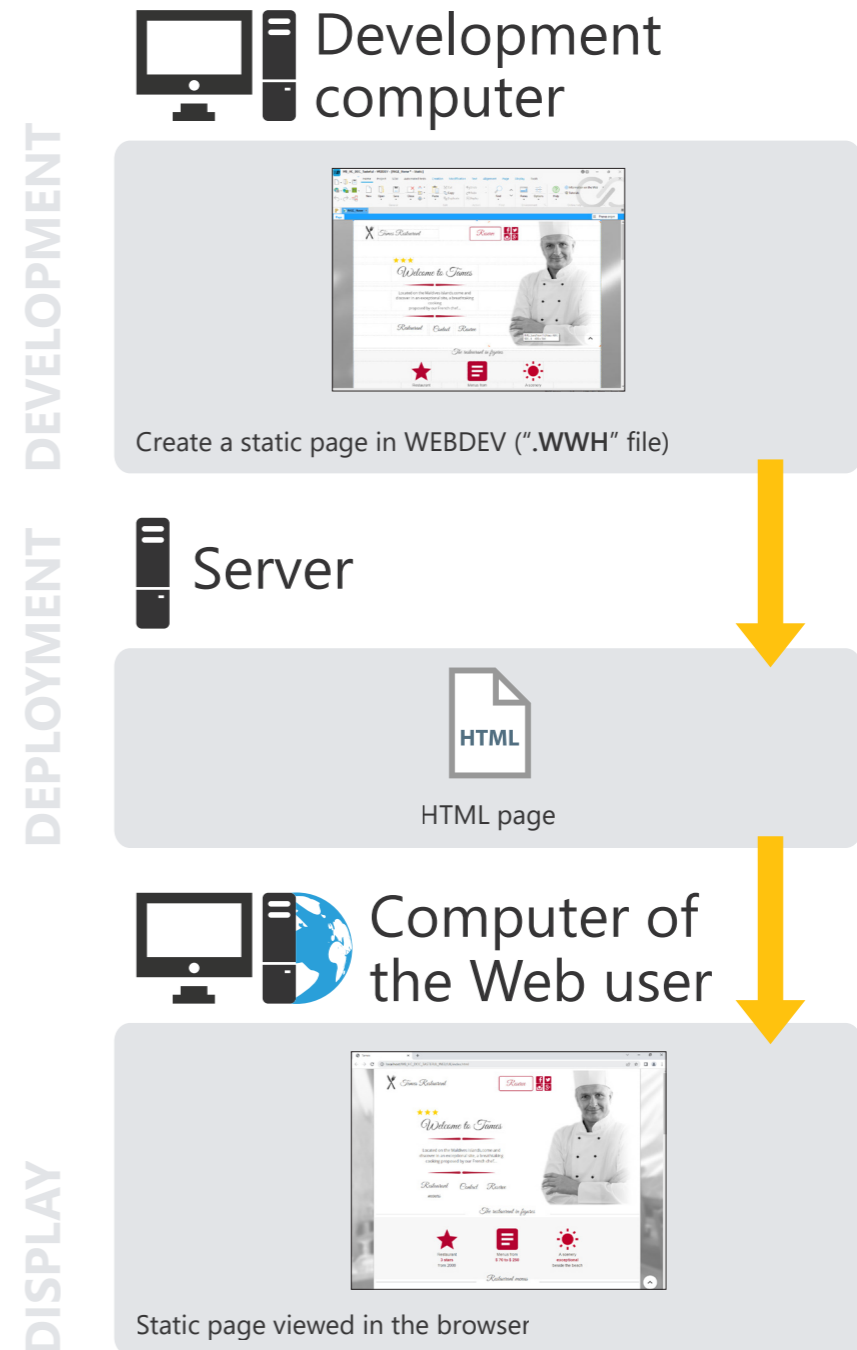
# Types of pages

Different types of pages can be used in the sites created with WEBDEV:

- **Static pages.** This type of page is used to display fixed data.
- **Dynamic WEBDEV pages in Session mode.** These pages are used to dynamically display the information found in a database. The content of the page depends on the displayed record. When a dynamic page in session mode is displayed, the associated page context is automatically created on the server. This page context mainly contains the global variables and the variables for positioning in the data files.
- **Dynamic Active WEBDEV Pages (AWP).** These pages are used to dynamically display the data found in a database. The content of the page depends on the displayed record. An Active WEBDEV Page is a dynamic WEBDEV page without a context persisting on the server. The Active WEBDEV Page context is temporary. It is created in a temporary session.
- **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.

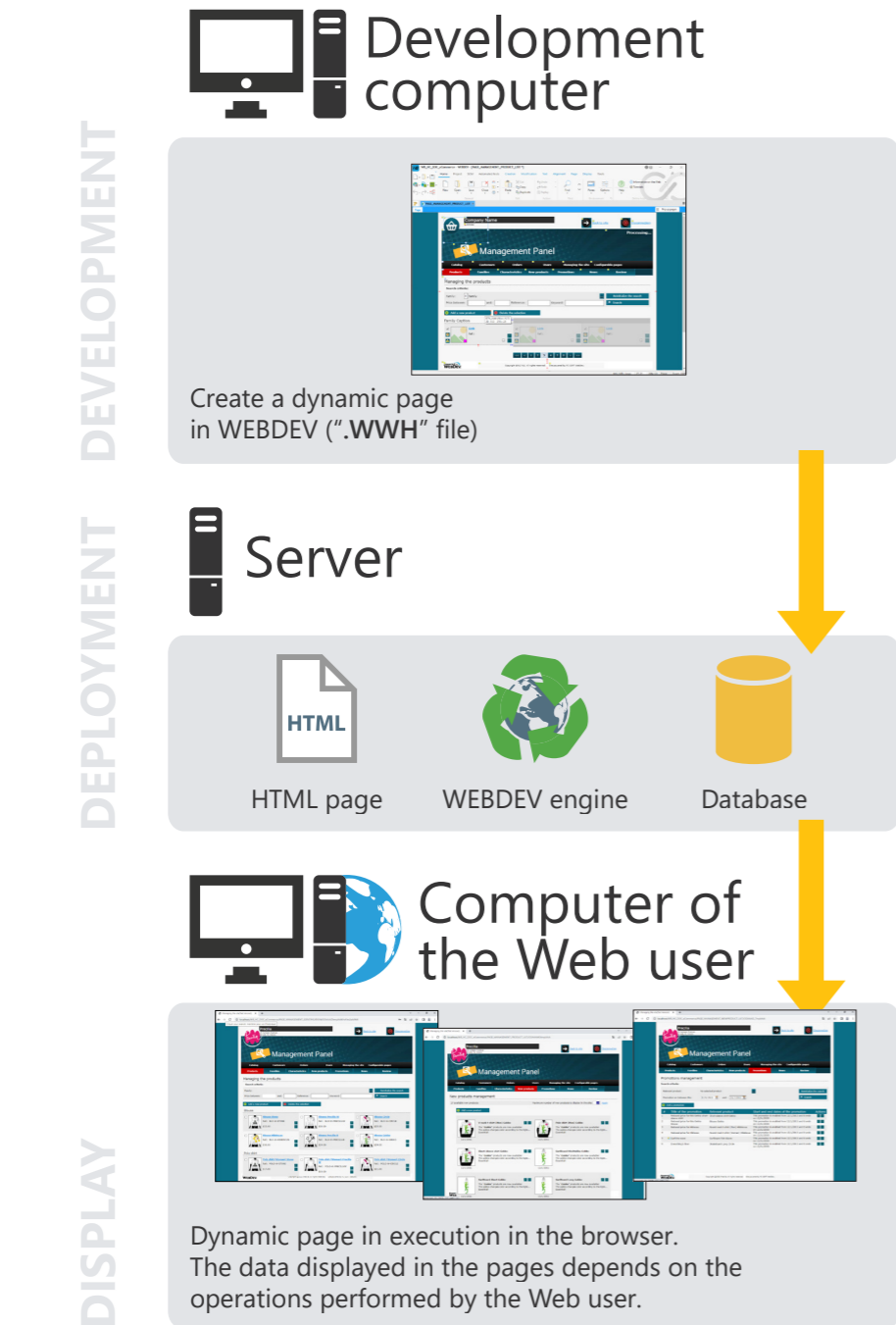
The following pages present these different types of pages.

## Principle of a static page

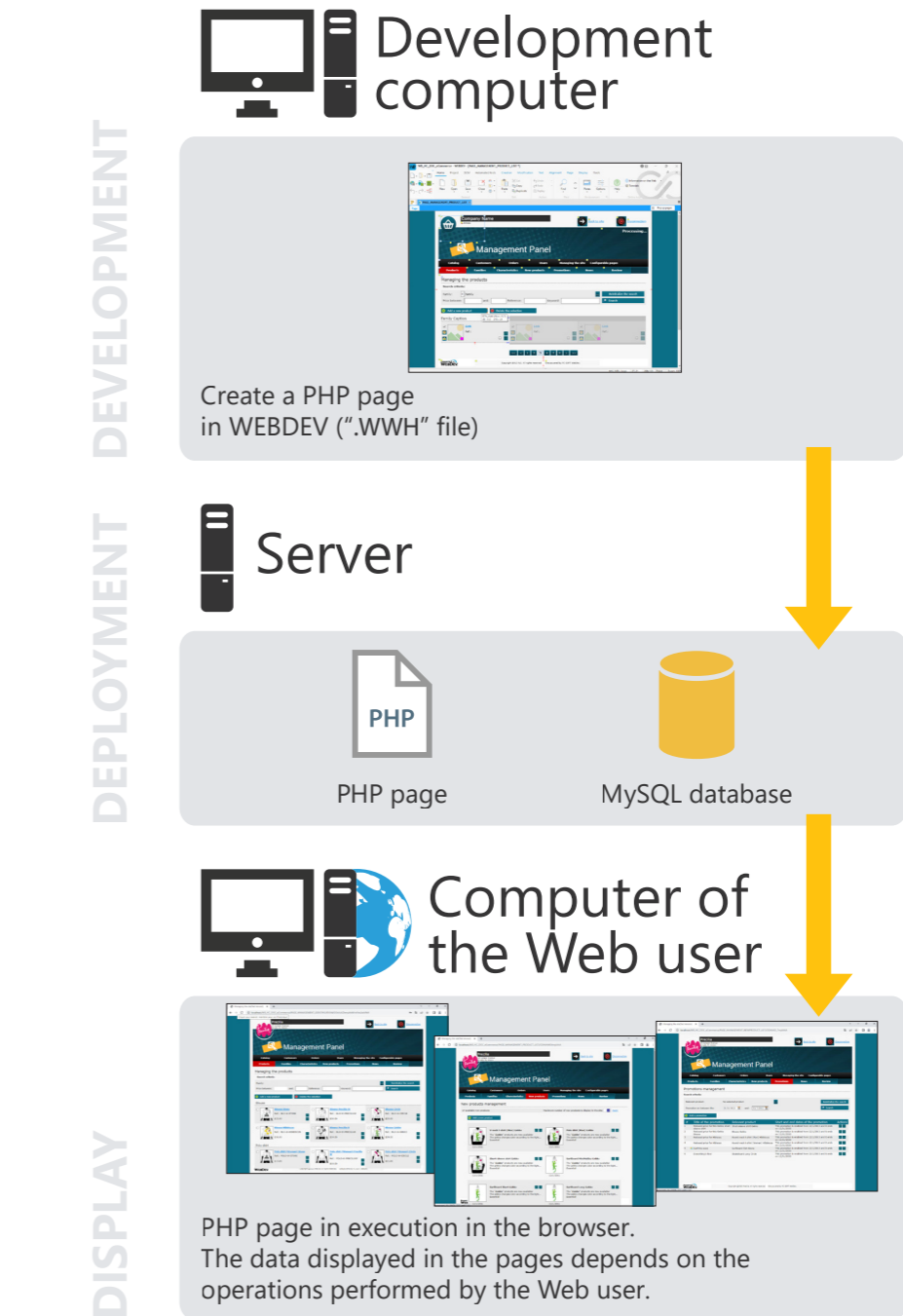




Principle of a dynamic page (Session or AWP)



Principle of a PHP type page



# How to display a dynamic WEBDEV site?

## WEBDEV session

The following operations are performed when displaying a WEBDEV page in Session mode:

1. Page display request.
2. The WEBDEV engine is started. This engine remains on the server until the application is closed.
3. The application context is created. This context remains on the server until the application is closed.
4. The WEBDEV engine executes the server code and builds the HTML page (e.g., from the data in the database).
5. Once the HTML page is entirely built, the server transmits the result to the client (the browser).

## Page contexts

A page context is automatically created on the server each time a page is displayed in the browser. This page context contains all the elements required to build the page viewed by the end user:

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

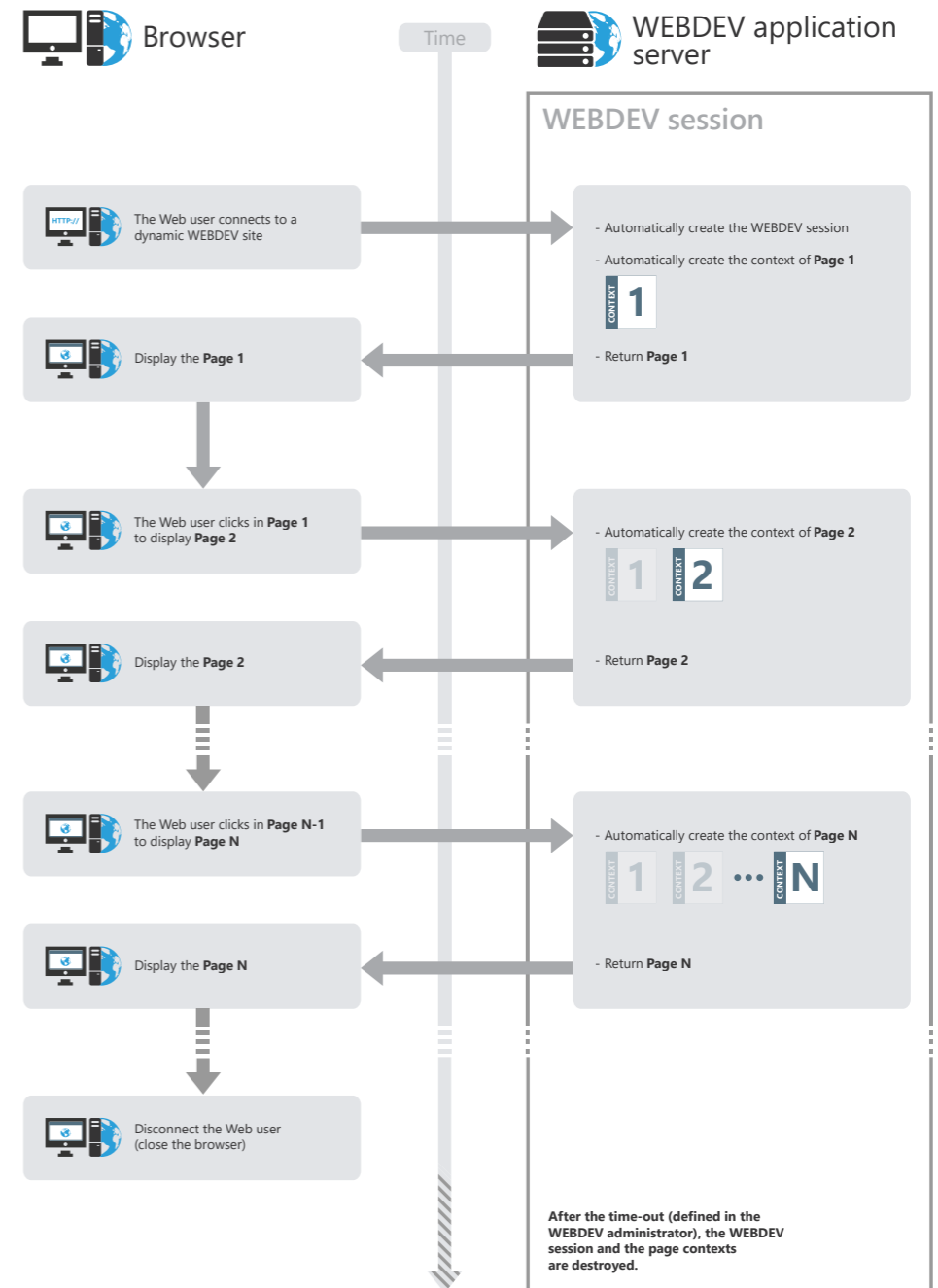
The page contexts remain in memory on the server until the end of the WEBDEV session.

When the same page is called several times:

- If **PageDisplay** is called to display the page, the page context is deleted and created again.
- If **PageRefresh** is called to display the page, the same page context is used again.

## Programming

The WEBDEV session and the page contexts are automatically managed by default. No programming is required.



# How to display an AWP site?

## What is an Active WEBDEV Page?

An Active WEBDEV Page is a dynamic WEBDEV page **without persistent context** on the server. The Active WEBDEV Page context is temporary. It is created in a temporary session.

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

## Operating mode of AWP sites

The following elements are automatically created on the server when an Active WEBDEV Page is displayed in the browser:

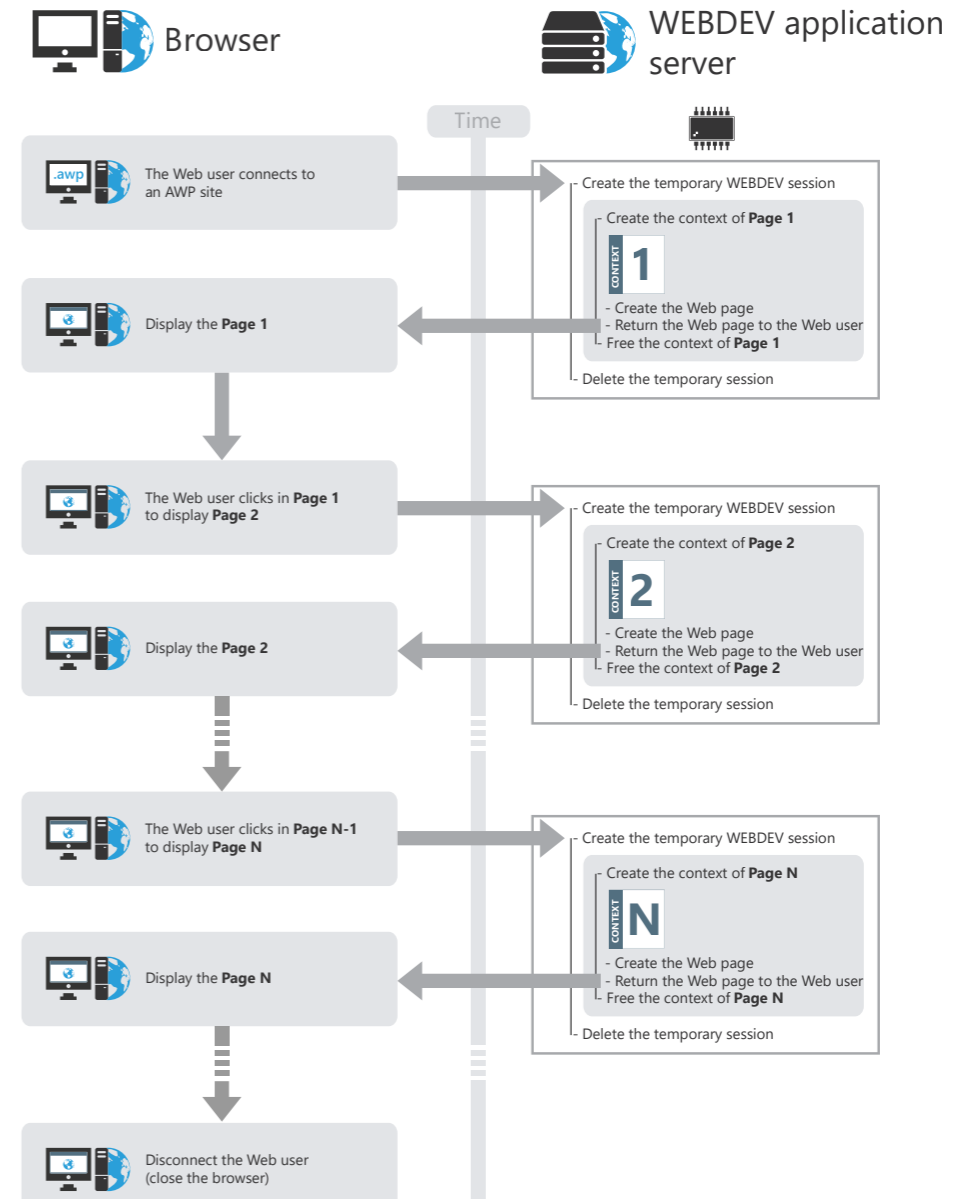
- a temporary session,
- a temporary Active WEBDEV Page context.

The **temporary session** contains the temporary Active WEBDEV Page context. When the Active WEBDEV Page is sent to the Web user, the temporary page context and the temporary session are destroyed. There is nothing left in memory on the server.

The **temporary Active WEBDEV Page context** contains all elements that have been required to build the page viewed by the Web user:

- local variables,
- server processes,
- database connections,
- contexts of data files, etc.

When the Active WEBDEV Page is sent to the Web user, these elements are destroyed.



### How to share information (values) between Active WEBDEV Pages?

There are two methods to share information (values) between Active WEBDEV Pages:

- Passing information in the URL. This method allows for better SEO.
- Saving information in AWP contexts (programmatically).

#### Passing information (values) between two pages in the URL

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

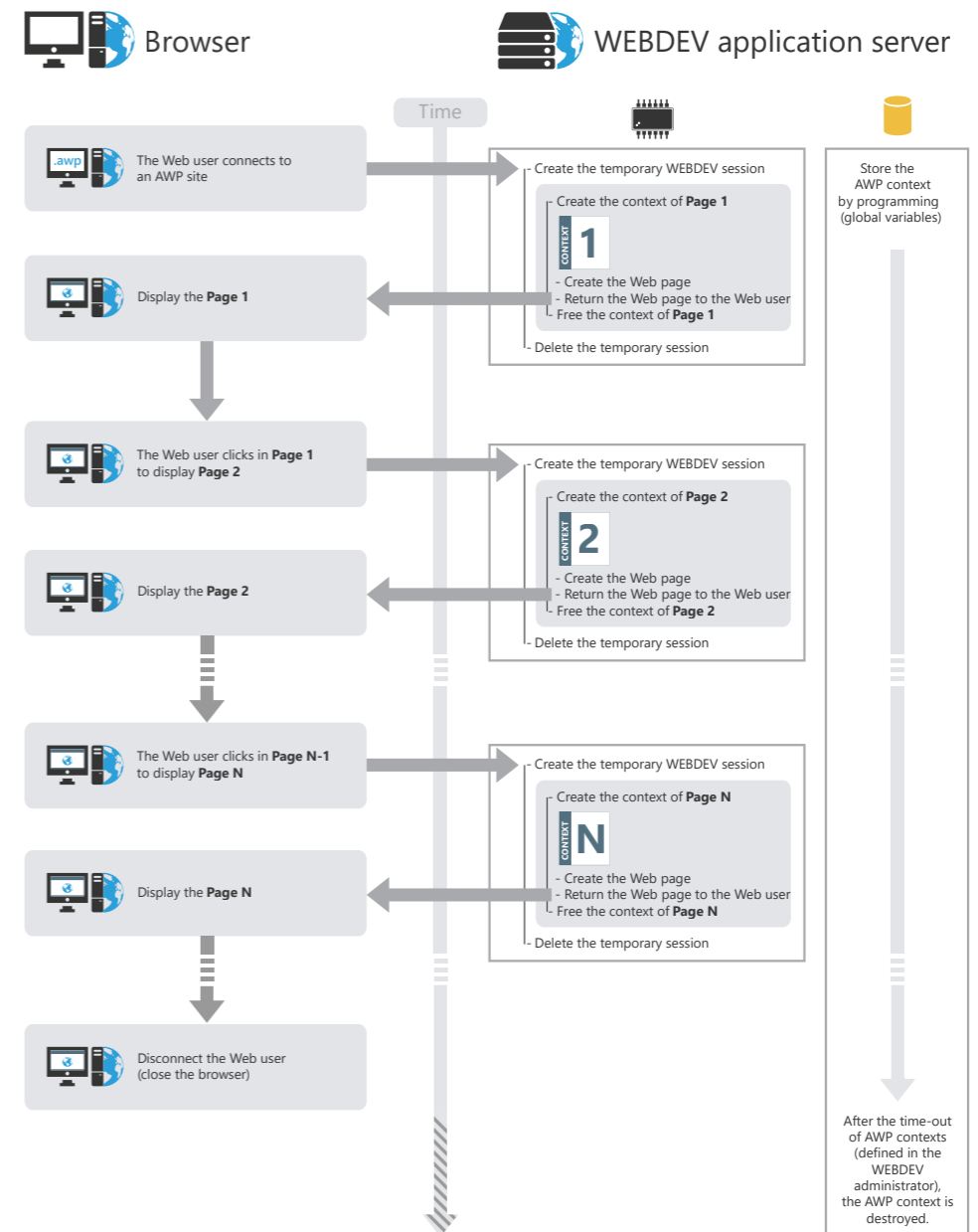
This method allows a better page SEO because the information passed in the URL is visible and analyzed by the SEO robots.

#### Saving information (values) in AWP contexts (programmatically).

You have the ability to store on the server values common to several Active WEBDEV Pages, via the **AWP contexts**. An AWP context is created on disk on the server. This context is available as long as Active WEBDEV Pages are displayed and as long as the timeout of AWP contexts is not exceeded. The timeout of AWP contexts can be defined in the WEBDEV administrator ("Configuration" tab, "Duration of AWP contexts" option).

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

For more details, see the online help.



# AJAX technology

The AJAX technology is available natively in WEBDEV.

## What is AJAX and what are its advantages?

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 in the page displayed (content of the cart, characteristics of a product, list of cities, map, etc.) are modified, only these elements will be refreshed. The server will not have to send the entire page to the user's computer.

This technology has several advantages:

- fewer requests are made to the server. Therefore, it can support a larger number of simultaneous connections.
- the information that circulates is less bulky.
- the transmission time is shortened.
- the user sees information immediately and without undesired visual effects.

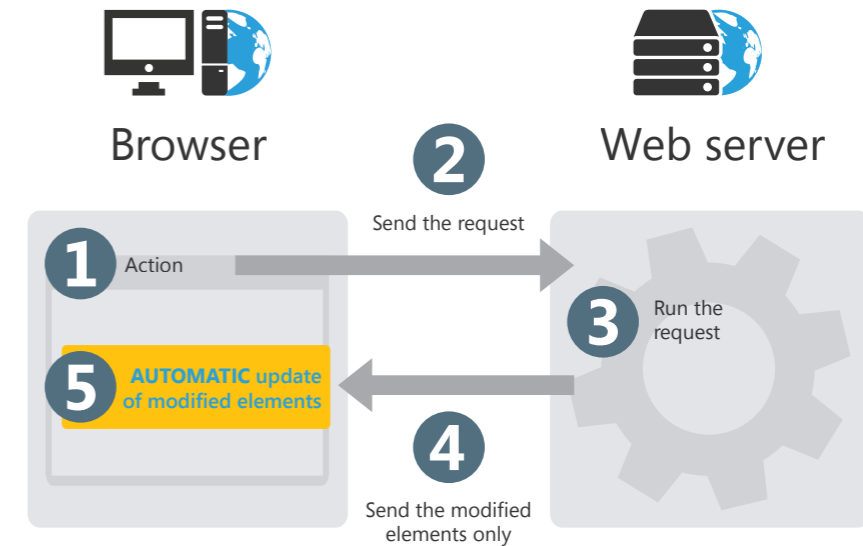
AJAX can be used at two different levels in a WEBDEV site:

- Automatic AJAX: a simple click allows you to access the AJAX features. The code remains the same.
- AJAX programming: AJAX-specific functions allow you to write complex processes.

**Remark:** Only sufficiently recent browsers support AJAX technology. **AJAXAvailable** is used to determine if the current browser supports AJAX technology. If a process that is using 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 AJAX

The diagram below presents how AJAX can be used automatically in a WEBDEV site:

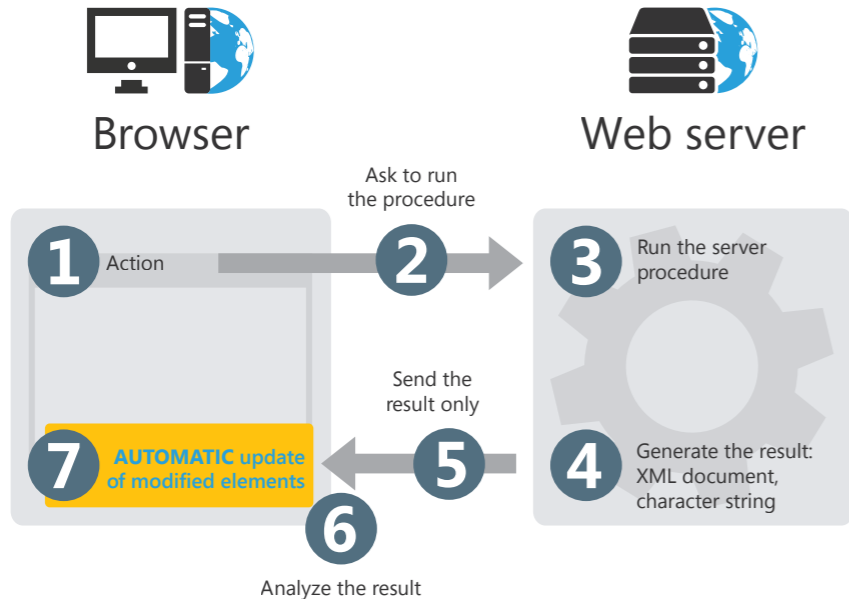


For example, a Web page contains the different characteristics of a country (capital, currency, flag, location, etc.). The corresponding information is displayed according to the country selected by the user.

- 1** Internet user's action. In our example, the Web user selects the country in the "Select a country" combo box.
- 2** The request is sent to the server.
- 3** The request is executed: the characteristics of the selected country are searched for.
- 4** The result of the request is sent:
  - without AJAX: the entire page is returned.
  - with AJAX: the characteristics of the selected country are returned.
- 5** The characteristics of the country are displayed:
  - without AJAX: the entire page is redisplayed.
  - with AJAX: the controls containing the country characteristics are refreshed.

### AJAX programming

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



- 1 Execution of a browser process (**AJAXExecute** or **AJAXExecuteAsynchronous**).
- 2 Request to execute a server procedure.
- 3 The server procedure is executed.
- 4 The result is generated. The result of the procedure will be expressed as a character string or an XML document.
- 5 The result of the procedure is sent (**RESULT**).
- 6 The result of the procedure is analyzed.
- 7 The modified information is displayed. Only the necessary controls are refreshed.

# If you are already familiar with WINDEV

Let's see the main differences between WEBDEV and WINDEV:

- WEBDEV is used to create pages while WINDEV is used to create windows.
- WEBDEV accepts different types of code: a WLanguage code run on the server, a WLanguage code run on the browser and a JavaScript code run on the browser. When a PHP page is created, an additional code appears: 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 a process/event meant to be executed in server code.
- Some WLanguage functions can only be used in a process/event meant to be executed in browser code.
- New types of controls specifically designed for Web applications are available:
  - Formatted display control,
  - Java Applet control,
  - Sliding Banner control,
  - Navigation Bar control,
  - Captcha control,
  - Cell control,
  - Breadcrumb control,
  - Web Component control,
  - IFrame control,
  - Link control,
  - Line control,
  - Internal Page control,
  - Peeling Corner control,
  - Site Map control,
  - Popup control,
  - Pager control,
  - Social Network control,
  - SilverLight control,
  - HTML Table control,
  - Upload control,
  - Thumbnail control,
  - Rich Text Area control.
- Some types of controls that are not used on Internet are no longer available: Scrollbar, ActiveX, OLE Object, Spin, etc.

# Hardware and software requirements

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

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

Server	Development computer
<ul style="list-style-type: none"> <li>• <b>1 server:</b> PC</li> </ul>	<ul style="list-style-type: none"> <li>• <b>1 computer:</b> PC</li> </ul>
<ul style="list-style-type: none"> <li>• <b>1 server operating system</b> (Windows or Linux): <b>Windows</b> 2008 or later.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>1 operating system:</b> Windows Vista or later.</li> </ul>
<ul style="list-style-type: none"> <li>• <b>1 Web server software:</b> IIS, Apache, ...</li> </ul>	<ul style="list-style-type: none"> <li>• <b>1 Web server software:</b> IIS, Apache, ...</li> </ul>
<ul style="list-style-type: none"> <li>• <b>1 WEBDEV Application Server</b></li> </ul>	
or	
<ul style="list-style-type: none"> <li>• <b>1 PHP engine</b> (version 4.3.2 or later)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>1 WEBDEV Development version</b></li> </ul>
<ul style="list-style-type: none"> <li>• <b>the site itself</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>the site being developed</b></li> </ul>
<ul style="list-style-type: none"> <li>• <b>the data*</b> (dynamic site only)</li> </ul>	<ul style="list-style-type: none"> <li>• <b>the data</b> (optional)*</li> </ul>
	<ul style="list-style-type: none"> <li>• <b>at least one browser:</b> Internet Explorer, FireFox, Chrome, ...</li> </ul>

\* The data can be on another computer connected to the network.


 PART 2

 Developing  
a site

28

DEVELOP 10 TIMES FASTER

# Project and analysis

The development of a **website** with WEBDEV is based on two main elements: the Project and the Analysis.

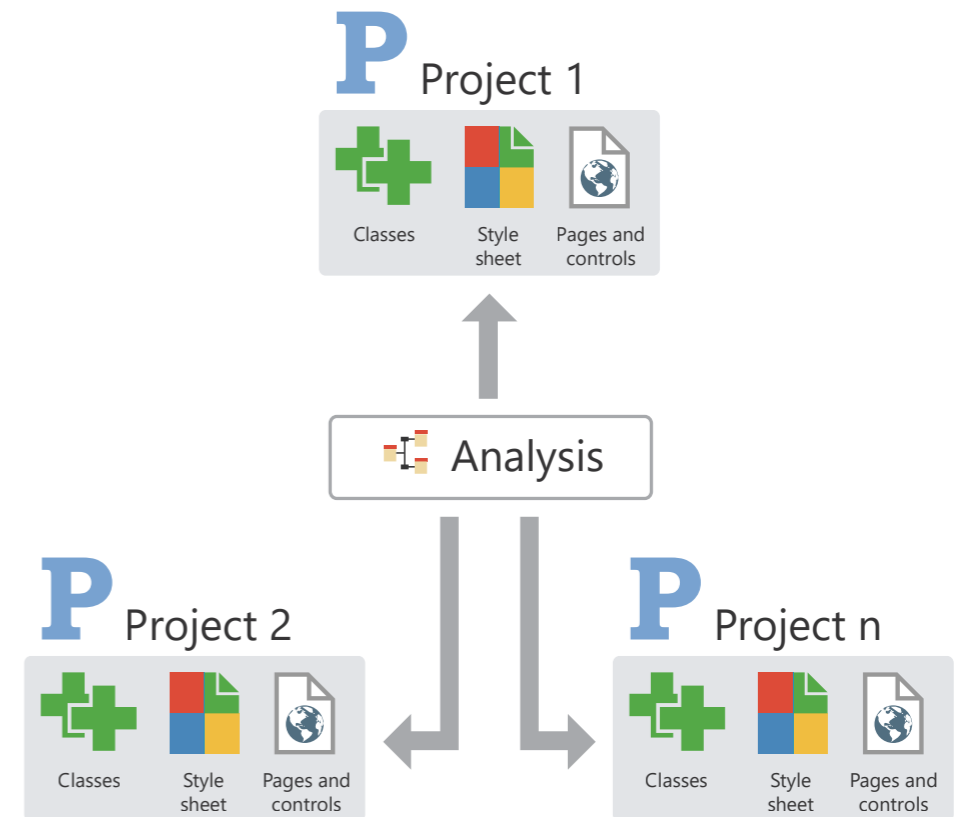
A **WEBDEV Project** is a set of elements (pages, reports, controls, classes, components, etc.), which put together, make it possible to create a website.

A **WEBDEV Analysis** is the database schema of the application. The analysis contains the different data files (tables) used by 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.





# Projects in practice


## 1 Overview

The first step in describing a site is to create a project. When creating a project, you will be asked multiple

questions so that your project meets all your expectations. All project characteristics specified when creating the project can be modified later.

## 2 Creating the project

### To create a project:

Click  in the quick access buttons in the WEBDEV ribbon. The creation window appears: click "Project". The project creation wizard starts.

1. Specify the different project options. The main options are as follows:

- **the type of project.** WEBDEV allows you to create sites, Webservices, components, etc.
- **the creation mode of the project.** You can create a blank project or a project based on an existing example (Application RAD).
- **the type of project generation.** This option allows you to define the type of site generated by the project: static site, dynamic WEBDEV site (session mode or AWP), dynamic PHP site, etc.
- **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.

- **whether the project will be used by several developers.** You have the ability to use the source code manager (SCM) to share the project.

- **the code style.** This code style is used to automatically prefix variables, names of controls, names of pages, etc.

- **the style book.**
- **the supported languages.** These languages will be proposed by default if an option can be translated in a control, window, report, etc.

2. Specify **whether the project is associated with a database.** If yes, you can use an existing database, or create a new one.

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

### Remarks:

- If you chose to create a database, the analysis creation wizard will start automatically.
- The specified information can be modified in the project description. To open the project description window, go to the "Project" tab, "Project" group and click "Description".

## 3 Dashboard and project elements

### 3.1 Dashboard

All project managers seek to have a global and synthetic vision of the progress status of the projects they lead.

The QA manager wants to know the number of bugs and their priority, and follow up their status.

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

The developer wants to quickly open the most frequently used project elements, sections of code, etc.

The dashboard meets all these requirements. Additionally, it contains different indicators or "Widgets". These widgets can be configured by the user: the user can add, delete or resize them.

You can also display the project statistics (e.g., the number of pages), the results of different audits (static audit, dynamic audit, etc.).

### To display the dashboard of your project:

1. Open the project context menu by right-clicking the "P" button next to the open document tabs.
2. Select "Display dashboard".

For more details, see "Project dashboard", page 89.

## 3.2 Project elements

The project includes pages, reports, queries, sets of procedures and more.

To know the list of elements of a project, on the "Project" tab, in the "Project" group, click "List of elements".

This option is used to:

- add to your project elements belonging to projects that can be accessed from your computer. The corresponding files will not be moved to 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

Let's see the main operations that can be performed on a project:



- Save a project,
- Restore a project,

- Duplicate a project,
- Copy or delete a project,
- Rename a project.

For more details, see the online help.

## 5 Technical documentation of the project

WEBDEV allows you to print different technical documentations describing all the elements of the project (pages, reports, data files, items, etc.). To print the technical documentation:

- on the "Home" tab, in the "General" group, expand  and select "Print project technical documentation". The technical documentation can contain all the characteristics of the project.
- on the "Home" tab, in the "General" group, click . In this case, the technical documentation will contain only the details of the current element (page, report, query, etc.).

### 5.1 Type of technical documentation

There are different types of technical documentation:

- Overall: Contains basic information about the project elements.
- Code: Contains the code of all the project objects and procedures.
- Full documentation: Contains all project information. This type of technical documentation can be very extensive.
- Custom: Corresponds to a selection of information to print.

### Generating the technical documentation

The technical documentation can be printed on the selected printer. The technical documentation is generated and displayed in the report viewer. You can:

- start the print job,

- export the technical documentation as PDF (via the "Export" tab of the report viewer).

For more details, see the online help.

### 5.2 Print areas

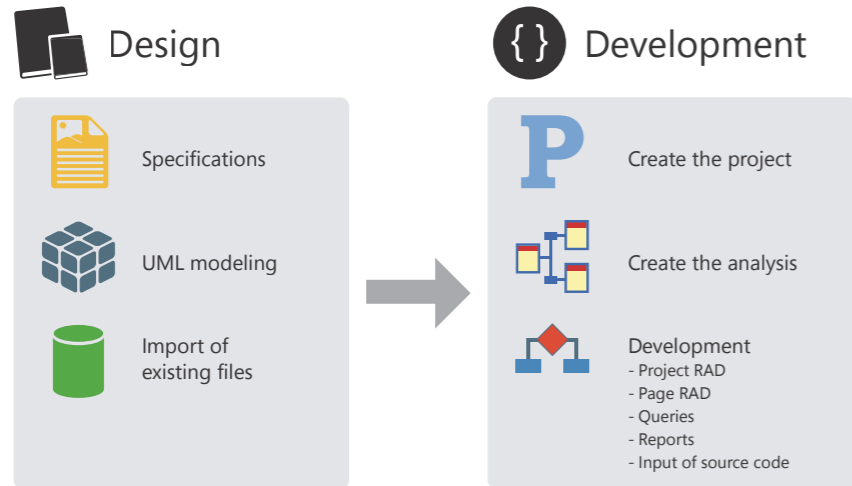
If graphical representations (LDM, UML diagrams, etc.) must be printed, the print areas must be set **before** printing the technical documentation.

### To specify the print areas of the current graphic representation:

1. Display the print areas in the current editor. For example, in the data model editor, go to the "Display" tab, "Help for edit" group, and check "Print areas".
2. A series of borders that represent the printable areas in the documentation will appear in the current editor.
3. Reduce the graphic representation (change the value in the combo box with a percentage in the "Display" pane, or use the Ctrl + Mouse wheel shortcut) to see the entire graph.
4. With the mouse:
  - move these borders to the desired 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 arrow).

# Site development life cycle

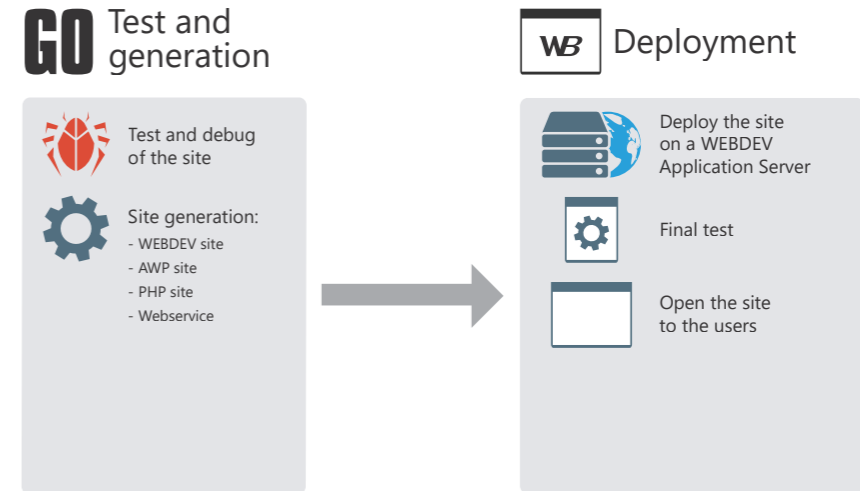
WEBDEV covers the entire site development life cycle:



**Details of the different steps:**

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

**Development phase:** The project and the analysis are created via specific wizards. The site can be developed in RAD (Rapid Application Development) mode, with automatic code and interface generation; or it can result from manually creating multiple project elements.



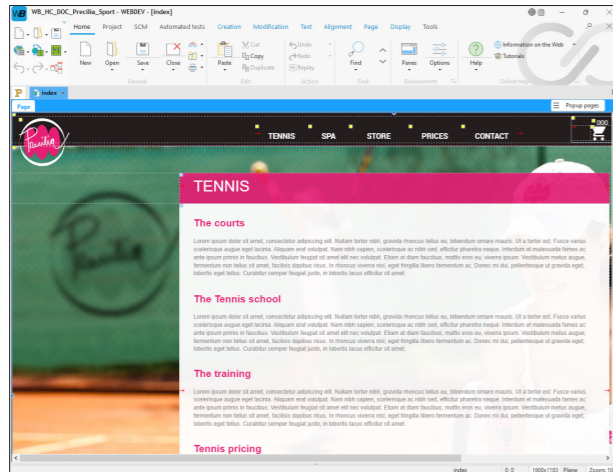
**Testing and generation phase:** WEBDEV includes several automated test tools to guarantee application reliability and make sure there are no regressions during the development steps.

**Deployment phase:** A dynamic WEBDEV site (in Session or AWP mode) is deployed on a WEBDEV Application Server. Once the site is deployed, you can test it before making it accessible to users.

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

The following steps are required to create a page with WEBDEV:

- 1 Create a page in the WEBDEV page editor.



- 2 Enter code (optional step).

```

Click on (onclick browser event) (CELL_AccountCreation1) LINK_ForgottenPass
// Check whether a login is entered
IF EDT_Existing_EmailAddress == "" THEN
  ErrorExistingAccount("Enter your email")
  ReturnToCapture(EDT_Existing_EmailAddress)
ENDIF
// Check whether the specified login exists
IF Val(AJAXExecute(bCheckConnection, EDT_Existing_EmailAddress, "")) = csCHECKCONNECTION
  ErrorExistingAccount("Identifier not found")
  ReturnToCapture(EDT_Existing_EmailAddress)
ENDIF
    
```

Browser code

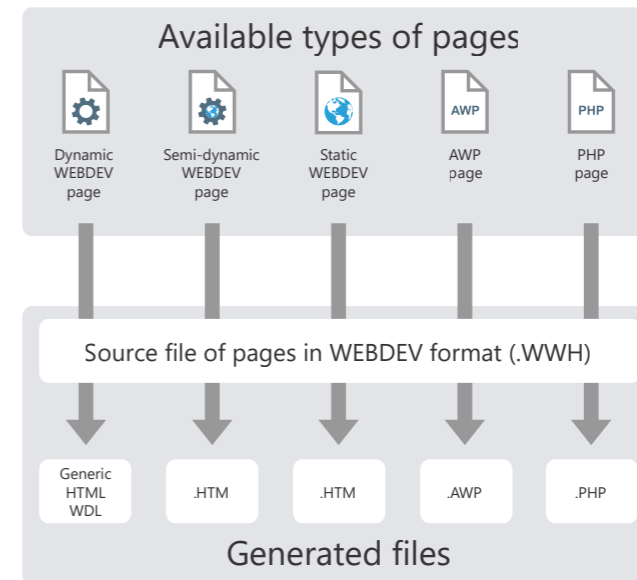
```

Click on (CELL_AccountCreation1) (server) BTN_ConnectYourself AJAX
// Read the customer record
HReadFirst(Customer, Email, EDT_Existing_EmailAddress)
IF HFound(Customer) THEN
  // Client connection
  Connection(Customer.CustomerID, Customer.Email)
  // Displays the next page (or the "customer_account" page by default)
  PageDisplay(getAccessPathNextPage("customer_account"))
ELSE
    
```

Server code

- 3 Save the page.

# The following files are automatically created



Legend:

**.WWH: Source format of pages**

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

**Generic .HTML**

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

**.HTM: HTML page of site**

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

**.WDL: Project library**

The library, which is generated when the WEBDEV site is deployed (in Session or AWP mode), contains the server code of the project pages. This file will be on the server.

**.AWP: AWP page**

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

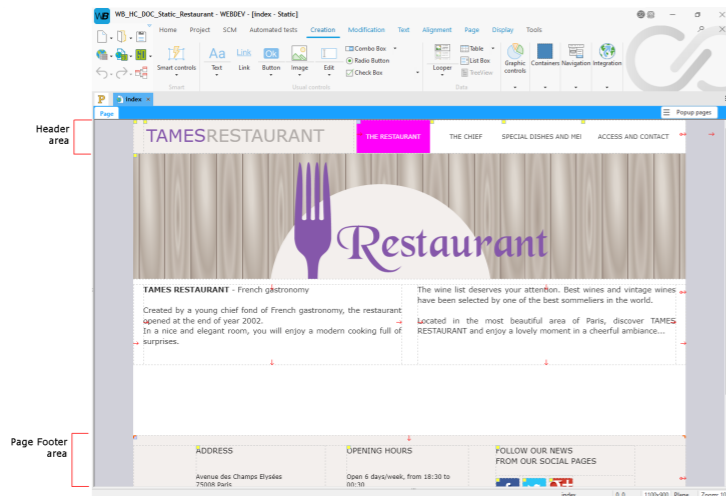
**.PHP: PHP page**

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

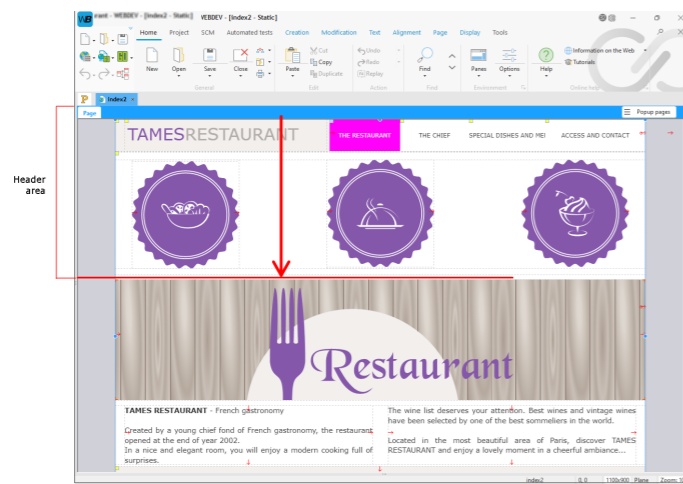
# Editing a page: zoning mode

There are many options to easily edit pages in zoning mode: you just need to split the page into different "areas" using the "Pencil" tool. These areas are automatically configured and allow you to define the page architecture: header, footer, etc.

These areas can be easily handled in the editor: when you increase the header height, the page body adapts automatically.



When the page is edited, the position and anchors of the controls in the different areas do not change.



You can associate HTML 5 text-level semantics to each area: this improves SEO in some cases.

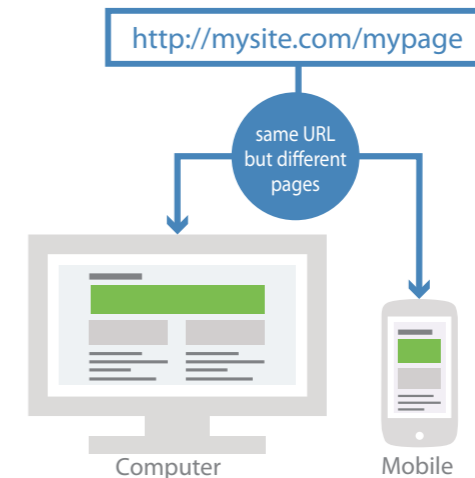
# "Mobile friendly" site: Dynamic serving

"Dynamic serving" is a technique where the same address (URL) leads to 2 different pages: a page for PC and a page for mobile.

The page will be displayed based on the device (PC or mobile) that opens it.

It can be used as a complement or alternative to "Responsive Web Design" to make a standard Web site "mobile friendly". With "Dynamic serving", you no longer need to modify the existing pages: simply add mobile-specific pages.

This technique improves SEO (Google).



## Principle

The site includes 2 sets of pages: PC-specific pages and mobile-specific pages. These pages are associated in groups of 2.

When the page address is typed in the browser, the environment used is automatically detected and the corresponding page (computer or mobile device) is displayed in the browser. The address displayed in the browser does not change: only the content displayed is adapted to the system.

All the links and pages open in the code are automatically redirected to the corresponding page in the current platform.

Remarks:

- The environment is detected based on the Google recommendations: tablets are not considered as mobile devices.
- Google detects that the pages are in "Dynamic serving" mode. Therefore, the different page versions are optimized. If the mobile pages are developed properly, the site is considered as "Mobile friendly".

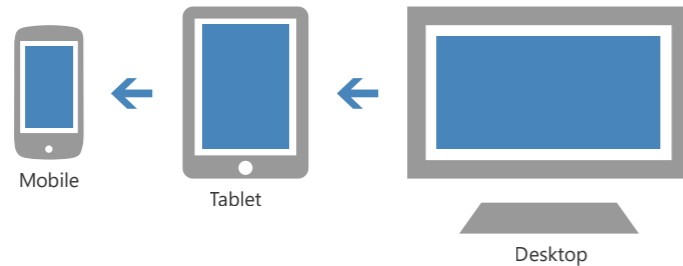
# "Mobile friendly" site: Responsive Web Design

Websites can be viewed on several platforms: mobiles, tablets, PCs, ... Depending on the device, the size of the browser and the space used to display data can be different.

The "Responsive Web Design" method consists in creating a single page that automatically adapts according to the platform used.

## Method used

WEBDEV uses the "Desktop first" method: developers must create the interface of their Website in "Desktop" mode. Then, they can create the version in "Tablet" mode, and finally, in "Mobile" mode.

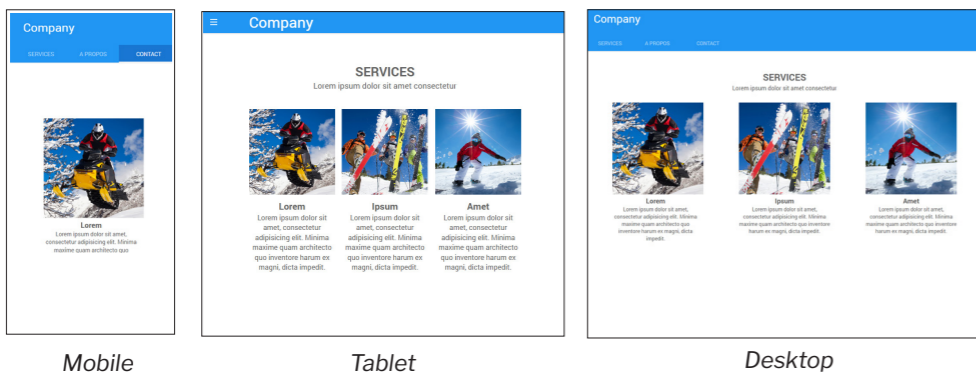


In terms of design, a website is developed for "desktop" use first. When reducing the display resolutions (page size), the content and the features are moved, resized or hidden.

In the WEBDEV editor, the different resolutions are represented by slices. A slice represents a browser size for a specific device. WEBDEV includes 3 slices:

- The Mobile slice, which corresponds to smartphones.
- The Tablet slice, which corresponds to tablets.
- The Desktop slice, which corresponds to PCs, MACs, etc.

## Example

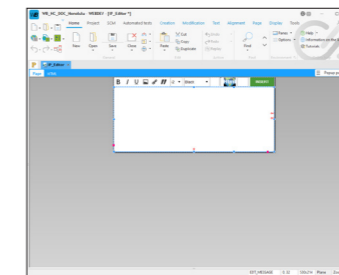


# Internal page

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

## 1. Creating an internal page

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

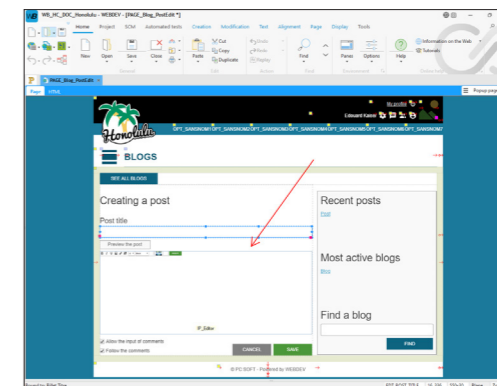


In the example, an internal page is used to manage a shopping cart.

## 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.



### Remarks:

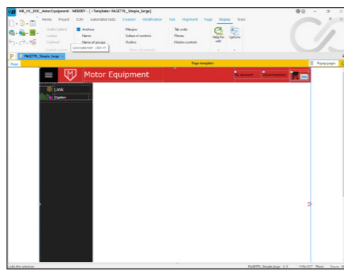
- The internal page used in the Internal Page control cannot be modified programmatically.
- The host area is rectangular and no override is allowed. To perform overrides, it is recommended to use control templates.

# Page templates

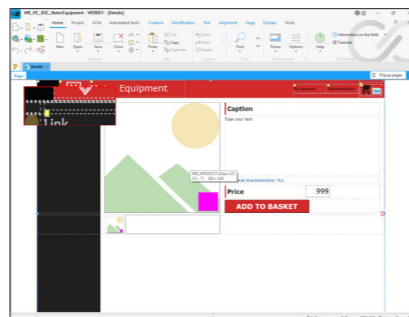
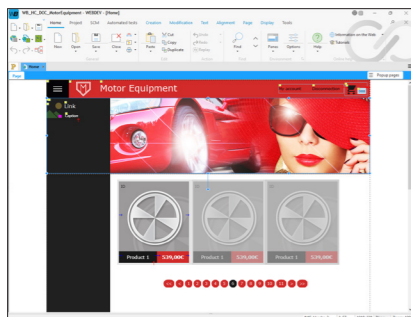
WEBDEV allows you to create page templates. A template contains elements common to a set of site pages.

The changes made to the page template are automatically applied to all the pages that use this template.

A page template allows you to use the style book defined for a site.



View of a page template.  
The template is displayed in an orange frame in the editor.



Template used in multiple pages.  
The elements that belong 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.

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

The characteristics of the elements can be dissociated from the template. For example, dissociate the position of a control from the template to move it elsewhere while keeping other changes (code, style, etc.). This is referred to as **inheritance**. In this case, the elements are identified by a blue square.

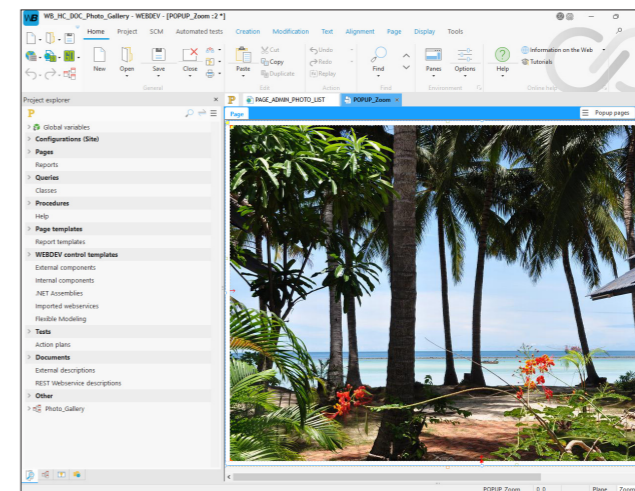
# Popup page

WEBDEV allows you to create Popup pages. Popup pages allow to interact with the user in a simple way.

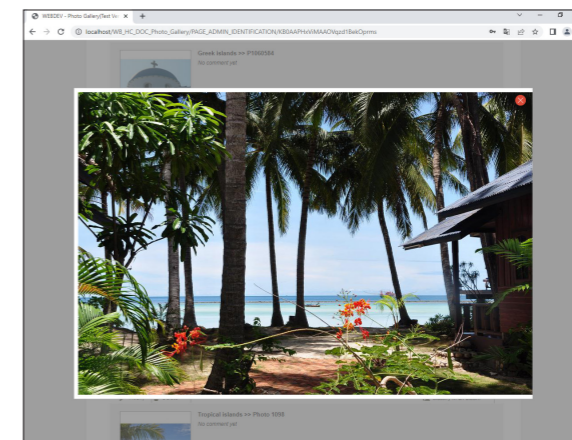
A Popup page is associated with a page. A page can have several Popup pages.

Using Popups allows you to handle the dialog window in the editor like a different element of the page, while being included in the page.

In the editor:



At runtime:



## 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, etc.

WEBDEV offers several options for creating the project pages:

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

Regardless of the method used, the page can be modified once it has been created: you can add, modify or delete controls and modify the page characteristics.

### 1 Creating a page


WEBDEV offers several methods for creating a page:

- Creating blank pages.
- Creating preset pages.
- Creating pages via the project RAD.

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

#### 1.1 Creating blank pages

To create a blank page:

1. Click  in the quick access buttons of WEBDEV. The new element window appears: click "Page" then "Page". The page creation wizard starts.
2. Choose the "blank" type.
3. Validate the creation of the page. A blank page is created.
4. The save window is automatically displayed.
5. Specify the page title and name. The page corresponds to a ".WWW" file. By default, this file will be created in the main project directory. This name will be used to handle the page.
6. Validate the save window.
7. Create the controls in the page.

For more details on how to manipulate the controls on a page, see "Controls in practice", page 54.

This chapter presents the following topics:


- "Creating a page", page 44.
- "Simple operations on a page", page 45.
- "Main characteristics of a page", page 45.
- "Internal page", page 46.
- "Page templates", page 46.
- "Popup page", page 46.
- "Refreshing a Session page", page 46.
- "Opening a page in a WEBDEV site", page 47.
- "Events associated with pages", page 48.
- "Menus", page 48.

Remarks:

- By default, the new page is of the same type as the current site (static, Session, AWP or PHP). The type of page can be changed in the page description window. For more details, see "Main characteristics of a page", page 45.
- By default, the new page uses the "Responsive (zoning and layouts)" edit mode. This edit mode allows you to use layouts to obtain a mobile-friendly site. The edit mode of the page can be changed in the page description window.
- The different types of controls are presented in "Different types of standard controls", page 50.

#### 1.2 Creating preset pages

To create a preset page:

1. Click  in the quick access buttons of WEBDEV. The new element window appears: click "Page" then "Page". The page creation wizard starts.
2. In the tabs, choose the type of page to create:
  - the "Standard" tab allows you to create:
    - a blank page without any controls.
    - a blank page based on a page template, if the project contains a page template.
    - a page based on a predefined template.
  - the "Internal pages" tab allows you to create internal pages.

- the "Error pages" tab allows you to create pages to customize the errors displayed on the site.
  - the "RAD" tab is used to create RAD pages that allow data display and input. This type of page can be based on a data file, query, etc.
3. Depending on the selected type of page, enter the requested information in the different steps of the wizard.
  4. Validate the page creation.
  5. The save window appears automatically. Specify the page title and name, then validate.
  6. The new page is displayed in the editor. If necessary, the page template associated with the page is also generated and saved in the project.

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

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

For more details, see the online help.

To create the dynamic site by RAD:

1. On the "Project" tab, in the "Generation" group, click "Full Application RAD".  
**Caution:** To use this option, the project must be associated with an analysis that has been 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. Validate the creation of your site via RAD.

### 2 Simple operations on a page

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

- Open a page in the editor: Simply double-click the page name displayed in the "Project explorer" pane.

- Save and copy a page.
- Import and export a page.
- Change the tab order.

For more details, see the online help.

### 3 Main characteristics of a page

A page can accept all types of controls. A page can:

- include a **background image**: this image is displayed in the background and the controls are overlaid on the image.
- include a **menu**: this menu allows Internet users to quickly access the main features of the site. To create a drop-down menu in a page, use a "Menu" control. The first two options are displayed in the page. The options can be manipulated via the context menu. Then, the menu options can be handled in the programming code.

- include a **status bar**: this status bar will display the help messages associated with the different page controls. For more details, see the online help.
- be **multilingual**: all the languages defined for the page will be available for all page controls. A page can be associated with more languages than the project (e.g., pages shared between multiple projects). For more details, see "Multilingual sites", page 133.

**Remark:** Web pages intended to be viewed on iPhones may benefit from special enhancements: splash screen, full screen mode, etc. These features make them look like real applications. WEBDEV allows you to easily configure these specific features while keeping the compatibility with other browsers (mobile or not) on the market.

## 4 Internal page

Internal pages are used to dynamically share a section of interface in one or more sites.

The interface to be used multiple times is created in an "Internal Page".

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

**Remark:** The internal page to be shared can come from a component.

For more details, see the online help.

## 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 changes made to the page template are automatically applied to all the pages that use this template.

A page template allows you to use the style book defined for a site.

Using page templates in your sites allows you to:

- easily create the site pages.

- define the layout of the site pages.
- easily update the site style book.

A page template can be created as easily as a page. To create a page using a template, simply choose the "based on a template" option.

By default, any change made to the template is applied to the pages in which it is used. However, you can manage special cases in a specific page by overriding the template elements.

For more details, see the online help.

## 6 Popup page

WEBDEV allows you to create popup pages. These popup pages allow you to interact with the users in a simple way. A Popup page can be used to:

- prompt users to enter information (SMTP parameters for sending emails, new address, etc.),
- ask to confirm a deletion,
- etc.

To open a popup page from another page, you can use the **PopupDisplay** browser function.

**Remark:** It is also possible to display a page as a popup. In this case, the popup will be displayed using **PopupDisplayPage**.

For more details, see the online help.

## 7 Refreshing a Session page

### 7.1 Overview

Refreshing a page consists in re-displaying a page that is already open in the browser. This action is the same as the "Refresh" button in the browser.

When a page is refreshed, the dynamic data in the page is updated according to the page context 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 cart) is refreshed each time a new item is ordered.

### 7.2 Implementation

To refresh a page, simply use the **PageRefresh** WLanguage function.

The following operations are automatically performed when a page is refreshed:

1. The existence of the page context on the server is checked.
  2. The page is displayed again according to its context on the server.
- Using **PageDisplay** to refresh a page is not recommended because it takes longer to load the page.

## 8 Page contexts

### 8.1 Overview

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

In a WEBDEV site in Session mode, a page context exists on the server for each page displayed on the browser of the Internet user.

The page context contains all the information related to the page:

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

### 8.2 Automatic operating mode

By default, WEBDEV automatically manages the contexts of pages in Session mode:

- A page context is opened when the Session page is displayed in the browser.
- The context of a page is updated according to the information the user enters in the browser. The context is updated when the user validates the page (via a "Submit" button or with **PageSubmit**).
- The existing page contexts are closed when **PageUse** is used. Then, the page context that corresponds to the page to be displayed with this function is opened.

**Remark:** WEBDEV allows you to perform an advanced management of contexts via the ContextXXX functions. For more details, see the online help.

## 9 Opening a page in a WEBDEV site

### 9.1 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, etc.),
- through programming in WLanguage.

### 9.2 Opening a page from a control description

You can set an action and the destination of this action in the description of Button, Link and Image (with clickable area) controls.

To open an "xxx" page directly from a Button, Link or Image control:

1. Open the element description ("Description" in the context menu of the element).
2. In the "Action" area, select the type of action: "Display a site page".
3. Select the page to display.
4. Select the destination of the action: current page, current browser, etc. The previously selected action will be performed in this destination.
5. Finally, indicate if the content of the controls should be sent to the server during the action.
6. Validate.

### 9.3 Opening a page through programming

Several WLanguage functions can be used to open a page.

- **PageRefresh:** Refreshes the page displayed according to its context.
- **PageDisplay:** Opens and displays a new page in the user's browser.
- **PageDisplayDialog:** Opens and displays a new page modally in the user's browser.
- **PageUse:** Closes all current pages (and their contexts) and opens a new page.
- **PopupDisplay:** Displays a popup in the page.
- **PopupDisplayPage:** Displays a page as a popup in the page.

When the page is opened, you can pass parameters to it. For more details, see the online help.

**Remark:** By default, when a page is opened through programming, the selected destination is:

- the destination set in the description of the object that opens the page (Button, Link control, etc.).
- the destination set for the current page.



## 10 Events associated with pages

### 10.1 Default events

WEBDEV manages the following events by default (order of appearance in the code editor):

- **Global declarations (server code):**  
Declaration of variables global to the page. This is the first code run when the page is opened.
- **Initialization (server code):**  
Executed when the page is opened. The initialization event of the page controls is run before this code.
- **Load (onload) page (browser code):**  
Browser code run when the page is displayed in the browser.
- **Unload (onunload) page (browser code):**  
Browser code run when a new page is displayed in the browser.
- **Whenever changing layout (server code):**  
Run when the page layout is changed.
- **Request for refreshing the display (server code):**  
Contains all the code for refreshing the page controls: content of controls, state (visible, grayed, etc.), caption to refresh, etc. This event is started upon request by the following functions:
  - **RequestRefreshUI:** Executes the event at the end of the current event (asynchronous execution).
  - **ExecuteRefreshUI:** Executes the event immediately.

- **Page synchronization (server code):**  
Server code used to manage the page synchronization when clicking the browser "Back" button.
- **Closing the page (server code):**  
Run when the page is closed.
- **Set the Value property (internal page only):**  
Executed when a value is assigned to the **Value** property in the Internal Page control.
- **Get the Value property (internal pages only):**  
Executed when the **Value** property is used in read mode in the Internal Page control.

### 10.2 Optional events

Several optional events can be managed.

To use an optional event, you must:

1. Open the page code window.
2. Click "Add other events to the page", at the bottom of the list of events. The optional events window appears.
3. Select the optional event to add and validate.
4. The selected event is automatically added after the default events.

You can manage, for example:

- the mouse hover over the control,
- left mouse button down, up, double click,
- right mouse button down, up, double click,
- the mouse wheel, etc.

## 11 Menus

WEBDEV allows you to create a menu in your HTML pages. This menu can be oriented vertically or horizontally. From the different options of this menu, you can:

- display pages,
- execute the code of the Button controls in the page,
- etc.

WEBDEV allows you to:

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

For more details, see the online help.

## 12 Menu options

### 12.1 Overview

A menu includes one or more options and sub-options. Each option runs a WLanguage code.

To handle the menu options, simply 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. Double-click the menu: a yellow border appears.

### 12.2 Handling 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 submenu,
- Delete a menu option.

The menu options can also be defined in the page editor, and their characteristics can be modified via a description window.


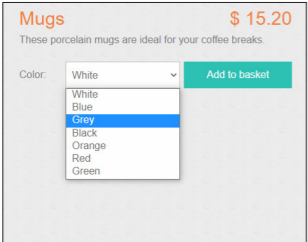
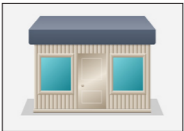
The description window of the menu options allows you to:

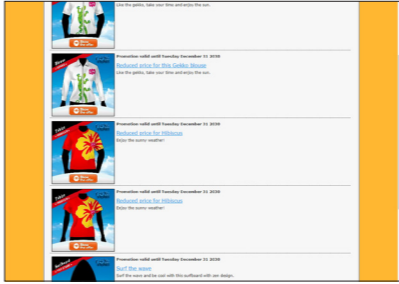
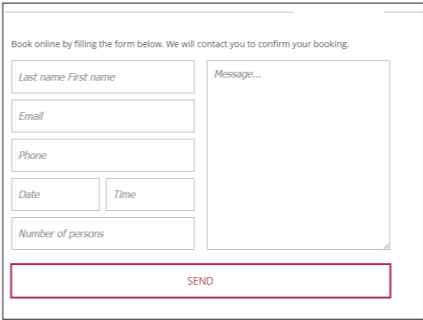
- modify the option text,
- check or uncheck a menu option,
- associate an image with a menu option,
- associate a keyboard shortcut with a menu option.

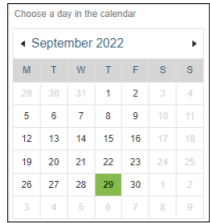
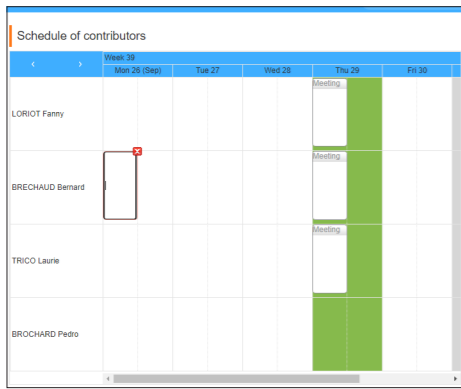
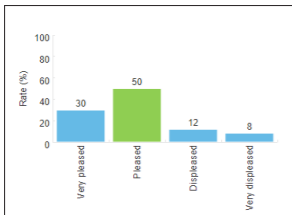
You also have the ability to associate a WLanguage code with a menu option. Only menu options without submenus can start a WLanguage process.


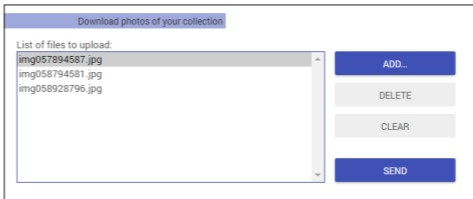

For more details, see the online help.

# Types of standard controls

In order to...	Use ...
Display a text, title, etc. 	Static control
Use text with a specific layout (images, links, etc.).	Rich Text Area control
Define a color area.	Cell control, HTML table
Display a price, quantity, address, date, time, etc.	Formatted display control
Select a value from a list (country, city, color, etc.). 	Radio Button control, Combo Box control, List Box control
Display an animation (animated Gif).	Image control
Display a page according to the area that was clicked in an image.	Image control (with clickable areas)
Use existing HTML code.	HTML control
Display a graphic image (photo, etc.). 	Image control
Use a Java applet (clock, etc.).	Java control

Select several values from a list (message recipients, files to download, etc.).	Check Box control, List Box control
Display the contents of a data file in a table (list of customers, order details, etc.).	Browsing Table control or memory Table control, Browsing Looper control or memory Looper control
Display an image refreshed on a regular basis.	Web Camera control
Display controls multiple times in a page (product catalogs with photos, etc.). 	Looper control
Enter information. 	Edit control
Display information in hierarchical order (directory content, etc.).	TreeView control
Align controls (HTML table).	Cell control, HTML Table
Program an action in a page (display a page, validate an input, etc.).	Button control, Link control
Display a video.	Video control
Display a page from another site in one of your pages.	IFrame control

<p>Select and display a date on a calendar.</p> 	<p>Calendar control</p>
<p>Display appointments in a scheduler or organizer.</p> 	<p>Scheduler control, Organizer control</p>
<p>Display a Column, Line or Pie chart.</p> 	<p>Chart control</p>
<p>Display a page from your site in one of your pages.</p>	<p>Internal page</p>
<p>Display an automatic menu (that is built while the site is browsed).</p>	<p>Breadcrumb control</p>
<p>Display a list of objects as a scrolling horizontal list.</p>	<p>Linear Looper control</p>

<p>Allow the Internet user to give or view a rate.</p> 	<p>Rating control</p>
<p>Display thumbnails.</p>	<p>Thumbnail control</p>
<p>Display a set of images in gallery format.</p>	<p>Gallery Looper control</p>
<p>Include SilverLight controls.</p>	<p>SilverLight control</p>
<p>Expand/Collapse a display area</p>	<p>Drawer control</p>
<p>Upload one or more files.</p> 	<p>Upload control</p>
<p>Display several contents successively.</p>	<p>Sliding Banner control</p>
<p>Display a dashboard with multiple information in thumbnail format.</p> 	<p>Dashboard control</p>
<p>Use external resources available on Internet (Angular JS components, Bootstrap snippets, etc.).</p>	<p>Web Component control</p>
<p>Create a visual effect in a page by presenting an image partially covered by another one.</p>	<p>Peeling Corner control</p>

# Controls in practice

WEBDEV proposes several types of controls that can be easily included in your pages. These controls can be created via the "Creation" tab of the

WEBDEV ribbon. All controls can be handled through programming.

## 1 Creating a control

The following controls are available in the WEBDEV page editor:

- Static control,
- Text Area control,
- Formatted display control,
- Button control,
- Link control,
- Image control,
- Popup control,
- Thumbnail control,
- Peeling Corner control,
- Social Network control,
- Multimedia control,
- Web Camera control,
- Edit control,
- Combo Box control,
- List Box control,
- Rating control,
- Captcha control,
- Radio Button control,
- Check Box control,
- Slider control,
- Looper control,
- Linear Looper control,
- Table control,
- Scheduler control,
- Calendar control,
- Diagram Editor control,
- Bar Code control,
- Progress Bar control,
- Cell control,
- Tab control,
- Drawer control,
- Sliding Banner control,
- Control Template control,
- Chart control,
- Site Map control,
- Image Editor control,
- Supercontrol,
- HTML Table control,
- Menu,
- Pager control,
- Internal Page control,
- Breadcrumb control,
- HTML control,
- IFrame control,
- Image gallery Looper control,
- Silverlight control,

- TreeView Table control,
- TreeView control,
- Dashboard control,
- Organizer control,
- Java Applet control,
- Line control,
- "Web component" control,
- Layout control.

### 1.1 Creating a new control

To create a control:

1. Select the type of control to be created via the corresponding icon in the "Creation" tab of the WEBDEV menu.
2. The shape of the new control appears under the cursor.
3. Click the position where the control will be created in the page. The control is automatically created.

**Remark:** Other creation modes are available. For more details, see the online help.

### 1.2 Creating a control associated with an item

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

- Button control,
- Tab control,
- Line control,
- "Web component" control,
- Java Applet control,
- HTML control
- Site Map control
- Pager control
- Cell control and HTML Table control

There are multiple methods to create a control that is bound to an item:

- Drag and drop the item from the "Analysis" pane.
- Using the "Binding" tab of the control description window.

## 2 Control characteristics

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

**Remark:** You can also view and/or change the characteristics of one or more controls in the modifier. For more details, see the online help.

### 2.1 Displaying the characteristics

To open a control description window:

- double-click the control.
- select "Description" in the context menu.
- select the control and press Alt + Enter.

**Remarks:**

- 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 opened at the same time. Each description window displays the characteristics of one or more controls.

### 2.2 Characteristics by tab

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

For more details, see the online help.

#### General tab

The "General" tab allows you to specify the name and all the display characteristics of the control (text, input mask, etc.).

#### UI tab

The "UI" tab allows you to define the parameters of the control interface:

- Initial state of the control when the page is opened,
- Control visibility,
- Control size,
- Anchoring, etc.

#### Details tab

The "Details" tab allows you to define the different parameters of the control:

- input parameters,
- Allow indirections, ...

The content of this tab depends on the type of the current control.

#### Binding tab

The "Binding" tab allows you to select the item (from a data file or query) to which the control is bound. Depending on the current record, the content of the bound item will be displayed in the control.

The binding can be established to a single file or to multiple files.

#### Content tab

The "Content" tab allows you to define:

- the initial content of the control (for Edit controls only).
- the data source used to populate the control (for List Box, Combo Box, Table controls, etc.).

#### Help tab

The "Help" tab allows you to:

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

#### Advanced tab

The "Advanced" tab is used to enter the HTML code generated before and after the control. It also allows you to enter the value added in the 'class' attribute of the control.

#### Style tab

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

- Change the control aspect: simply select the control element to be modified, and then choose the style characteristics. Only the aspect of the current control is modified.
- Create or modify a style.
- Choose a style.

## 3 Handling controls in the editor

### 3.1 Control handles

Handles appear around new or selected controls. Handles allow you to:

- view the control size,
- change the control size.

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

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

## 4 Aligning controls

Control alignment allows you to create "professional" and cohesive UIs. You can create standard interfaces using the following WEBDEV tools:

- rulers.
- grids.
- real-time interface checker (automatically proposed when positioning a control).
- advanced interface checker.
- alignment options.
- customizable zoom.

### 4.1 Rulers

The alignment rulers can be displayed in the page editor.

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

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

### 3.2 Available operations

The page editor is used to:

- Select one or more controls (with the lasso for example).
- Group selected controls.
- Move a control.
- Change the control caption.
- Display an advanced tooltip when hovering a control. This tooltip contains: the control name, position, size and initial state (if the control is invisible).

### Displaying the rulers

To display the rulers, go to the "Display" tab, "Help for edit" group and check "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.

**Remark:** To show or hide the page guides, select "Show guides" in the context menu of the rulers. Guides are always enabled even if they are not visible: any control moved toward a marker is automatically snapped by it.

### Handling markers

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

## 4.2 Snap-on grid

The snap-on grid is used to place 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, 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 controls in the different site pages.

The advanced interface checker proposes to apply, to the current page or to the selected controls, the interface rules from the Windows standard: alignment of controls, standardization of button sizes, etc.

### 4.5 Alignment options

To align multiple controls, WEBDEV allows you to define:

- a preset alignment (available on the "Alignment" or "Modification" tabs, in the "Depth and Alignment" group).
- a custom alignment (available on the "Alignment" tab, in the "Other alignments" group). The custom alignment allows you to use specific alignment properties.

In the custom alignment, you can define:

- **the horizontal or vertical alignment:** controls can be aligned to the left from the beginning of the control or from the input area. They can also be aligned to the right according to the end of the control or input area.
- **the even spacing:** the spacing between controls is adapted in order to be the same between each control (horizontally or vertically).
- **the size:** the size of the selected controls automatically adapts to obtain controls with the same height and/or width.
- **the reference control:** to align and define the size of the controls, the reference control can be:
  - the first selected control,
  - the last selected control,
  - the largest selected control,
  - the most top left selected control.

### 4.6 Customizable zoom

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

To specify a precise zoom factor, you can:

- on the "Display" tab, in the "Help for edit" group, specified the desired zoom factor.
- enter the zoom factor in the status bar of the editor (bottom right).
- press the Ctrl key and change the zoom factor with the mouse wheel while holding down the Ctrl key.

## 5 Options for editing controls in a page

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

The edit options allow you to set:

- the display options.
- the snap-to behavior of the controls.
- the clicks and double-clicks on the controls.

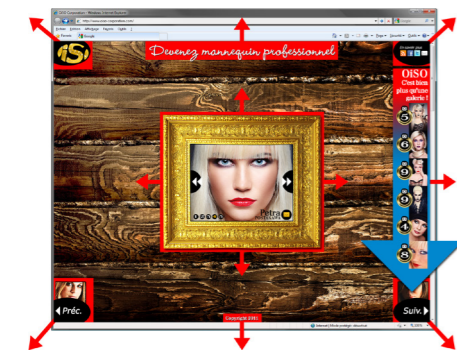
- the control selection options.

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

To open these options, go to the "Display" tab and click the dialog box launcher (☰) in the "Options" group.

## 6 Anchoring controls

The pages of a WEBDEV site can be resized at runtime: the browser can be resized, the site can be displayed on different platforms (iPhone, PC, etc.). The anchoring mechanism allows controls to automatically adapt their size and position when the page is resized.



The anchoring of a control is defined via several parameters:

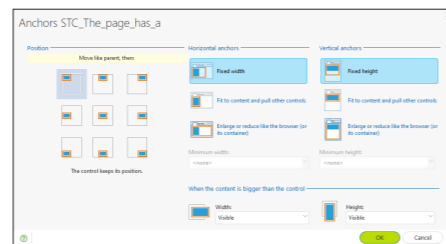
- **the control anchors:** this parameter defines the modification that will be applied to the control according to the change of the page size. The control can move left and/or down and can be enlarged vertically and/or horizontally.
- **the control width and/or height:** this parameter is used to define the behavior of the control when resized. The height or the width can be adapted to the control content or to the browser. You can also define the minimum control size.

These parameters can be configured in the page editor.

### 6.1 Anchoring a control in the editor

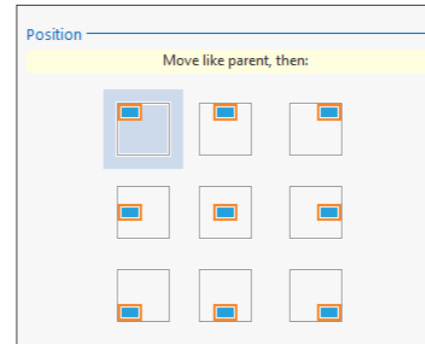
To anchor a control:

1. Select the controls you want to anchor.
2. Define the control anchoring options:
  - in the "UI" tab of the control description window ("Description" in the context menu).
  - via "Anchor" in the context menu of the control (right click).
3. Choose the desired type of anchoring in the window that appears.

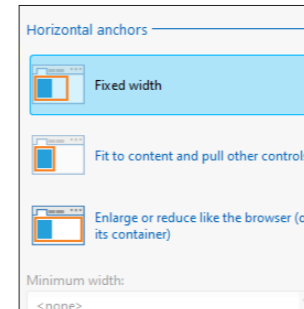


Three types of anchors can be defined:

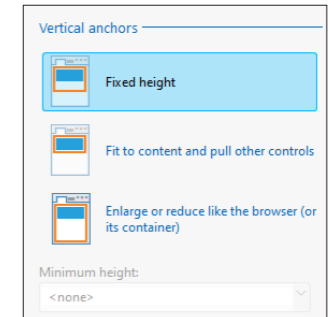
- position anchors: allows you to define the position of the control when the browser is resized.



- horizontal anchors: allows you to define how the control will react (horizontally) when the browser is resized. The control can:
  - keep the same width,
  - adapt to the content,
  - change along with the browser.



- vertical anchors: allows you to define how the control will react (vertically) when the browser is resized. The control can:
  - keep the same height,
  - adapt to the content,
  - change along with the browser.



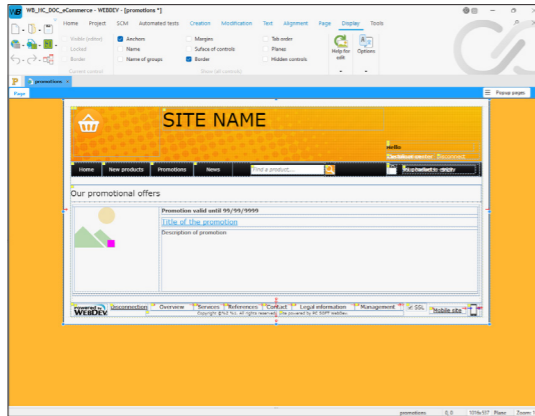
4. It is also possible to define the behavior of the control when its content overflows. You can:
  - enlarge the control and push the other controls in the page,
  - truncate the content,
  - always enable a scrollbar,
  - display a scrollbar only when the content is larger than the control.

5. Validate. If the "Anchor" option is enabled (on the "Display" tab, in the "Show (all controls)" group), the anchor signs are automatically displayed in the control (red arrows).

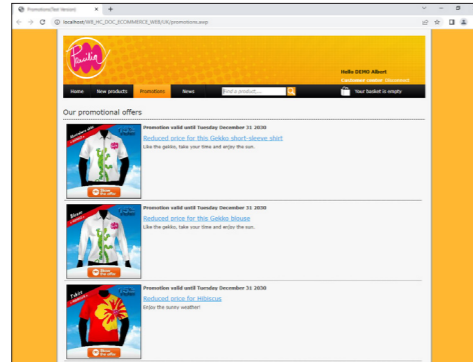
Remark: To define the anchoring of the controls, you can also use positioning tables. For more details, see the online help.

# Repeating a group of controls: loopers

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



Looper control in the WEBDEV editor.



Looper control in the browser.

The characteristics of the controls can be modified in each row of the Looper control: caption, text color, etc.

Controls can be repeated:

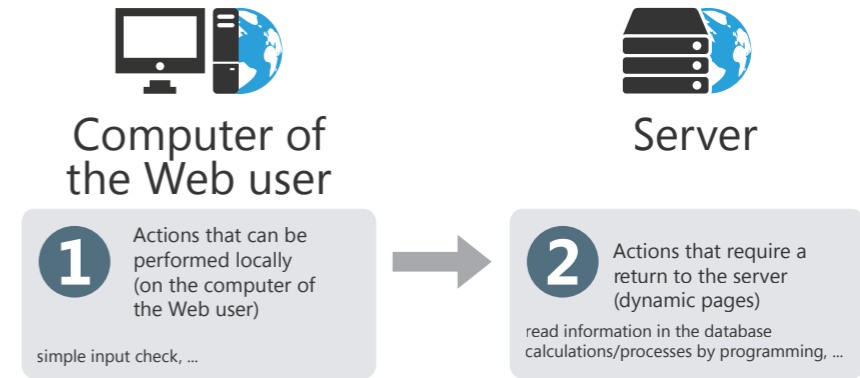
- on a column,
- on several columns. In this case, it is referred to as "Horizontal Looper".

When creating a Looper control, you can link it directly to a data file or query. The attributes of the Looper control are automatically defined. No programming is required.

# Types of code

## Server code or browser code?

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

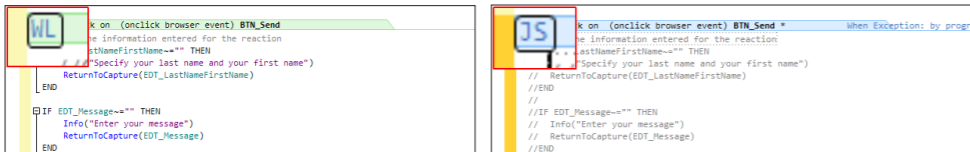


To manage these two types of actions, the WEBDEV code editor differentiates two types of code:

- **Server code** (yellow or pink code in the code editor): This code is written in WLanguage (yellow code) or in PHP (pink code, only available in PHP pages). This code is run on the server. This code is only available in dynamic pages (in Session or AWP mode).
- **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 converted to JavaScript and included in the WEBDEV HTML pages. This code is run locally (on the computer of the user) and requires no server action.

### Browser code: WLanguage code or JavaScript code?

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



- Green bar = WLanguage: the WL symbol is displayed on the code header.
- Blue bar = JavaScript: the JS symbol is displayed on the code header.

To switch from green code (WLanguage) to blue code (JavaScript), simply click the WL symbol on the code header (and conversely by clicking JS) and choose the desired language. When the page is saved, the WLanguage code is automatically converted to JavaScript.

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

### Additional events

The link "Add other events" in the code window is used to manage events in addition to those proposed by default. Simply click on "Add other events to xxx" at the bottom of the code window: the list of additional events is displayed.

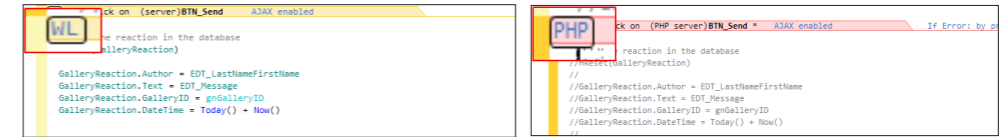
Simply select the desired events to add them to the codes displayed by the code editor.

Example of additional events: double-click, key down, key pressed, key released, button pressed, mouse moved, etc.

**Reminder:** Browser code is available in dynamic (Session or AWP mode) and static pages.

### Dynamic site in PHP: Server code: WLanguage code or PHP code?

The code run on the server is represented by a Yellow or Pink bar in the code editor.



- Yellow bar = WLanguage: the WL symbol is displayed on the code header.
- Pink bar = PHP: the PHP symbol is displayed on the code header.

To switch from yellow code (WLanguage) to pink code (PHP), simply click the WL symbol on the code header (and conversely by clicking PHP) and choose the desired language. When the PHP page is saved, the WLanguage code is automatically converted to PHP.

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



# WLanguage: a simple and powerful language

WLanguage is the programming language used in WEBDEV. This language allows you to easily program all the desired web processes without being fluent in HTML, JavaScript or PHP.

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

Let's see an example that illustrates the power of WLanguage: checking the input of an address:

In WLanguage, only a few lines of code are needed:

```
IF NoSpace (ADDRESS) = "" THEN
    Error ("Specify your address")
    SetFocusAndReturnToUserInput (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 _VALIDORD_BUTTON_OK_CLI()
{
if ((EXE_NoSpace (ADDRESS, 3)==""))
{
alert("Specify your address");
return;
}
}
}</SCRIPT>
```

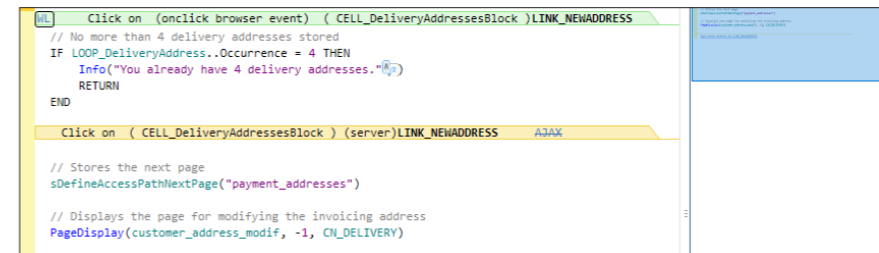
WLanguage code is easier to write, understand and maintain. It is much more reliable. Changes are easier to make.

Use WLanguage in all your processes.

# Order of execution of the code of buttons/links

Two click codes are available for buttons/links:

- a **browser click code**, which can be written in WLanguage (or JavaScript) in the code editor. This code is included in the HTML page and is executed on the user's computer.
- a **server click code**, which can be written in WLanguage (only in dynamic pages). This code is executed on the server.



The table below presents the different codes run (according to the type of button/link) and the associated action.

The appearance and target of the button/link do not affect the order in which the codes are executed.

Tip: Think as the user would to understand and remember the order in which the server and browser codes are executed: the user clicks the button in the browser, therefore the browser click must be executed first.

Action	Sequence of codes run	Remarks
Send values to the server (submit)	<p><b>Associated action:</b> Run browser codes only Run server and browser codes</p> <p><b>Code run:</b></p> <ol style="list-style-type: none"> <li>1. Browser click code of control.</li> <li>2. Browser code of page submit (if it exists).</li> <li>3. Sending the values found in the page controls to update the page context on the server.</li> <li>4. Server code of control.</li> </ol>	Page context automatically updated on the server (Session or AWP dynamic pages only).
	<p><b>Associated action:</b> Display a site page</p> <p><b>Code run:</b></p> <ol style="list-style-type: none"> <li>1. Browser click code of control.</li> <li>2. Browser code of page submit (if it exists).</li> <li>3. Sending the values found in the page controls to update the page context on the server.</li> <li>4. Server code of control.</li> <li>5. Page display.</li> </ol>	Page context automatically updated on the server (Session or AWP dynamic pages only). Automatic page display.
Reinitializing pages (reset)	<p><b>Associated action:</b> Run browser codes only</p> <p><b>Code run:</b></p> <ol style="list-style-type: none"> <li>1. Browser click code of control.</li> <li>2. Displaying the initial page (page with all controls initialized to empty or to 0 and execution of initialization code for each control then for the page).</li> </ol>	
Do not send anything to the server	<p><b>Associated action:</b> Run browser codes only</p> <p><b>Code run:</b> Browser code of control only.</p>	
	<p><b>Associated action:</b> Run server and browser codes</p> <p><b>Code run:</b></p> <ol style="list-style-type: none"> <li>1. Browser click code of control.</li> <li>2. Server code of control.</li> </ol>	
	<p><b>Associated action:</b> Display a site page</p> <p><b>Code run:</b></p> <ol style="list-style-type: none"> <li>1. Browser click code of control.</li> <li>2. Server code of control.</li> <li>3. Page display.</li> </ol>	Automatic page display

# Upload: Sending files to the server

Uploading consists in sending one or more files from the user's computer to the server.

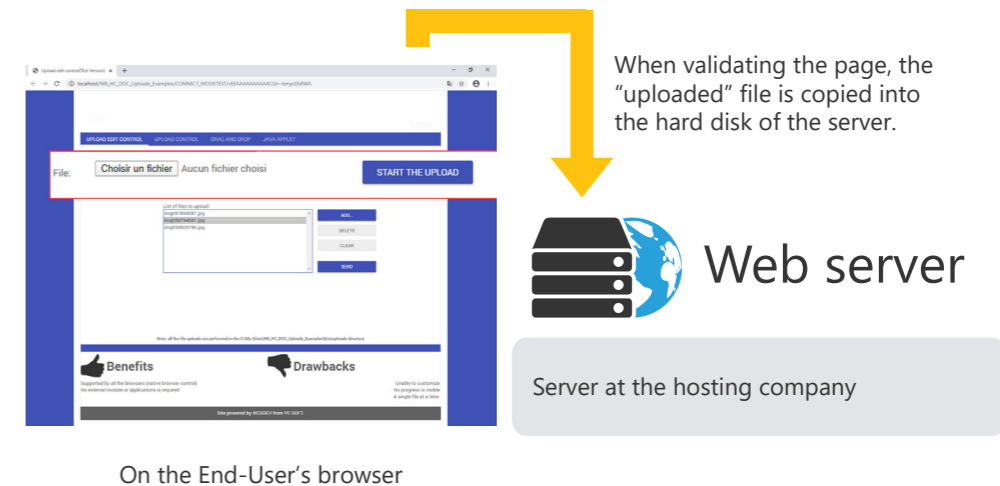
**A simple example: Create a contact in a directory.**

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

The UPLOAD control allows the user to select the image via a "Select a file" button.

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

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



On the End-User's browser

**Caution:** The appearance of the "Select a file" button (shape, color or text) is defined by the browser and cannot usually be customized.

**Remark:** WEBDEV also includes an upload Smart control. This control supports progress bars, multiple file selection, etc.

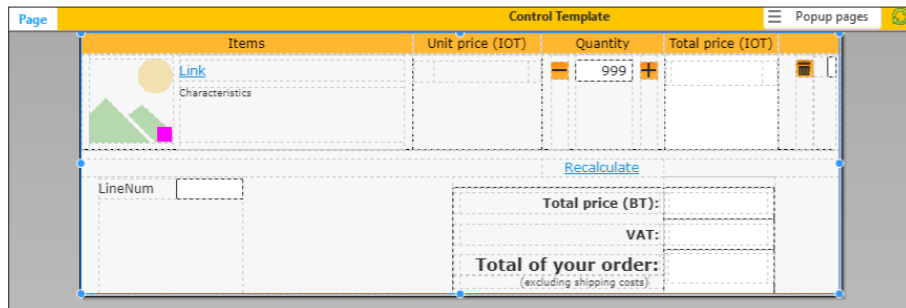
# Control templates

WEBDEV allows you to create control templates. A control template is a set of controls that can be reused in several pages.

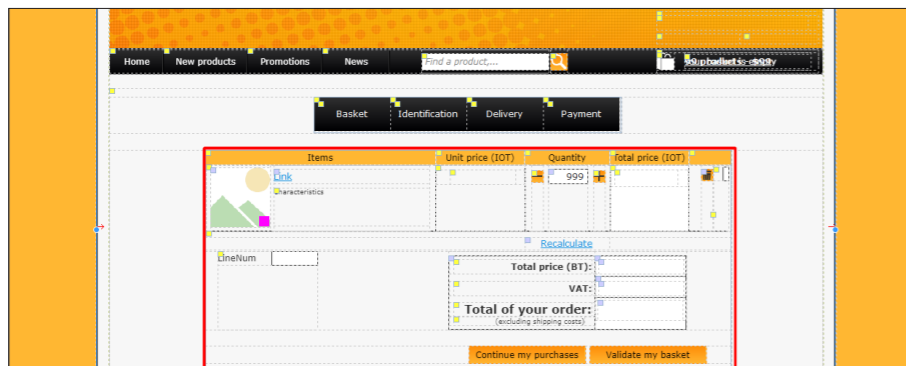
The changes made in a control template are automatically applied to all pages using this template.

A control template is used to:

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



View of a control template.  
The template is bordered by an orange line in the editor.



Template used in a page.  
Elements from the template have a red border and are identified by a yellow square.

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

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

The characteristics of the elements can be dissociated from the template. For example, dissociate the position of a control from the template to move it elsewhere while keeping other changes (code, style, etc.). This is referred to as **control inheritance**. In this case, the elements are identified by a blue square.

# Reports: concepts

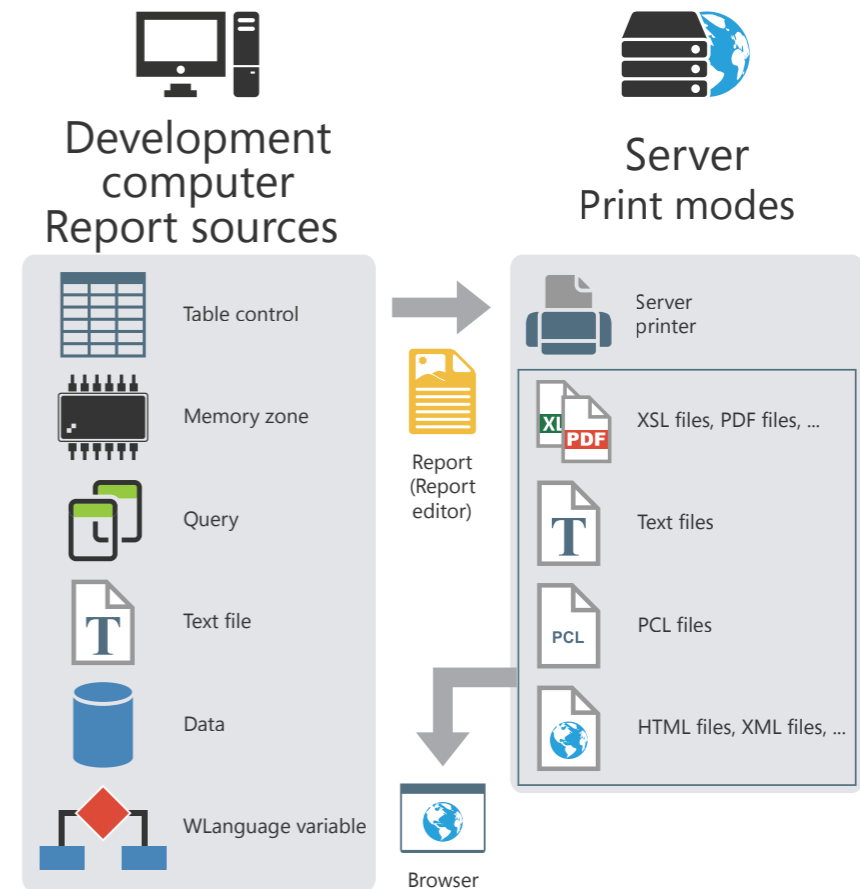
A report provides a custom view of data: data entered in the database, data in a text file, data from a Table control, etc.

The principle for creating a report is as follows:

- the data to be printed comes from a data source (data file described in an analysis, HFSQL view, query, memory area or text file).
- the report groups, sorts and formats the data.

The created report can be:

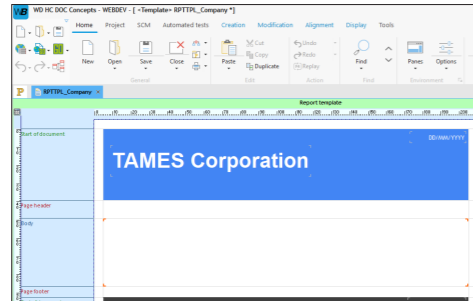
- printed on the printer of the server (for an Intranet site).
- displayed in the user's browser (HTML, PDF, etc.).
- sent to the user by email (PDF).



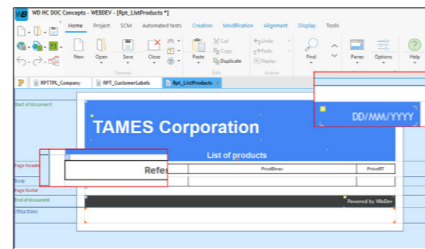
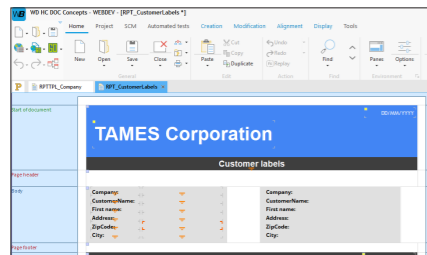
# Report templates

Most organizations use a standardized appearance and layout for their print jobs: date in the upper-right corner in a specific format, page footer with print time and file name, logo in the upper-left corner, etc.

Report templates are used to easily standardize report layouts.



View of a report template in the report editor.



Template used in different reports. Elements from the template are identified by a yellow square. Overridden template elements are identified by a blue square.

A report template can be created:

- via the icon in the quick access buttons. In the "New" window, click "Report", and then "Report template".
- from the current report (on the "Home" tab, in the "General" group, expand "Save" and select "Save as a template").

To create a report based on a template, select the template that will be used when creating the report.

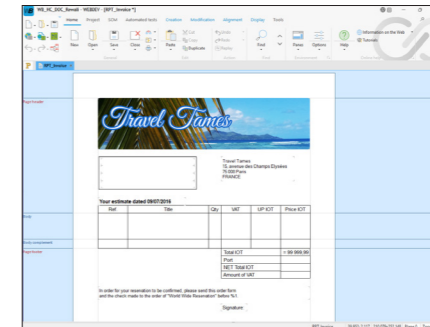
The characteristics of the elements can be dissociated from the template. For example, dissociate the position of a control from the template to move it elsewhere while keeping other changes (code, style, etc.). This is referred to as **inheritance**. In this case, the elements are identified by a blue square.

# Print modes

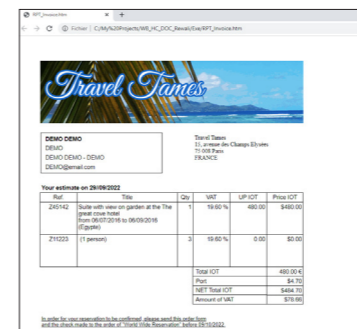
Create reports in the report editor to print the documents generated by your site (invoices, quotes, etc).

To print reports regardless of the user's computer configuration, it is recommended to set at least two print modes:

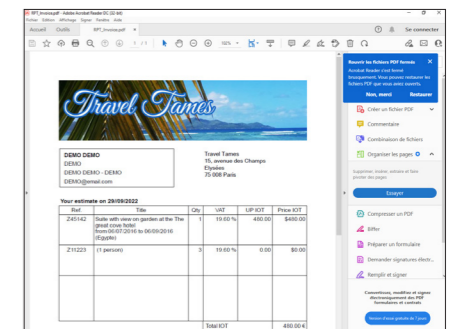
- **Print to HTML:** The report is displayed in a new page of the site. The report can be printed via the browser "Print" button. No specific software is required.
- **Print to PDF:** This print mode requires Acrobat Reader (or a similar program) on the user's computer. The print job is started from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



Report in the WEBDEV report editor



Print to HTML



Print to PDF

# Displaying and linking pages

In most cases, a site includes a set of pages. These pages are linked together via buttons (or links). Clicking a button displays a new page.

Two points must be considered before displaying a page:

- which page should be displayed?
- where should the page be displayed (new browser, frame, etc.)?

## Which page should be displayed?

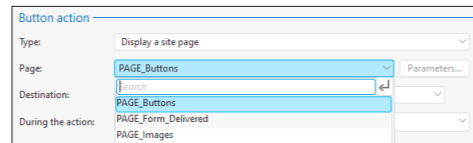
The page to display can be defined:

- in the page editor (no programming is required).
- in the code editor, in an event of the Button control (or Link or Image (with clickable areas), etc.).

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

To define the page to display:

- Open the control description window (right click, "Description").
- In the type of action, select "Display a site page".
- Select the page to display.



### Selecting the page to display through programming

The page to display must be selected through programming when:

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

## Where should the page be displayed?

A page is displayed in a specific "target": current page, new browser tab, etc.

The target can be defined:

- in the page editor: no programming is required.
- in the code editor, in an event of the Button (or Link) control.

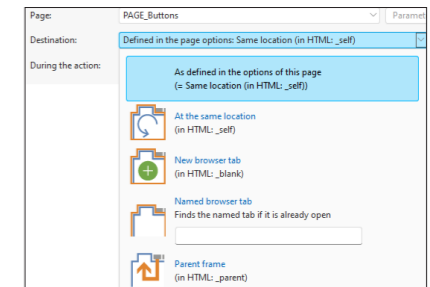
Several WLanguage functions allow you to open a new page (or frameset). The main functions are:

- **PageDisplay** (for pages),
- **FramesetDisplay** (for framesets),
- **PopupDisplay** (for popup pages),
- **PageDisplayDialog** (for modal page display).

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

In a Button control, to define the target of the page to display:

- Open the description of the Button or Link control (right click, "Description").
- In the list of targets, select a preset target or a frame (in the case of a frameset).



**Remark:** The preset targets will be presented later in this manual.

### Selecting the target in the code editor

When the target depends on a choice made by the user, this target must be defined through programming. For example, if the user does not enter their password, an error page will appear in a new browser tab; otherwise, the next page will appear in the current browser.

**ChangeTarget** is used to change the target of an action through programming. This function must be used in browser code.

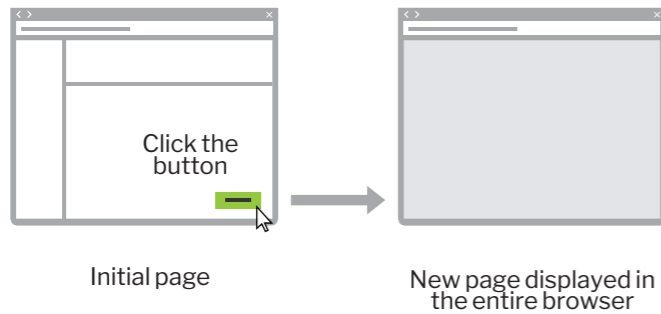
### Choosing a target

When choosing the target for a Button, Link, Image (with clickable areas) control, or a menu option, ... you can select:

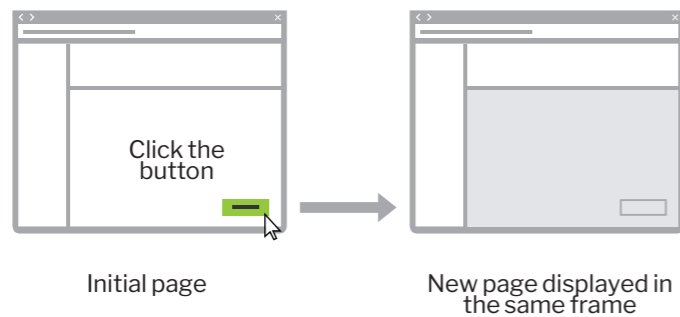
- a preset target: four preset targets are available:
  - Current browser (`_top`)
  - Current frame (`_self`)
  - Parent frame (`_parent`)
  - New browser (`_blank`). In this case, the new browser can be a new browser tab or a new browser with specific characteristics (menu bar, status bar, etc.).
- a frame in the current frameset.

**Remark:** in the diagrams below, the grayed area represents the area where the page is displayed when the Button control is clicked on.

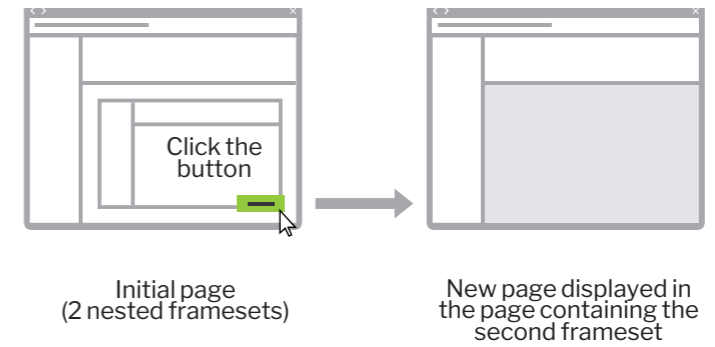
#### Current browser (`_top`)



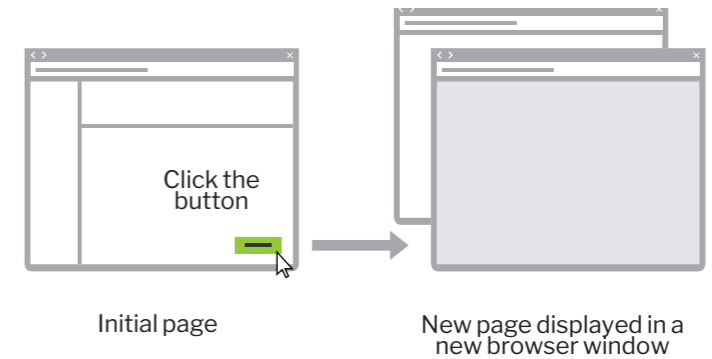
#### Current frame (`_self`)



#### Parent frame (`_parent`)



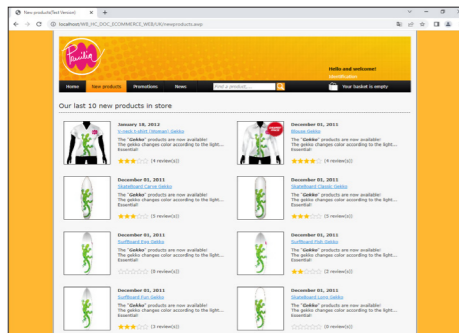
#### New browser (`_blank`)



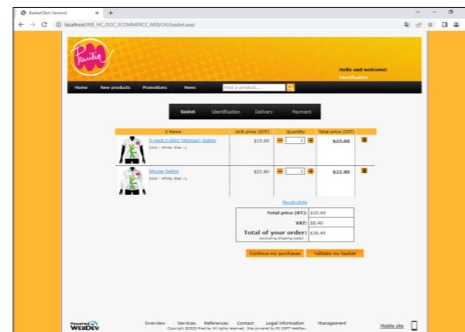
# Performing a process on several successive pages

In a business site, orders can be placed on several pages:

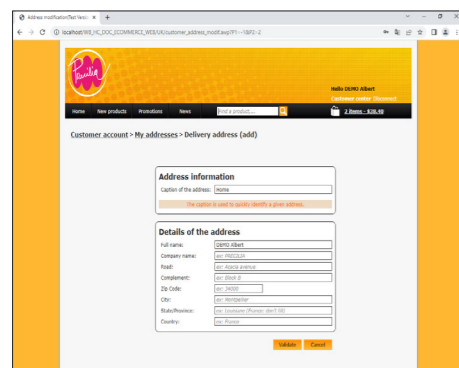
- page 1: Select products to order.
- page 2: View basket.
- page 3: Validate the order and enter the customer details.



Page 1: Products.



Page 2: Basket.



Page 3: Customer details.

In this case, the write-to-file operation must ONLY be performed in the code of the button that will validate the order: all elements must be stored until the order is validated. Don't write to the data files being processed.

**Remark:** You also have the ability to use a transaction but we recommend that you store the order details.

# Style sheets: to simplify the layout

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

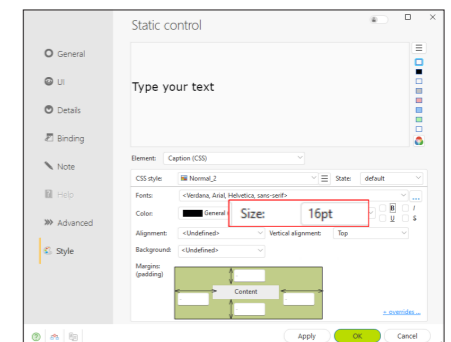
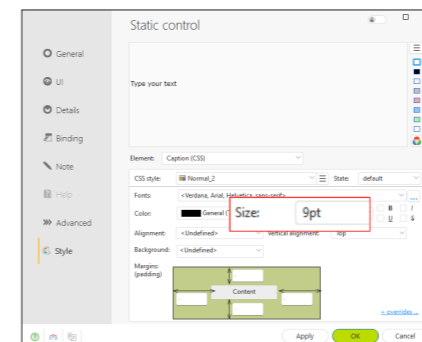
Style sheets are used to define:

- the size, layout and font of the text,
- the alignment and indentation of the paragraphs,
- the style (bold or italic) and color of the text, etc.

And much more.

When you make changes to the style, all the controls associated with it are updated. This gives you the possibility to change the site style book by making only a few changes to the styles used.

Usage example of a style sheet: increasing the size from 9 to 16:

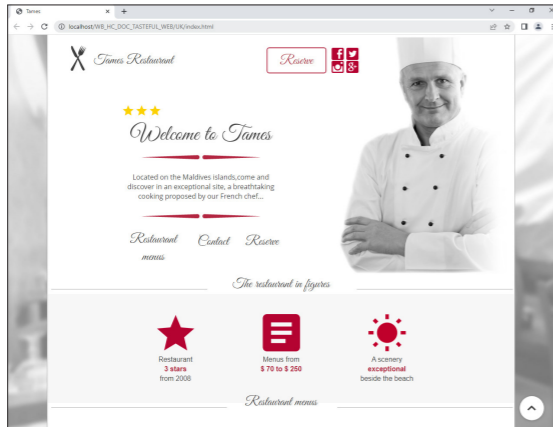


**Remark:** You can edit or create a CSS style. You can write custom CSS code in the "Advanced" tab of the CSS style description window. You can also import CSS code. This code will be added to the style sheet.

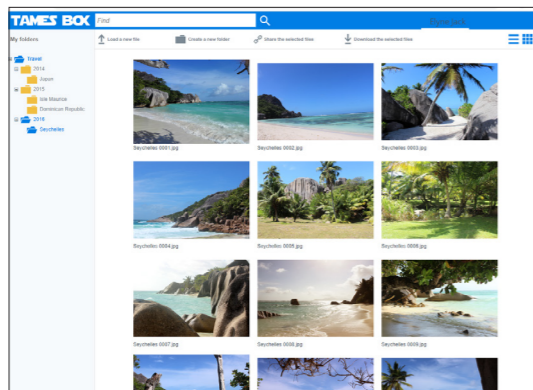
# Center or anchor a site horizontally

The page layout is one of the most important elements when building a site. A site can have one of the following page layouts:

- **site centered:**  
Pages are centered in the browser. If the resolution on the user's computer is higher than the site's optimal resolution, white margins will appear on both sides of the page. The content of the site is centered.



- **site anchored horizontally:**  
Page content takes up all available space in the browser.



**Remarks:**

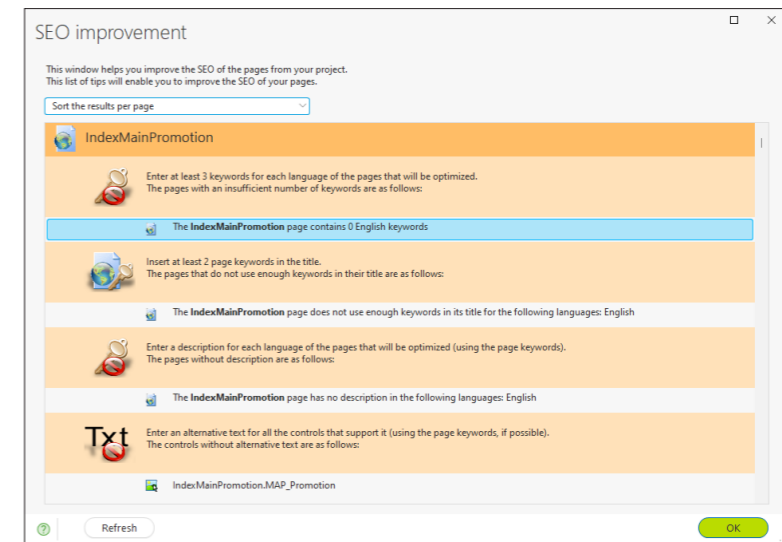
- The type of layout is defined in the page anchoring options ("Anchor" option in the context menu of the page).
- The type of layout (centered or anchored horizontally) set on a page template will be automatically used on the pages associated with the template.

# Site SEO

To attract users to your site they must be able to find it. To do so, your site must come up when the user enters keywords that match your site in a search engine.

To improve your site's SEO, WEBDEV includes:

- an SEO wizard. This wizard lists all the optimizations that can be performed to improve your site's SEO. To start the wizard, on the "Project" tab, in the "Web" group, expand "SEO" and select "SEO wizard".



- the possibility to enter a description and keywords for each page of the site.

You can create a new page and define it as the home page when you develop your project.

**Remark:** SEO is a technique whose rules are defined by search engines (Google, Bing, etc.). These rules are not always explicit and are constantly evolving. Don't hesitate to read books or to visit sites dedicated to this topic.



# SEO of a site in practice

## 1 SEO principles

In order for your WEBDEV site to be visited, Internet users must be able to find it. To do so, your site must come up when the user enters keywords that match your site in a search engine.

### 1.1 Improving the SEO of a WEBDEV site

To improve the SEO of your pages, we recommend that you use Active WEBDEV Pages.

To optimize a site in Session mode, use:

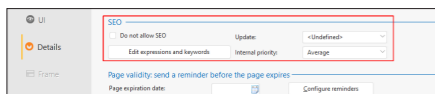
- a home page.
- a section of the site in static mode. The static site will be used for the SEO and will start the site in Session mode.
- a section of the site in AWP mode. The AWP site will be used for the SEO and will start the site in Session mode.

Remark: In AWP, to pass parameters between your pages, we advise you to pass parameters via the URL rather than using AWP contexts. The URL content can be optimized whereas AWP contexts cannot.

## 2 Using (or not) a static or Active WEBDEV Page for SEO

To optimize a static or Active WEBDEV Page of your site:

1. Open the page description window.
2. In the "Details" tab, a specific section allows you to configure the page SEO.



### 1.2 Available tools

To help you, WEBDEV offers:

- **An SEO of each static and Active WEBDEV Page:** for each page, it is possible to define a description and a set of keywords. These keywords will allow users to access the page directly. You can also choose not to optimize a page.
- **A search engine optimization wizard,** which indicates the improvements that can be made on each page for SEO purposes.
- The possibility to **include or use a specific page as home page.** This solution is used, for example, to optimize pages in Session mode, by associating keywords to the home page. This home page contains all the keywords that will allow users to find your site.

3. It is possible to:

- Disable SEO for the current page.
- Allow SEO for the current page by associating expressions and keywords ("Edit expressions and keywords" button).

## 3 Listing a page and entering its keywords

To optimize a static or Active WEBDEV Page of your site:

1. Open the page description window.
2. In the "Details" tab, a specific section allows you to configure the page SEO. In this area, click "Edit expressions and keywords".
3. **In the "Description" tab of the SEO 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. Tips:
  - Use short sentences, limit the number of words (up to 200 characters)
  - This description must entice Internet users to visit the page.**Remark:** This description can be modified dynamically using the **Description** property.

4. **In the "Keyword" tab of the SEO window,** enter the keywords associated with the page. These keywords (or expressions) will be used to optimize the page: if the user enters one of these keywords, the search engine will propose the corresponding page. The keywords are typed as follows: type the first keyword and press Enter to type the next one. 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, etc.).

**Remark:** These keywords can be dynamically modified using the **Keywords** property.

5. Validate the page SEO and description windows.

## 4 Using the SEO wizard

To help you improve the SEO of your sites and pages, WEBDEV proposes an SEO 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 changes to improve the ranking of the pages in the search engines.

Examples of optimizations detected by the wizard:

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

**To start the page SEO wizard:**

1. Open the desired page in the editor.
2. On the "Page" tab, in the "SEO" group, click "Optimize".
3. The SEO 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 SEO wizard for all the pages of a site:**

1. On the "Project" tab, in the "Web" group, expand "SEO" and select "SEO wizard".
2. The SEO 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 Registering your site in search engines

The SEO of a site is made on the search engines directly.

**Remark:** Sometimes, the SEO of a site implies certain fees.

Two types of search engines are available:

- **"Automatic" search engines:**

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

**How can I get my site optimized?**

An "Optimize your site" link is often available for this type of search engine. Simply specify the address of the site and the email of the Webmaster. The site will then be automatically analyzed (analysis based on the keywords and contents of your site) and optimized.

- **"Directory" search engines:**

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

**How can I get my site optimized?**

An "Optimize your site" link is often available for this type of search engine.

In most cases, the SEO procedure is as follows:

1. Choosing a category for the site (leisure, culture, etc.).
2. Filling out a questionnaire about the site to optimize: Internet address of the site, email of the WebMaster, description of the site, etc.
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 they find its content interesting.

# PART 3

Development  
environment

# 28

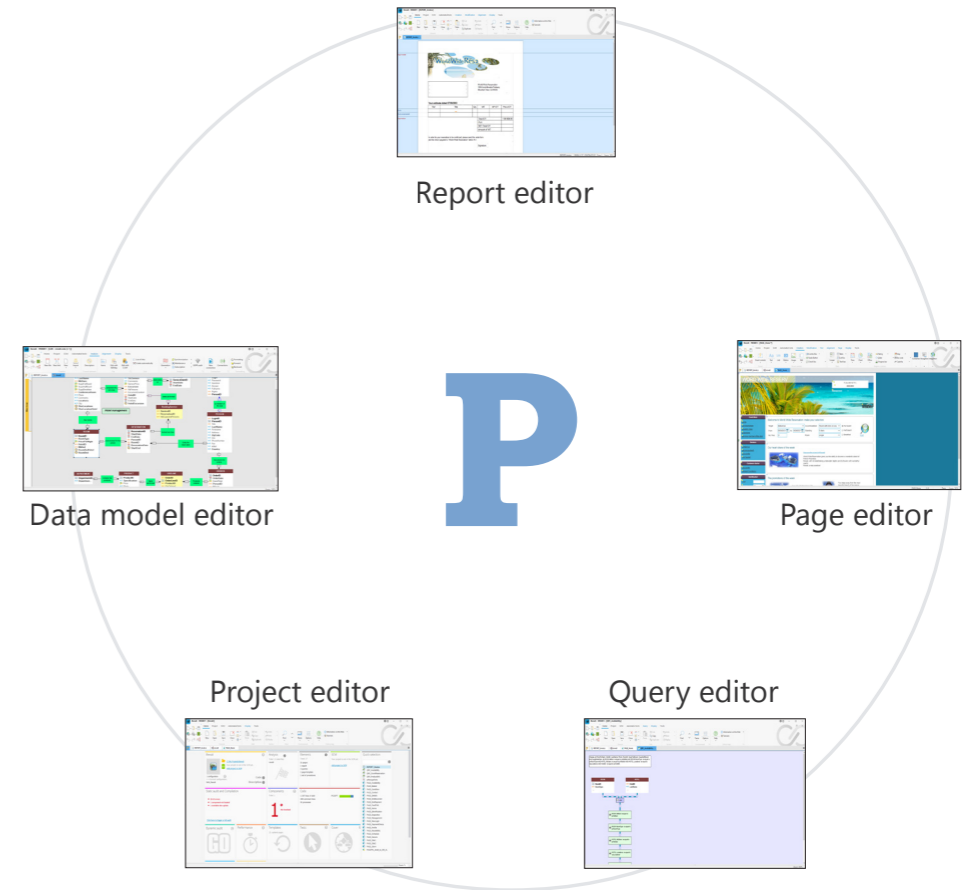
DEVELOP 10 TIMES FASTER

# 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, which adapt to the developer's needs and are integrated into a single environment:

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

- page editor.
- report editor.
- code editor, debugger and compiler.
- image editor.

These editors allow you to create the different elements (pages, reports, databases, programs, etc.) used by the site.

## 2 Types of editors

### 2.1 Project editor

The project editor allows you to manage the project using a comprehensive dashboard.

Various widgets allow you to see:

- the number of project elements,
- the registered incidents,
- the tasks to be performed,
- the checked out elements,
- ...

### 2.2 Data model editor

The data model editor allows you to describe the characteristics and structure of the data files.

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

The data model editor supports two types of database descriptions:

- direct description of the analysis linked to the project (also called Logical Data Model (LDM))
- description of the Conceptual Data Model (CDM) and 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 directly view all the data files and links of your project.

- Creation and description of the data files and their items.
- Automatic formatting of the analysis links.
- Information about the links via tooltips.
- Simplified retrieval of the description of an external database (SQL Server, Oracle, etc.).
- Independent data file descriptions.
- Encryption of data files.
- Automatic modification of the data files when the structure of the files changes.
- Verification of the referential integrity.
- Automatic generation of the analysis.
- Zoom in the data model editor.
- Insertion of comments in the graph of the analysis.
- Relation between the different panes and elements of the analysis.

### 2.3 UML editor

UML is a visual language that can be used to:

- represent an information system using objects.
- generate the object structure of the application (structure of the application in the form of object classes) that corresponds to the information system being analyzed.

The UML editor is mainly used to:

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

Let's see some characteristics of the UML editor:

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

### 2.4 Query editor

The query editor allows you to automatically create queries based on data files.

A wizard helps you create queries: choose the items you want to include and enter the selection conditions via the wizard. The query is automatically generated (in optimized SQL code) and can be tested immediately.

Let's see some characteristics of the query editor:

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

For more details, see the online help about the query editor.

### 2.5 Page editor

The page editor allows you to describe the characteristics of the project's UI.

Multiple templates, skins and types of pages are available. They help you make your pages and 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 pages as they will appear to the user.
- Creation and description of pages and controls.
- Drag and drop to copy, paste or move controls from one page to another.
- Page templates and preset control styles.
- Icon catalogs to associate images with controls.
- Real-time interface checker to position controls more easily.
- Ability to enter the captions of the controls directly in the editor work area.
- Context-sensitive help in the pages.
- Zoom in the page editor.
- Relation between the different panes and elements of the editor.

### 2.6 Report editor

The report editor allows you to easily create reports that can be printed.

Let's see some characteristics of the report editor:

- WYSIWYG editor ("What You See Is What You Get"). You can directly view the reports as they will be printed.
- Easy-to-create, zero-code reports.
- Drag and drop to copy, paste or move controls from one report to another.
- Report skin templates and preset control styles.
- Real-time interface checker to position controls more easily.
- Use of a form in a report background.
- Creation of multi-column labels.
- Ability to edit a report in HTML (to publish it on the Internet for example) or RTF (to use it in a word processing software).
- Zoom in the report editor.

For more details, see the online help about the report editor.

### 2.7 Code editor

The code editor allows you to write all processes/events in WLanguage (programming language common to WINDEV, WEBDEV and WINDEV Mobile). For example, it allows you to enter the source code of the following elements:

- controls,
- windows or pages,
- reports,
- local and global procedures,
- classes and methods, etc.

Let's see some characteristics of the code editor:

- Automatic formatting.
- Automatic completion.
- Glossary of functions.
- Typing error detection and spelling checker.
- Incremental compilation.
- Insertion of WLanguage events specific to the use of the mouse or keyboard.

- History of code changes and ability to roll back.
- Ability to open multiple code windows (to compare code, for example).
- Zoom in the code editor.
- Relation between the different panes and elements of the editor.

For more details, see the online help about the code editor.

### 2.8 Image editor

The image editor allows you to edit images and icons in the project and its elements.

Let's see some characteristics of the image editor:

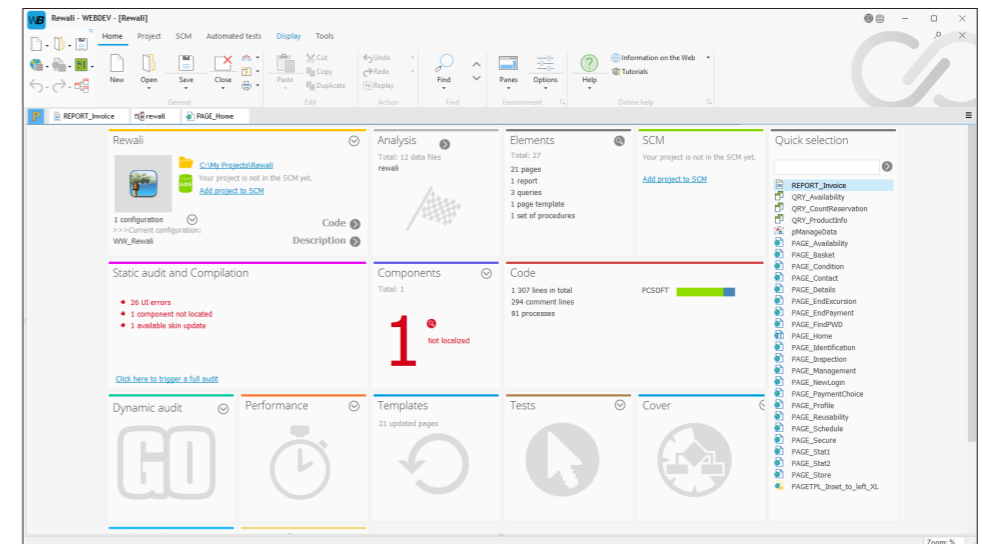
- Crop.
- Resize.
- Symmetry, rotation.
- Text.
- Layers.
- Access to image catalog.
- Transparency.

# Project dashboard

The project dashboard is an essential element for managing WEBDEV projects. It shows an overall and detailed view of the project's progress.

The project dashboard includes several indicators about the project content:

- project statistics,
- incidents,
- tasks,
- status of automated tests,
- result of different audits,
- list of elements checked out from the SCM (Source Code Manager),
- result of action plans (continuous integration), etc.



The elements in this dashboard are displayed as widgets. These widgets can be configured, moved, enabled, disabled, etc. You can also add new indicators.

The dashboard configuration is saved for each user. The dashboard configuration is the same for all the projects of a given user.

# WEBDEV, WINDEV, WINDEV Mobile: 100% compatible

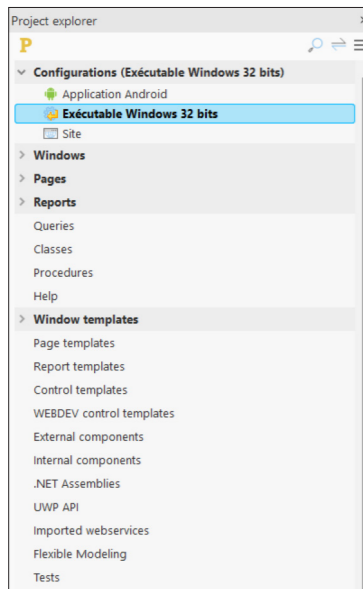
New projects are often cross-platform.

For example, for an ERP system on Windows, it is very likely that beside the main application, there will be sales agents equipped with PDAs or smartphones, stores with mobile devices for inventory management, and Intranet and Internet sites.

**All elements, excluding the UI (pages and windows), are 100% compatible and can be shared between WINDEV, WEBDEV and WINDEV Mobile projects.**

It is thus possible to share sets of procedures or classes, for example, between several projects.

A project can be opened in any IDE regardless of the one that was used to create it.



When you open a project in a different IDE, a wizard appears, allowing you to create a project configuration specific to the IDE used.

For example, if you open a WINDEV project in WEBDEV, you will be able to create a project configuration named "Site", to group all the elements required by the WEBDEV site.

You can also view the elements of each platform from each environment. A project in WEBDEV displays the thumbnails of WINDEV and WINDEV Mobile windows, for example. Clicking a WINDEV window from the WEBDEV project editor opens the WINDEV window (WINDEV must be installed on the computer).

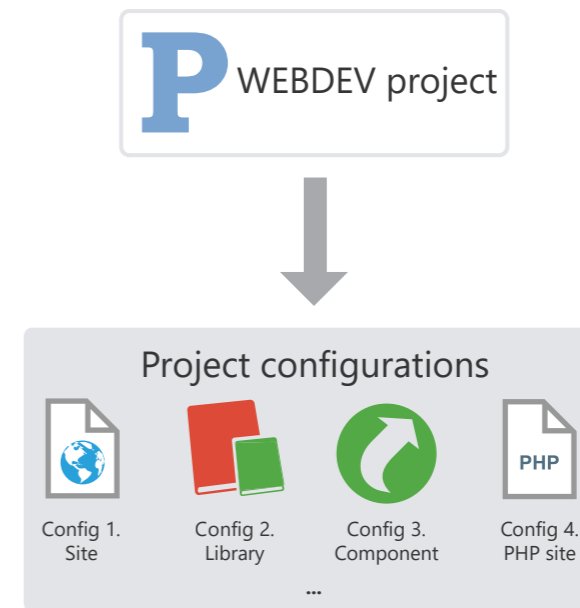
# Project configuration

Project configurations allow you to adapt the same project to multiple "targets".

From the same project, you can create:

- sites with different elements, names, etc.
- different components,
- a Webservice.

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



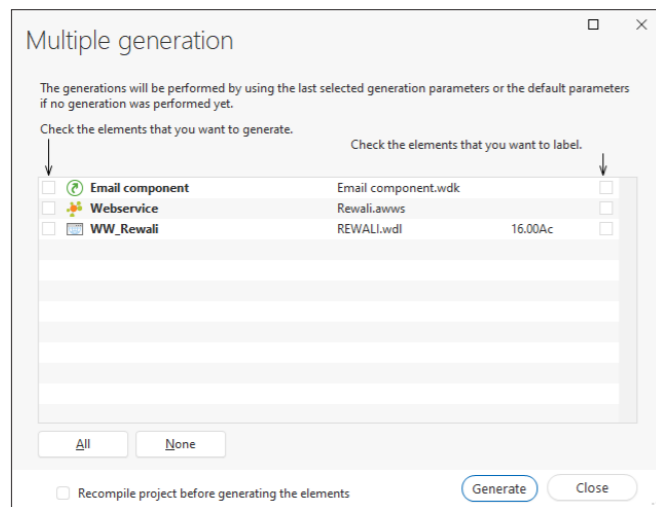
The multiple generation allows you to generate all project configurations (or some of them) in a single operation.

# Multiple generation

Project configurations are used to easily define the different "targets" of your project. Several sites, libraries or 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.

There is a faster method: **the multiple generation**. The configurations to be generated are selected in a single operation and the result is obtained immediately.



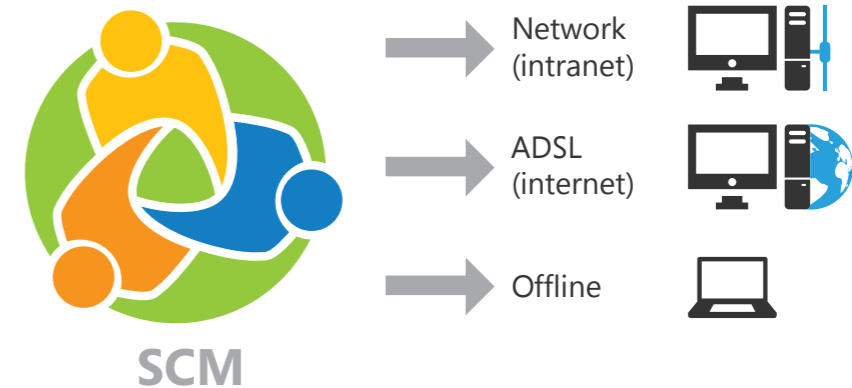
To start a multiple generation, go to the "Project" tab, "Generation" group, expand "Generate configuration" and select "Multiple generation".

# Source Code Manager (SCM)

## Overview

To simplify teamwork, WEBDEV includes a Source Code Manager.

The Source Code Manager allows developers to work together on the same project at the same time and share elements between multiple projects.



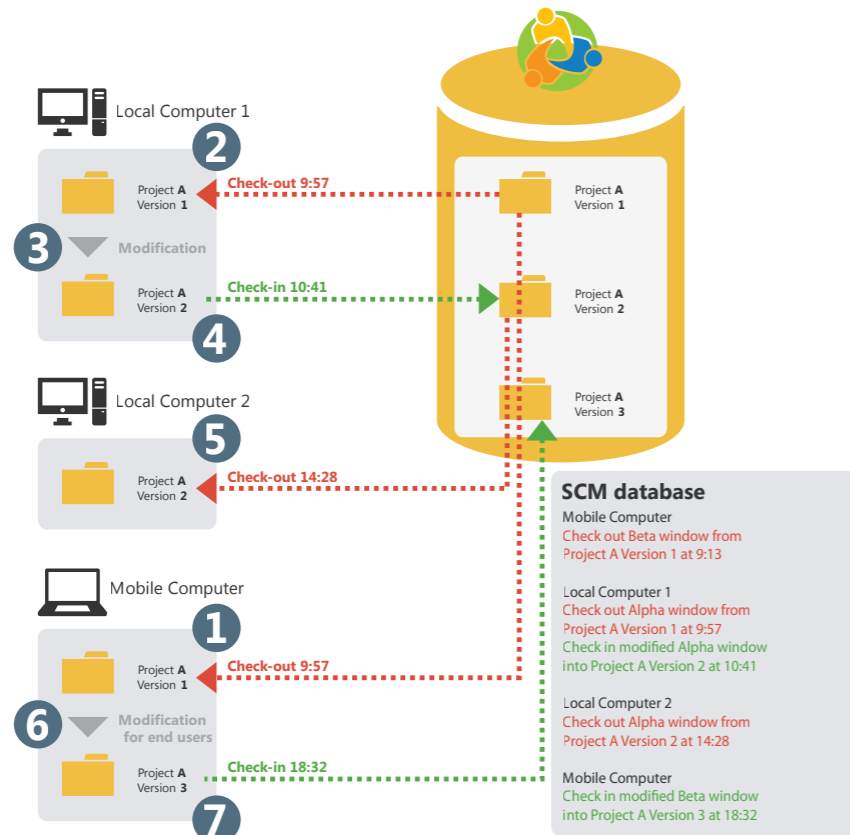
*A repository contains a collection of project elements.  
Each computer has a local copy of the elements required for development.*

Elements in the SCM can be shared:

- via PCSCloud (paid turnkey solution, on a dedicated server),
- via SCMDrive (paid turnkey solution, on a shared server).
- via a network,
- via the Internet,
- in offline mode. In this case, the necessary elements will be checked out from the 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:



A project element (window, report, etc.) can only be checked out by one developer at a time.

Once the checked out elements are modified, they must be checked back in so that the source project can take the changes into account. The repository stores a history of all the project elements since they were created.

Each time 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 multiple developers. These developers must work on a single WEBDEV project and share different resources (queries, classes, etc.).

To share a project, you can use the Source Code Manager.

### 1 Source Code Manager

#### 1.1 Overview

WEBDEV innovates in teamwork management with the Source Code Manager (SCM).

The Source Code Manager (SCM), which is fully integrated in the environment, is used to:

- make teamwork easier and faster,
- store the history of changes and versions,
- save the sources of the development team automatically.

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

**The SCM uses a repository:** procedures, classes, pages, reports, components, analyses, etc.

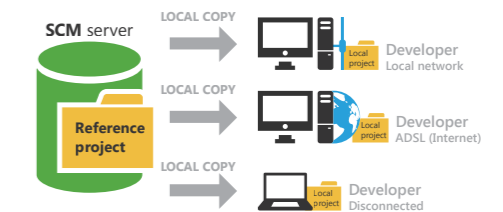
You can choose between a repository:

- in HFSQL Client/Server mode:
  - installed on a custom server,
  - available via a PCSCloud (paid turnkey solution, on a dedicated server) platform,
  - available via SCMDrive (paid turnkey solution, on a shared server).
- in HFSQL Classic mode (installed locally or in a shared directory).

The SCM can be run locally or remotely, via Internet or via an HTTP or HTTPS connection. This feature enables you to work on a project from your office or from a remote site without losing any changes. The SCM can also be used in offline mode.

#### 1.2 Principle

##### Setup



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

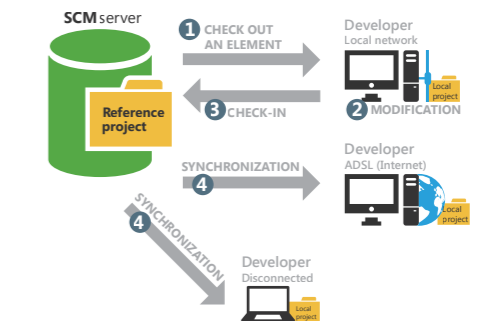
Each developer who uses the Source Code Manager retrieves a local copy of the project.

##### Use

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

If the element is already checked out, the developer can send a message to request that the element be checked back in.

To use the latest version, the other developers must synchronize their local project with the reference project (located in the repository).





**Tips**

The sources of your applications are essential.

These sources must be handled with great care!

Tips for configuring the server that will host your sources:

- Use a dedicated server with a disk with enough space (at least 200 GB).
- Use the Source Code Manager (SCM) in Client/Server mode.

## 2 Creating a repository for the SCM

### 2.1 Overview

To share a project via the Source Code Manager, a repository must be created. This repository must be created once only on the server.

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

The repository can be created:

- via SCMDrive, a "turnkey" solution (recommended). In this case, a shared server is used. This paid solution simplifies all the SCM server setup operations. For more details, see SCMDrive.
- via PCSCloud, a "turnkey" solution. In this case, a dedicated server is used. This paid solution simplifies all the SCM server setup operations. You will use a platform adapted to your needs. For more details, see PCSCloud.
- on an HFSQL Client/Server server: the repository will be in HFSQL Client/Server format. In this case, you must specify:
  - the server,
  - the database,
  - the port used,
  - the name and password of an HFSQL database administrator.

- The hard disks may encounter physical problems: use a RAID I system on your server (several disks that store the same information).
- Use a UPS to protect the power supply of your server.
- Make backup copies of the repository on a regular basis (at least once a week).
- Place the server in a "secure" area and use a firewall.

- locally, in a shared directory on the network: all the SCM users must have full rights on this directory. The repository will be in HFSQL Classic format. Simply specify the network directory where the repository will be created.

### 2.2 When to create a repository?

**The repository must be created only once.**

WEBDEV allows you to create this repository at different times:

- when installing WEBDEV.
- when creating a project that uses the SCM.
- when importing a project into the Source Code Manager.
- in the SCM administrator directly.

Once the repository is created, all shared WEBDEV projects can be imported into it.

### 2.3 Backups

We advise you to make backup copies of the repository on a regular basis. These backups can be performed via the SCM administrator.

## 3 Configuring the project to work with the SCM

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

### 3.1 Adding a project to the SCM

**To add a local project to the SCM:**

1. Open your project in WEBDEV.
2. On the "SCM" tab, in the "Add project" group, click "Add project to SCM".
3. In the wizard, select the repository to be used:
  - A new repository. In this case, select the type of repository you want to create (SCMDrive or PCSCloud, Client/Server or local repository). Provide the required information according to the chosen option.

- An existing repository. Simply select the desired repository.

4. Validate the wizard. The project is added to the SCM.

### 3.2 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, etc.).

Sharing is an important concept of the SCM. There are several methods to share resources. For more details, see the online help.

## 4 Working with the SCM

### 4.1 Project options related to the SCM

Several options are available to configure a project handled by the Source Code Manager. These options are grouped in the "SCM" tab of the project description ("Project" tab, "Project" group, "Description").

- **Propose to get the latest version of the elements when the project is opened:** When opening a project located in the SCM, this option prompts you to get the latest version of the project elements. By default, the latest version of the elements is automatically retrieved.
- **Suggest checking the elements back in when the project is closed:** When the project is closed, this option displays a list of the checked out elements to check some or all of them back in. By default, when the project is closed, the checked out elements are not checked back in.

- **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 actions have been completed in the project, the project file is automatically checked back in. This option disables the "Master/Guest" management on the project. This option can also be enabled on the "SCM" tab, in the "Project" group, by expanding "Master/Guest" and selecting "Manage project check-out automatically".

## 4.2 Checking out an element

The SCM includes two modes to check out the project elements:

- **standard mode:** if you open an element that has not been checked out from the SCM, a panel indicates that the element must be checked out to modify it. The element can be checked out immediately (check-out button found in the dialog box).
- **automatic mode:** if you try to modify an element that has not yet been checked out, the SCM automatically prompts you to check it out. Once the check-out is validated, the element can be modified.  
Remark: this mode is not recommended when using the SCM with a slow Internet connection.

To change the check-out mode:

1. On the "Home" tab, in the "Environment" group, expand "Options" and select "General options of WEBDEV".
2. In the "General" tab, the "Check out elements on first modification" option allows you to switch all subsequent projects to automatic mode.

### Opening a project element to change its characteristics

To change the characteristics of a project element included in the SCM:

1. Check out the element from the Source Code Manager.
2. Select the check-out mode for the element. The check-out mode can be:
  - **exclusive:** nobody can check out this element until it is checked back in. The element can be checked out for tests only.
  - **for test:** the element can be modified but the modifications will not be checked back in.
  - **multiple:** the element can also be checked out by other users. In this case, the differences between the different element versions can be viewed when the element is checked back in.
3. Validate. The element is opened. The title bar indicates that the element is checked out.

## 4.3 Checking an element back in

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

To check in an element, go to the "Project explorer" pane and select "Check in" in the context menu of the element.

A screen allows you to perform the following actions before checking the element back in:

- view the changes made.
- compare the element in the repository with the local (checked out) element.
- access the element history in the repository.

You can check in the changes made to the element while keeping the element checked out ("Keep element checked out" option).

## 4.4 Project management modes

Two management modes are available with the SCM:

- Managing the project in Master/Guest mode.
- Managing the project in automatic mode (by default).

### 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 only required to:

- change the project characteristics and check these changes back into the repository.
- check all elements back in to create the site setup program.

To switch from master mode to guest mode, go to the "SCM" tab, "Project" group, expand "Master/Guest" and select "Become guest on the project (and check all in)".

In this case, the Source Code Manager prompts you to check in all the project elements (including the .WWP file).

To switch from guest mode to master mode, go to the "SCM" tab, "Project" group, expand "Master/Guest" and select "Become master on the project".

## Caution: Changing the options of the project:

All project users (master or guests) can change the project characteristics (first page of the project, animation, code style, etc.). However, only the project master will be able to check these changes back into the Source Code Manager. Changes made by a guest will be lost when the project is updated from the repository.

## 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 actions have been completed in the project, the project file is automatically checked back in.

With the automatic mode, you do not need to manage the "Master/Guest" mode in the project.

## 5 Working in offline mode with the SCM

The Source Code Manager allows you to easily work in offline mode.

For example, this mode allows a developer who works on a laptop to continue working on a project located in the repository while offline.

The principle is simple:

- before disconnecting, go to the "SCM" tab, "Other actions" group, expand "Remote work" and select "Disconnect to work offline". Before working offline, it is recommended to check out

the different elements you will work on (these elements will appear as "already checked out" for the other users). Then, you can work on your project locally. You can handle the different elements in the project directly.

- on reconnection, on the "SCM" tab, in the "Other actions" group, expand "Remote work" and select "Reconnect and synchronize". Then, simply check the modified elements back in.

For more details, see the online help.

## 6 SCM administrator

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

It allows you to:

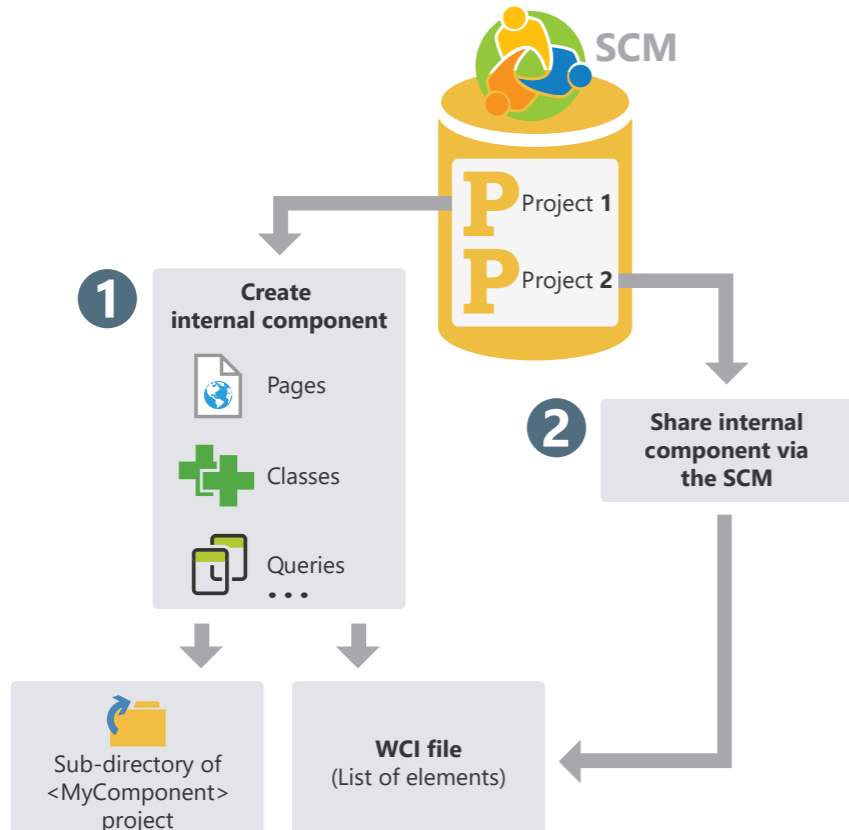
- manage repositories (create or connect to a repository).
- manage branches.
- manage files and directories located in a project in the repository (add, delete, rename files and directories).

- manage the different files of the repository (check-out, check-in, share, etc.).
- start some tools (options, maintenance, etc.).
- restore a project version.
- see the history and differences in the versions.
- undo check-outs (in administrator mode).
- clear, save and restore a repository.
- add any type of file to the repository (.doc, .xls, .pdf, etc.).

# Internal component

An internal component is a set of elements from a project. This set is used to:

- Organize a project: you can create internal components to group the elements of a project, by functionality, for example.
- Share elements between multiple projects, via the SCM (Source Code Manager).



The elements of an internal component can be private or public:

- Private elements can be handled by the other elements of the component.
- Public elements can be handled by the elements of the project that uses the internal component.

## Internal components in practice

### 1 Overview

An internal component is a set of elements from a project. This set is used to:

- **Organize a project:** you can create internal components to group the elements of a project, by functionality, for example.
- **Share elements between multiple projects:** The elements in an internal component can be shared between multiple projects via the SCM.

Compared to standards components, internal components have the advantage that they can be debugged directly from the project in which they are used.

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

When you use a standard component, the "public" elements cannot be handled directly. To modify the standard component, you need to open the corresponding project.

### 2 Creating an internal component

#### 2.1 Steps to follow

To create an internal component:

1. On the "Project" tab, in the "Project configuration" group, expand "New configuration" and select "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 that corresponds to the description of the internal component. This name will also be used to create a subdirectory in your project with all the elements of the internal component.
  - its caption.
  - its description.
3. Select the elements that will be included in the internal component. An internal component can contain all types of elements from a project: pages, reports, templates, etc.

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

**Remark:** The automatic completion will display accessible (or "public") elements. They can be used by the elements of the project or from another internal component. Inaccessible (or "private") elements can only be used by another element of the internal component (the automatic completion will display "private" elements only from the elements of an internal component).

5. Specify how the data and execution contexts are managed. There are three modes to manage data and execution contexts:

- **Use the project analysis or no analysis (full share):** The internal component accesses the data files of the project. The internal component and the project use the same execution contexts. This is the default mode if the internal component does not use an analysis.

- **Use the project analysis with different execution contexts (advanced mode):** The internal component accesses the data files of the project. The internal component and the project use different execution contexts. This mode is reserved to specific cases.
- **Use a specific analysis:** The internal component accesses its own data files. The internal component and the project use different execution contexts. This is the default mode if the component uses an analysis. In this case, the analysis used by the internal component must be specified. You can also create a new analysis directly. This analysis will be associated with the internal component.

6. Validate the new internal component. At any moment, you can:

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

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

### 3 Sharing internal components (via SCM)

The internal components can be shared between multiple projects via the SCM.

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

- create the internal component from a project located in the SCM. The internal component will be automatically saved in the SCM.

### 2.2 Internal component and analysis: total autonomy

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

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

The elements generated by the RAD for the internal component will be saved in the internal component directory. If changes are made to the analysis of the internal component, the RAD will prompt you to generate the elements these changes applied to.

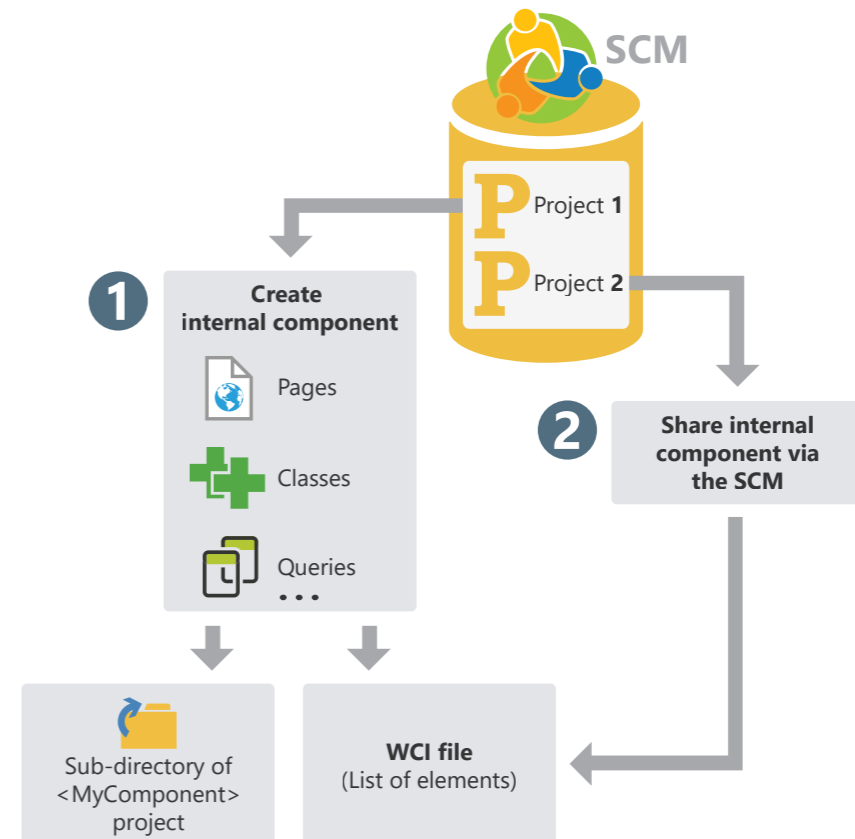
- import a project containing one or more internal components into the SCM. The internal components will be automatically included in the SCM.

# External component

An external component is a set of WEBDEV elements: pages, reports, analysis, etc. This set of elements performs a specific operation. For example, an external component can correspond to one of the following actions:

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

A WEBDEV external component can be redistributed to other WEBDEV developers (free or not). Developers will be able to easily integrate the feature included in the external component into their websites without having access to the corresponding source code (if it has not been distributed with the component). 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 WINDEV, WEBDEV or WINDEV Mobile elements: windows, pages, reports, analysis, etc. This set of elements performs a specific operation. For example, a component can correspond to one of the following features:

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

#### Remarks:

- To optimize the organization of your projects, you can use internal components. For more details, see "Internal component", page 100.
- In this chapter, we will refer to "external component" as "component".

**An external component can be redistributed to other WINDEV, WEBDEV or WINDEV Mobile developers** (free of charge or not). These developers will be able to easily integrate the feature included in the component into their application (or site). The component will therefore be integrated into and distributed with the application (or site).

When creating the external component, the author specifies:

- which elements in the component are visible to the user. Visible elements will be accessible programmatically, or via the "Project explorer" pane.
- how to use the component (use mode, the parameters used to access the component functionalities, etc.). A short documentation (to be completed) is automatically generated and associated with the component.

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

An external component can be:

- defined and generated from an existing WINDEV, WEBDEV or WINDEV Mobile project. A "Component" project configuration will allow you to select the elements of the existing project to be integrated into the component.
- created from an "External component" project. By default, this project is composed of an "External component" platform which will contain the elements required to create and generate the component.

This external component can then be integrated into other WINDEV, WEBDEV or WINDEV Mobile projects.

All the elements of the component are totally independent of the project in which it is integrated. The external component can have its own analysis, windows, reports, data files, etc.

When generating the external component, simply specify if the project that uses the component will access these elements. The project will not be able to read or manipulate elements that are not accessible.

The component includes three files:

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

### 1.3 What elements are included in an external component?

An external component contains:

- the elements to be distributed (windows, reports, etc.). These elements can be accessible (or not) when importing the component into a WINDEV, WEBDEV or WINDEV Mobile project. These elements cannot be modified from the project that uses the component.

- a short description of the component.
- a help text to reuse the component. This help is generated from the code comments.

Remark: By default, if an external component uses an analysis and data files, the HFSQL functions will handle these elements in an independent HFSQL context. These parameters can be changed in the advanced options of the component. For more details, see "Advanced options of an external component", page 108.

## 2 Creating and generating an external component

Below are the steps to create an external component:

1. Creating the external component.
2. Developing the elements of the external component.

3. Defining the advanced options of the external component.

4. Generating the external component.

The external component can be distributed and reused.

## 3 Distributing an external component

### 3.1 Overview

Once the component has been created, tested and generated, it can be distributed to other developers. Multiple methods are available:

- **Direct use of the external component**  
Example: the external component is directly used on the computer of the developer who created it.
- **Standard distribution**, providing the necessary files.  
Example: the external component is intended to be used within the same company, by several developers. In this case, the necessary files can be copied to a network drive.
- **Distribution via the SCM:**  
This distribution mode allows all projects in the SCM to directly use a component published in the SCM. For more details, see SCM and components. Example: A company shares several projects via the SCM. The projects used to create the external components and the projects that use the components share the same SCM.
- **Distribution via a setup program** (with a setup procedure of the component similar to the one used for the applications).  
Example: This distribution mode is recommended if the external component is intended to be sold or distributed on a large scale, with regular updates.

### 3.2 Direct use of the external component

The external component is created and used on the same computer.

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

The generation directory is the directory associated with the "External component" project configuration.

**Caution:** The files in the <Component name>.WDO file will be automatically copied to the EXE directory of the project that uses the component.

If the external component uses specific files (data files, text files, etc.), a <Component name>.WDO file must be created (or completed) once the component is generated. This file contains the list of external files (data files, text files, etc.) used by the component. For more details, see WDO file.

### 3.3 Standard distribution of external components

The standard distribution of an external component consists in copying the files required by the component to a given location. The files are copied to a specific directory. This directory can be located on a network server, for example.

When the external component is imported into a project, all you have to do is select the WDI file corresponding to the component in the distribution directory.

To distribute an external component, you must provide:

- The files automatically generated by WINDEV or WEBDEV (<Component name>.WDK and <Component name>.WDI files)
- If necessary, the specific files handled by the component (data files, initialization files, etc.) as well as the <Component name>.WDO file. The <Component name>.WDO file contains the list of files that must be provided with the component. For more details on this file, see "Distributing an advanced component: WDO file" page 109.

### 3.4 Distribution via a setup procedure

The distribution of external components using a setup procedure requires providing a setup program to the users. This program installs all the files required for the external component in the directory specified by the user.

This installation mode allows you to automatically manage:

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

## 4 Using an external component in a website

An external component can be reused at any time in any WEBDEV site: you just need to know the directory that contains the component files.

When a new version of the component is available, you just have to install the new files in the setup directory of the component (depending on the setup mode used).

For more details on the different modes to install a component, see "Distributing an external component", page 105.

- the installation of specific tools (MDAC, ODBC driver for HFSQL, etc.).
- the automatic update of data files used by the external component (if necessary).
- the uninstallation of the external component.

To create a setup procedure for an external component:

1. Create the setup procedure of the external component directly from the quick access buttons:

2. In the different steps of the wizard, specify:

- the name of the external component.
- the languages of the setup procedure.
- the title of the main window of the setup program. You can configure the interface of the installer via the "Advanced" button.
- the files installed. You can add specific files on top of the default component files. The WDO file will be automatically created from this list.
- the default component setup directory. You can configure the changes to be made to the system via the "Advanced" button.
- the optional modules to install.
- the installation media, security options and advanced parameters.
- ...

By default, the files required to install an external component are created in the "<Project name>\EXE\INSTALL\_COMPO<Component name>" subdirectory.

**Reminder:** When creating the installation of an application, the necessary files are created in the "<Project name>\EXE\INSTALL" subdirectory.

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

If the component was distributed with a WEBDEV site, the component must be updated. For more details, see "Deploying a site containing an external component", page 107.

## 5 Deploying a site containing an external component

### 5.1 Overview

Sites that use one or more components are deployed in the same way as standard sites: on the "Project" tab, in the "Generation" group, expand "Deploy site".

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

- <Component name>.WDK,
- the files required for the component and the application.

### 5.2 Updating components and deployed websites

When updating a component and/or a deployed site, the following elements can be installed on the deployment computer of the site:

- the site, the component and the necessary files.
- the site and the files required for the site.
- the component and any necessary files.

For more details on the compatibility between sites and components, see the online help.

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

#### 1. Recompiling 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 to the component and must be taken into account.
- The parameters of some procedures have been modified.
- Incompatibility between new and earlier versions of the component.
- ...

#### 2. Distributing the .WDK file directly

Provide an upgraded version of the component (.WDK file) to the end users without recompiling the project.

In most cases, this possibility applies when:

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

## 6 Modifying an external component

### 6.1 Overview

A component can be modified at any time once it has been created and generated. These are some of the changes that can be applied:

- add elements to the component.
- remove elements from the component.
- change rights on the elements of the component.
- make changes to one of the elements of the component.

In any case, the component must be regenerated to take these changes into account.

### 6.2 Types of compatibility

Compatibility is linked to the versions of the component. If changes are made to the component, the sites that use the component may encounter runtime problems if they are not synchronized with it.

Compatibility is an important step when making changes to a component. There are two types of compatibility:

- **backward compatibility:** the version of the component (.WDK file) used to compile the sites must always be greater than or equal to the version that has already been deployed. This type of compatibility is automatically managed. If the version of the component installed with the site is earlier than the version used by the site, the site will not run.
- **forward compatibility:** you can prevent the use of a new version of a component with a project compiled using a previous version. You need to recompile the projects to use the new version of the component.

## 7 Advanced characteristics of external components

### 7.1 Automatic documentation

External components must include a technical documentation.

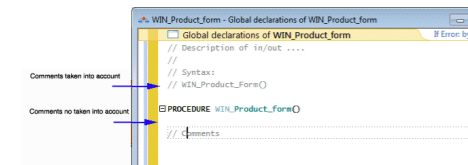
WINDEV, WEBDEV and WINDEV Mobile help you create this documentation via the following options:

- **A general overview of the component.** This overview can be entered when generating the component.
- **A technical documentation** generated automatically from the comments in the code of the component elements.

#### Which code comments are taken into account?

The following comments are automatically taken into account to create the component documentation:

- Comments at the beginning of the WLanguage procedures.



- Comments at the beginning of the following events:
  - Initialization code of windows or pages,
  - Initialization code of reports,
  - Initialization code of classes,
  - Initialization code of sets of procedures.

#### When is the documentation generated?

The documentation of the component is created the first time the component is generated ( in the quick access buttons).

When generating the documentation:

- the comments in the code are used to create the documentation.
- if there are no comments in your code, the expected input and output parameters of each accessible element of the component will be automatically included in the documentation. The corresponding comments are automatically created in the different elements.

**Remark:** The component documentation can be regenerated each time the component is generated ("Regenerate" button).

**Caution:** If you use the "Regenerate" option, the changes made to the documentation in the generation wizard will be lost.

#### How to access the component documentation?

The **general component overview** is automatically displayed in the component description window.

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

- when a component is included in a project: on the "Project" tab, in the "Project" group, expand "Import" and select "An External component".
- in the description of the components included in the project.
  - In the "Project explorer" pane:
    - In the "External components" folder, select the desired external component.
    - Right-click to open the context menu and select "Description".
    - In the "Details" tab, you can get the details of the generated documentation.

The **documentation specific to an element** of the component (window, etc.) can be viewed:

- by double-clicking the desired element in the Project explorer pane or by pressing F2 in the code of the element.
- from the "Project explorer" pane:
  - Select the "External components" folder.
  - Right-click to open the context menu.
  - Select "List of external components imported into the project".

### 7.2 Component element visibility

When you create a component, you can define which elements of the component will be accessible to end users.

- If an element is accessible, the user will see it in the list of project elements. Users will be able to handle these elements programmatically (like any other element in the project).

**Remark:** The code of this element is not visible.

- If the element is not accessible, the user will not even know that it exists.

**Caution:** Depending on the declaration mode the of the project elements (classes, sets of procedures, etc.), the accessible elements may change.

### 7.3 WDO file

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

- `<ComponentName>.WDK:` Contains all the elements that must be redistributed (pages, reports, ...) with the component
- `<ComponentName>.WDI:` Interface of component. This file contains:
  - a help for using the component when it is checked back in
  - the elements required for using the component in the project (compilation information, ...)

These files must be distributed along with the component.

**If the external component uses additional elements** (data files, for example), the following files must be added in the project EXE directory:

- a `<Component name>.WDO` file: This file contains the list of external files (data files, text files, etc.) used by the external component. These files must be provided and installed with the external component.
- the files that must be distributed with the external component. These files can be in a specific subtree. In this case, the code of the external component must manage the access to these files based on this tree structure.

### 7.4 What is a WDO file?

The `<Component name>.WDO` file is a TXT file that can be created and edited at any time. This file can be created and edited with Notepad, the Windows standard text editor.

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

This ".WDO" file can contain:

- **the full name of the file.** For example: `C:\ComponentsWD\PickerComponent\InitialStatus.INI`
- **the file name.** This file will be searched in the current directory of the external component. For example: `InitialStatus.INI`
- **a file name that uses a relative path.** The available syntaxes are as follows:
  - `Directory\FileName.xxx` to specify a subdirectory of the current directory.
  - `.\FileName.xxx` to specify the current directory.
  - `..\FileName.xx` to specify the parent directory. For example: `\PickerComponent\InitialStatus.INI`

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

When including the external component, the tree structure specified in the WDO file will be kept and reproduced in the EXE directory of the project. For more details, see Using an external component in an application.

**Example:** The "Zip Code" component uses a "Cedex" data file (Cedex.fic and Cedex.ndx files). This data file is located in the EXE directory of the component creation project.

To provide and install the component with the data file, the WDO file must be created in the EXE directory of the component creation project. This file must contain the following lines:

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

#### Distributing a component with a WDO file

To distribute a component that uses a WDO file, you need to provide:

- the WDK file (if necessary)
- the WDI file
- the WDO file
- all the necessary files referenced in the WDO file.

# Generation modes

In addition to websites, WEBDEV allows you to generate various types of projects.



## Sites

Sites are the most common generation mode. WEBDEV is used to generate:

- static or dynamic sites (Session or AWP).
- PHP sites.



## Libraries

A library is a single file that groups several elements of a WEBDEV project: pages, reports, etc. You have the ability to generate stand-alone libraries that can be used by other sites.



## External components

The external components are application bricks allowing you to share one or more specific features between several applications. A component generated by WINDEV can also be used in a WEBDEV or WINDEV Mobile project.



## SOAP or REST 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, SOAP or REST protocols.

**Remark:** A Webservice must be deployed on a WEBDEV Application Server in order to be used.

## PHP generation in practice

### 1 Overview

WEBDEV allows you to generate PHP sites without prior knowledge of PHP. The site can be entirely developed in WLanguage, like any standard WEBDEV site. However, you can write 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, an ODBC driver (MyODBC, ODBC for Oracle, etc.) or a Native MySQL Connector (also called Native Access) will be used to access this database.

The ODBC driver is specific to each database (it can be downloaded from the Website of the database publisher).

A PHP site includes ".php" pages and requires a PHP engine to be run.

To compile a PHP project created with WEBDEV, a PHP engine must be installed on the development computer. Several PHP engines are available on the Internet. Download and install the desired one.

For example, you can use EasyPHP (<http://www.easyphp.org/>). This engine is given as an example. You can use any other PHP engine.

**Remark:** Prerequisite:

- PHP 5 servers: Minimum recommended version: version 5.3.
- PHP 7 servers: Minimum recommended version: version 7.0.22.
- PHP 8 servers: Minimum recommended version: version 8.0.10.
- Required MySQL database.

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


It may be useful to develop some sites (especially small ones) using PHP to make it easy to host them at no cost. You can now do so using WEBDEV.

**Remark:** 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

A PHP project is created like any other WEBDEV project:

1. Click  in the quick access buttons of WEBDEV. The new element window appears: click "Project".
2. The project creation wizard starts.
3. In one of the first steps of the wizard, you can choose the type of project that will be generated. Select "PHP" and follow the steps of the wizard. Once the project has been defined as a PHP project, all available WEBDEV tools automatically adapt to that type of project:

- The RAD tool specific to PHP generates PHP code for many 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 PHP site is deployed via WDDeploy.

**Remark:** It is recommended to create a new WEBDEV project to develop a PHP site rather than converting a WEBDEV site to PHP.

#### 2.2 RAD for PHP

As with any project associated with an analysis, you can use the RAD methodology to generate the pages of your site.

The RAD methodology takes into account all the specific features of the PHP generation (controls, events and functions that can be used in a PHP site generated from WEBDEV).



### 2.3 Testing a PHP site in the editor

When testing a PHP site, the site is compiled in PHP and opened in your browser. You can directly see the PHP pages generated by WEBDEV.

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

The compilation of a PHP site is used to:

- identify WLanguage programming errors.
- see the controls, events and WLanguage functions not available in PHP. These errors appear in purple in the "Compilation errors" pane.

For more details on WEBDEV elements that are not available for PHP sites, see the online help.

### 2.4 Deploying a PHP site

PHP sites and static sites are deployed using the same method. When creating the setup procedure, WDDeploy helps you implement your PHP site.

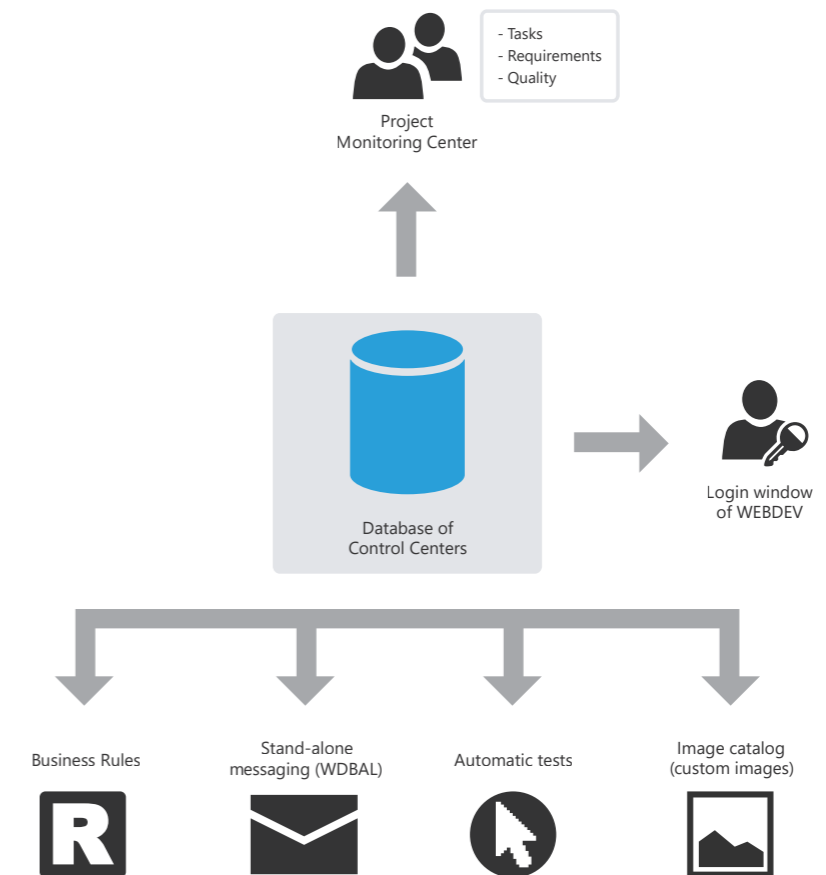
For more details, see the online help.

# Project Monitoring Center

WEBDEV includes a Project Monitoring Center to manage development projects. The Project Monitoring Center allows you to:

- Manage project requirements,
- Monitor the progress of a project (schedule tasks),
- Manage bugs and changes made to a project.

The Project Monitoring Center uses an HFSQL Classic or Client/Server database: the Control Center database. This database is shared between different tools available in WEBDEV:



Remark: When installing WEBDEV, the setup program prompts you to:

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

# Managing requirements

A requirement is a need regarding an action that should be performed by a product (or a service).

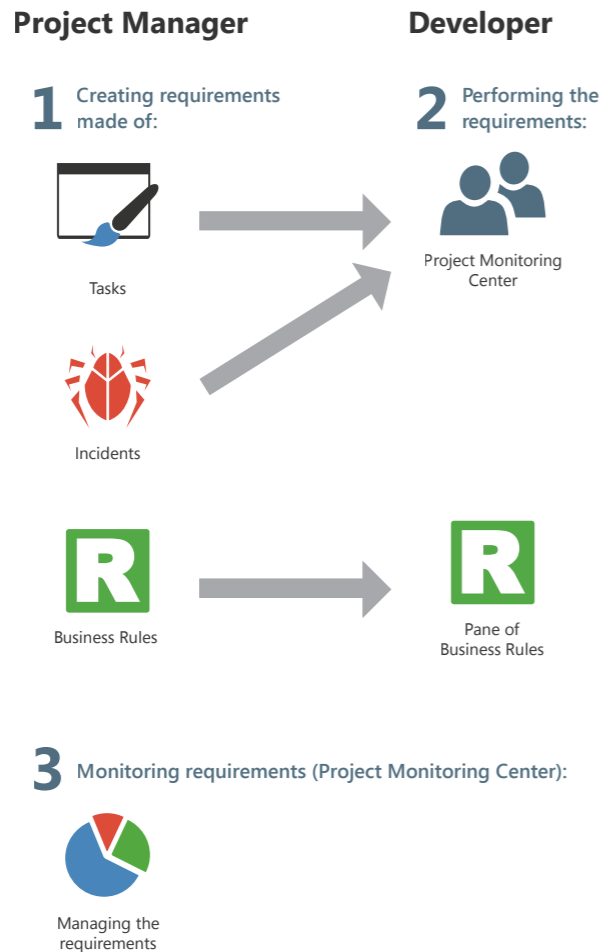
The Project Monitoring Center allows the project manager to manage a development project.

To do so, you must:

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

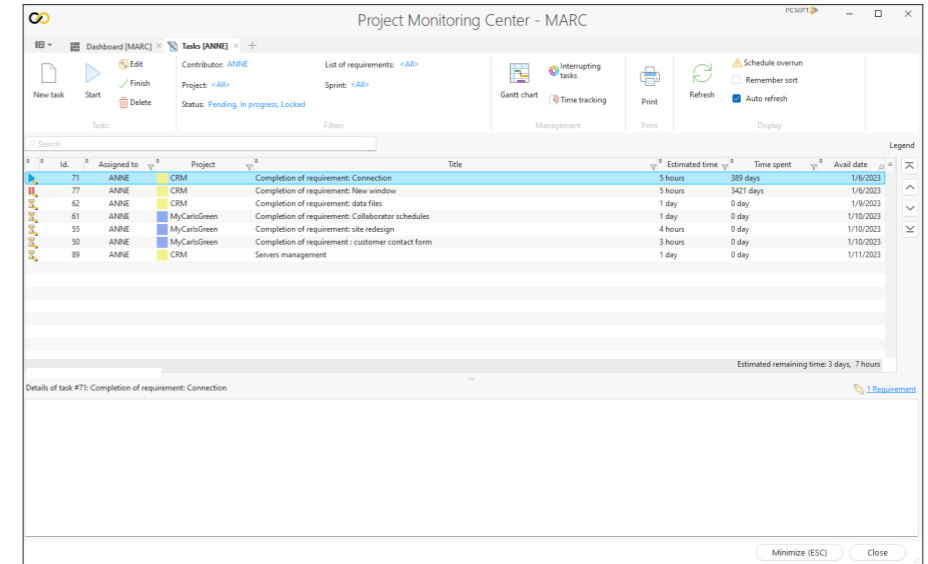
Each developer performs the different tasks assigned to them.

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



# Managing tasks

The Project Monitoring Center allows the project contributors to manage their task schedule. These tasks can be linked to certain requirements and correspond to multiple projects.



Once the project tasks have been defined, you can follow their progress in the Project Monitoring Center. The time spent on a task is calculated automatically, it does not require any specific action and does not generate any particular constraints.

When the relevant project is opened, the Project Monitoring Center requests or indicates the current task. As soon as a task is completed, simply indicate that the task is finished and specify the new task.

A task can be linked to a project element (window, report, etc.). When the element is opened, the time spent on it is counted and stored in the Project Monitoring Center. Conversely, the element that corresponds to the desired task can be automatically opened from the task list.

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

# Managing business rules

WEBDEV allows you to manage business rules. A business rule is used to define a specific operating mode or particular 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 commercial commission, a discount rate, a decay coefficient, etc.



A business rule can be simple or complex.

Business rules can come from:

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

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

PART 4

Advanced concepts

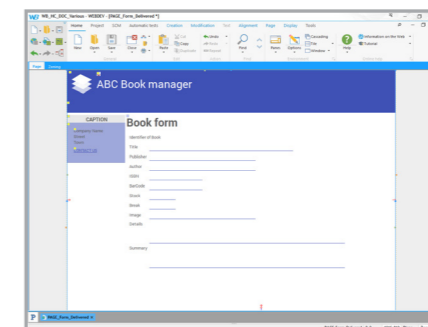
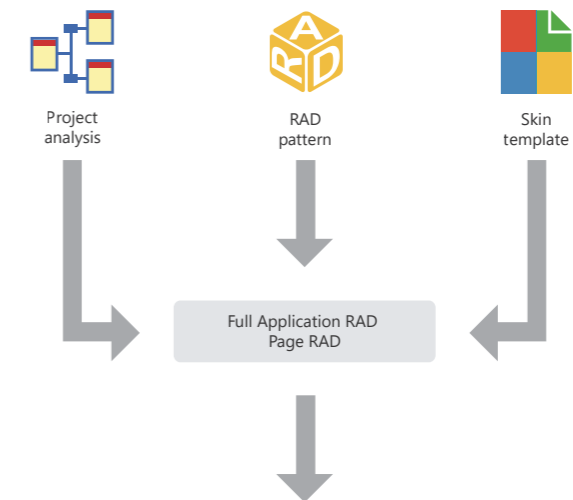
DEVELOP 10 TIMES FASTER

# RAD (Rapid Application Development)

The RAD (Rapid Application Development) is 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 necessary for their operation. These pages can be tested immediately with the data from the development computer.

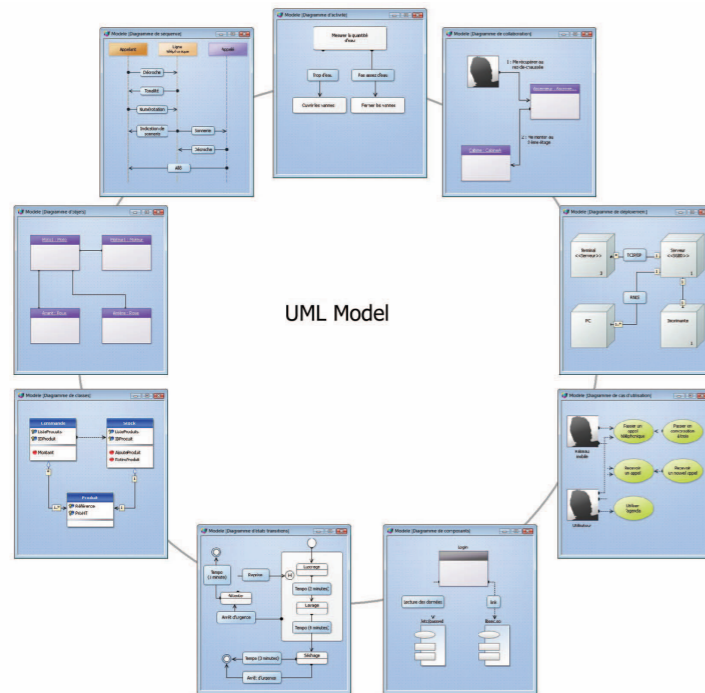


Example of page generated by RAD

# UML models

WEBDEV allows you to create nine types of UML models:

- **Class diagram:** describes the overall structure of a system.
- **Use case diagram:** represents the system features from the user's point of view.
- **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:** represents the chronological order of messages sent and received by a set of objects.
- **Collaboration diagram:** represents the structural organization of objects that send and receive messages.
- **State-transition diagram:** represents a sequence of states.
- **Deployment diagram:** shows the distribution of devices (nodes) used in a system as well as the association between the executable programs and these devices.



## UML models in practice

### 1 Overview

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

Modeling an application defines the structure and the expected behavior of a system. It helps understand its organization and detect simplification and reuse opportunities as well as manage potential risks.

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

A diagram is the graphical representation of a set of elements that build a system. To view the system from different perspectives, the Unified Modeling Language (UML) defines nine different diagrams. Each diagram represents a system state.

WINDEV, WEBDEV and WINDEV Mobile allow you to create the following nine types of UML models:

- 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. For more information about UML, see the specific documentation.

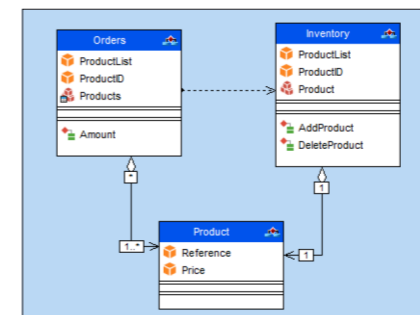
### 2 Types of UML diagrams

#### 2.1 Class diagram

A class diagram describes the structure of a system via classes and the relationships among them.

Class diagrams are the most common diagrams in object-oriented systems modeling.

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



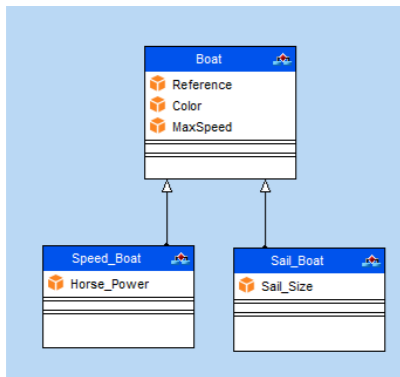
A class diagram includes the following elements:

- **Class:** represents the application structures. Each class is divided into three compartments:
  - the name of the class indicates what the class is, not what it does.
  - the attributes of the class determine the class characteristics.
  - the class operations represent the actions the class can execute.

For example, the Inventory class contains the ProductList attribute. This class also groups the AddProduct and DeleteProduct operations. These operations can be applied to the class instances. **Remark:** UML provides three visibility levels for attributes and operations:

- **Public:** the element is visible to all other classes.
- **Protected:** the element is visible to the class and its subclasses.
- **Private:** the element is visible to the class only.

- **Relationship:** describes how classes interact with each other. There are three types of relationships:
  - **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 contain at least one product.
  - **Dependency:** Use relationship that establishes that the instances of a class are linked to the instances of another element. For example, the Orders class uses the Inventory class: before adding a product to an order, you must make sure the product is available in stock.
  - **Generalization:** Relationship between a general class (parent) and a specific class (child) that derives from it. For example, the Sail\_Boat and Speed\_Boat classes are derived from the Boat class.



- **Package:** divide and organize the diagram representation (in the same way that directories organize files). Each package can contain classes and relationships.

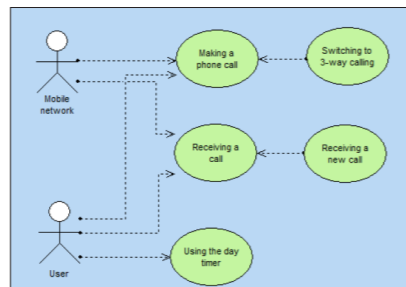
The generation of a class diagram allows you to create the structure of the WINDEV, WEBDEV and WINDEV Mobile classes used in your application.

### 2.2 Use case diagram

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

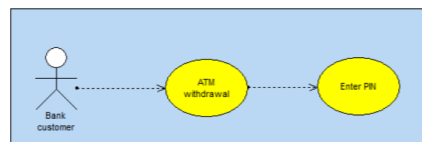
- the user can understand how to use each element.
- the developer can implement these elements.

For example, the behavior of a cell phone can be described via a use case diagram.



A use case diagram includes the following elements:

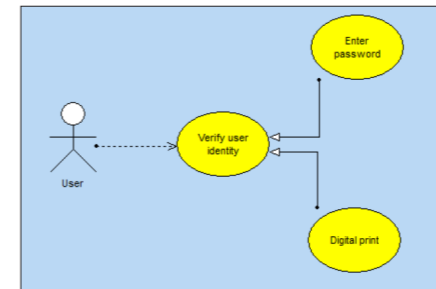
- **Actor:** represents the role of the users that interact with the application. For example, a person who works in a bank as a loan manager. If they have an account in the same bank, they will also play the role of customer.
- **Use case:** describes a sequence of actions performed by the application. For example, place an order, enter an invoice, create a new customer entry, etc, ...  
A use case describes the actions performed by an application but it does not specify how the application performs these actions.
- **Relationship:** describes the behavior of an actor with a use case. There are three types of relationships:
  - **Association:** Structural relationship between two linked elements.
  - **Dependency:** Indicates that one element uses or depends on another element. For example, a bank customer may get cash from an ATM. In this case, the Get Cash action depends on the Customer.  
To withdraw money, the Customer must enter a PIN number. In this case, the Get Cash action depends on the Password.



- **Generalization:** Relationship in which elements are organized based on a hierarchy.

For example:

- the Customer actor can be of two types: Individual customer or Company customer.
- there are two ways to verify the user's identity: a password or a fingerprint.



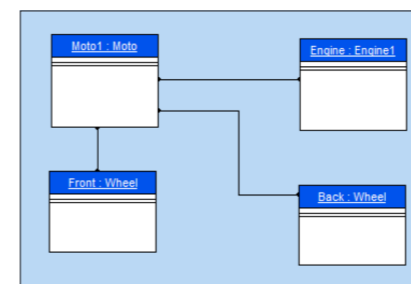
- **Package:** divides and organizes the diagram representation (in the same way that directories organize files). Each package can contain actors and use cases.

### 2.3 Object diagram

An object diagram represents a set of objects and how they relate to each other at a certain moment in 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 section of the general structure of motorcycles:



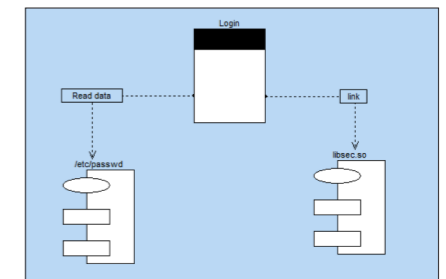
An object diagram includes the following elements:

- **object:** represents a class instance.  
Remark: If a class diagram is opened, you can create an object from a class in this diagram (Drag and drop it from the "UML analysis" pane).

- **composite object:** visually represents an object made of other objects. For example: a window that contains scrollbars, buttons, etc.
- **link:** represents the relationship between different objects.

### 2.4 Component diagram

A component diagram describes the physical and static architecture of a software application. For example: source files, libraries, executables, etc.  
For example, the diagram below presents the steps of a program that allows users to log in in text mode in Unix.

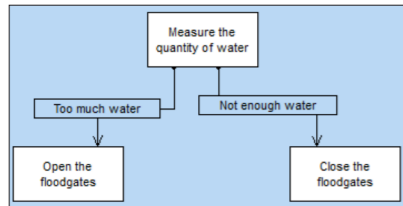


A component diagram includes the following elements:

- **module:** represents the different physical elements that make up a software application. For example: a file, a library, etc, ...  
A module can be represented:
  - by a **specification** that shows the module interface. This specification can be **generic** for classes that can be configured.
  - by its **body** that presents the module implementation.
- **task:** represents a component that has its own control flow (thread).
- **main programs** of the software application.
- **subprograms:** groups the procedures and functions that do not belong to any class.

### 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:

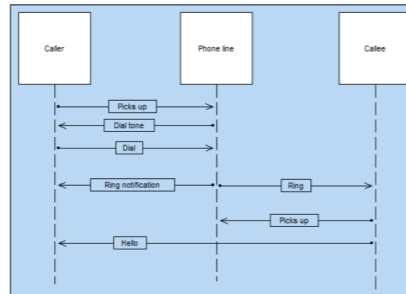


An activity diagram includes the following elements:

- **activity**: represents a specific step in a workflow. For example: "Print an estimate", "Open the window", etc.
- **synchronization bar**: synchronizes different activities:
  - by indicating the activities that must be performed before a given activity. For example: "Press clutch" and "Change gear" before "Release clutch".
  - by indicating the activities that will happen in parallel.
- **object**: relates activities to the object that performs them. For example, the "Order" and "Pay" activities are related to the "Customer" object; the "Teach" and "Evaluate" activities are related to the "Teacher" object.
- **send signal**: represents a signal sent to an object.
- **accept event**: represents the acceptance of an event coming from an object.
- **transition**: represents the transition from one completed activity to another. For example: "Too much water", "Enough money", etc.

### 2.6 Sequence diagram

A sequence diagram represents the chronological order of messages sent and received by a set of objects. For example, the following diagram represents the beginning of a phone call:

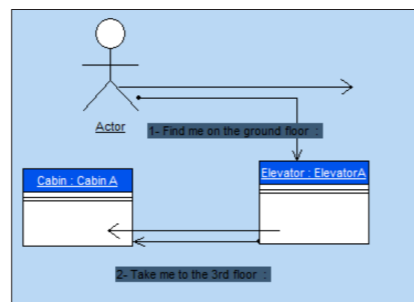


A sequence diagram includes the following elements:

- **object**: represents the different objects used. Each object is represented by a square at the top of a dotted line. This line represents the object lifespan. For example: "Caller", "Callee", etc.
- **activation period of an object**: activation periods can be inserted into the object lifeline. These periods indicate when the object is active.
- **message**: represents, via horizontal arrows, the message exchanged between the different objects. These arrows are oriented from the sender 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: "Picks up", "Ring", etc.

### 2.7 Collaboration diagram

A collaboration diagram presents the structural organization of objects that send and receive messages. For example, the diagram below represents a person that uses an elevator:



A collaboration diagram includes the following elements:

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

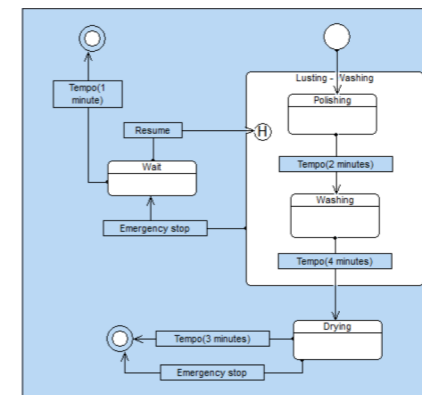
### 2.8 State-transition diagram

A state-transition diagram presents a sequence of states that an object goes through during its lifecycle. It describes how the states of an object or component change.

A state is defined by its duration and stability. A transition represents the change from one state to another. 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 includes the following elements:

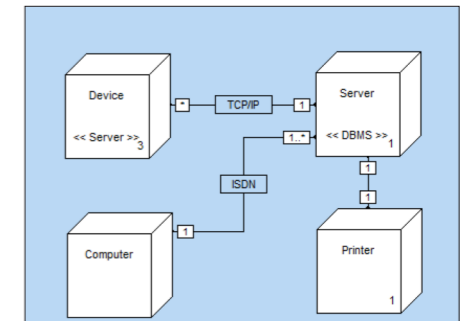
- **state**: represents the value of the object attributes at a given time.
- **initial state**: represents the state when the system is started.
- **final state**: represents the state of the system at the end of the operation.

- **superstate**: used to structure the diagram by specifying several distinction levels between the states.
- **history**: represents the last active state of a superstate (or composite state).
- **entry/exit points**: represent the states in a superstate. This allows you to link these states to other states that do not belong to the superstate.
- **transition**: represents the change from one state to another.

### 2.9 Deployment diagram

A deployment diagram shows the physical arrangement of devices (hardware/nodes) in a system as well as the relationships between the executables and these devices.

For example, the diagram below represents the hardware used in a company:



A deployment diagram includes the following elements:

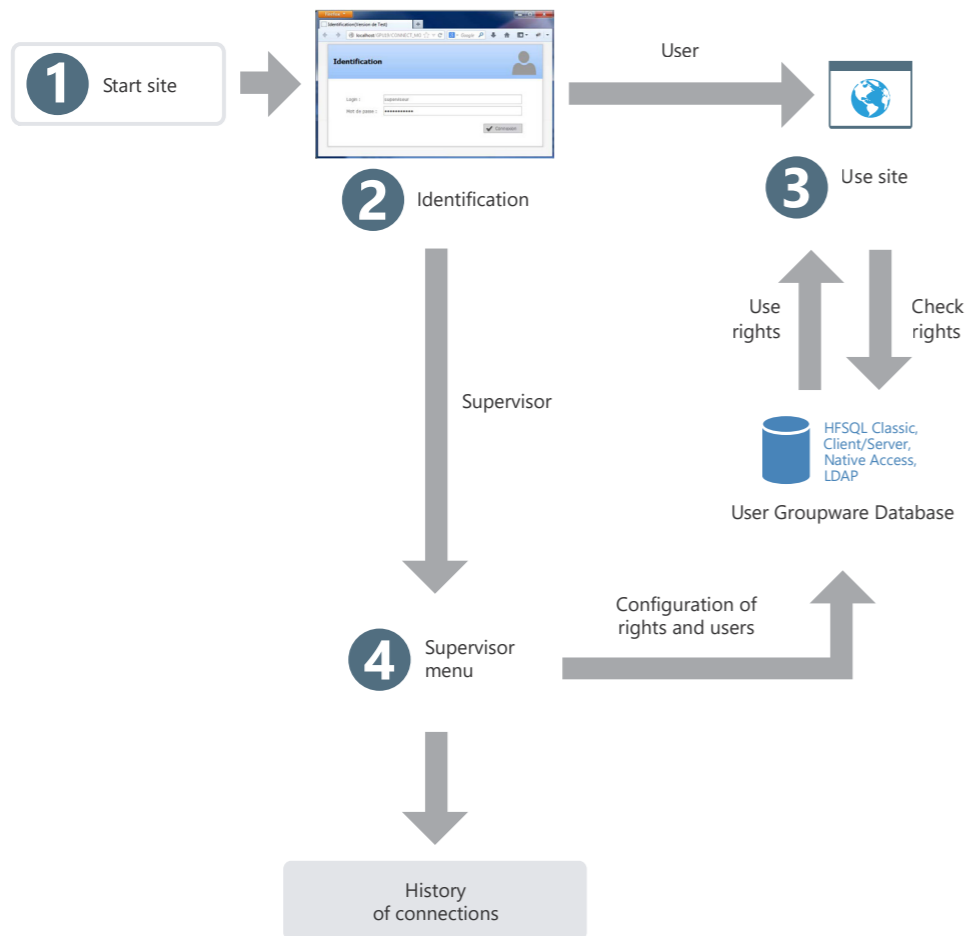
- **node class**: represents a class of a physical resource. For example: server, PC, printer, etc.
- **node instance**: represents a physical resource. For example: server 3, printer 70, etc.
- **connection**: depicts the communication path between two nodes. For example: ISDN or TCP/IP link.

# User Groupware

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

There are two types of users:

- simple user, who directly uses the site.
- supervisor: configures the users and their rights.



## User Groupware in practice

### 1 Overview

In an Intranet or Internet site, it is necessary to define the role of the different contributors. Several access levels can be defined according to the user.

All users have different responsibilities and requirements. Their field of action must therefore be more or less limited.

Let's take a simple example: a sales management application that offers the following features:

- View prices,
- Modify prices,
- Enter orders,
- Enter customers.

Access may vary according to the user. For example:

- administrative assistants can see the price list and create orders.

- sales representatives can see the price list, place orders and create new customers.
- sales directors have access to all options.

To manage these access levels in your WEBDEV sites, all you have to do is include the User Groupware in your site. In just a few clicks, a standard application can be transformed 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 username and password) and give them access to some of the site features.

**Remark:** The User Groupware is available in dynamic WEBDEV sites running on Windows or Linux. The User Groupware is not available in static, PHP, AWP or Ajax sites.

### 2 How does User Groupware work?

A site that uses the User Groupware has two levels of use:

- user level.
- supervisor level.

#### 2.1 User level

The user logs in to the application via a login page and only has access to some features.

**Remark:** You can also log in using an LDAP directory.

#### 2.2 Supervisor level

The supervisor logs in to the application via a login page and accesses an advanced menu to configure users and their rights, or to start the application.

The advanced menu allows the supervisor to:

- start the site.
- configure the groupware.

Configuring the groupware consists in creating and associating users and groups, and managing the rights granted to each user (or group) on each page of the site. Rights can be defined for menu options, groups of controls and controls. The supervisor can gray out these elements, make them inactive or invisible. These configurations take precedence over any configuration defined in the program.

For more details on how the supervisor can configure the groupware, see "Configuring a site that uses the standard user groupware", page 131.

**Remark:** All user information and rights are stored in HFSQL data files.

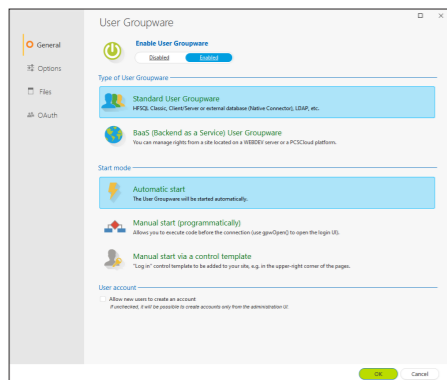


### 3 Implementing the User Groupware

#### 3.1 Adding the User Groupware to a site

To implement the User Groupware in a WEBDEV site:

1. On the "Project" tab, in the "Project" group, click "User Groupware".
2. In the description window of the current project configuration, in "User Groupware: Disabled", click "Change".
3. The User Groupware settings window allows you to enable the User Groupware and define:



- The type and start mode of the User Groupware.
- The type of customization.
- The type of database and the location of the User Groupware data files (Standard User Groupware).
- The user authentication mode (Standard User Groupware).
- The use of a specific connection service (Standard User Groupware).
- The BaaS mode settings.

#### Type of User Groupware

There are various types of groupware:

- **Standard User Groupware:** The User Groupware will be integrated in its "Standard" version: the User Groupware will be associated only to your site, and the rights will be defined from the site. If this option is selected, two specific tabs are displayed:
  - the "Files" tab, which allows you to specify the type of database and authentication used.
  - the "OAuth" tab, which allows you to specify the connection services available to users.

- **BaaS User Groupware:** This option allows you to integrate a BaaS (Backend as a Service) User Groupware and manage rights via a server. If this option is selected, two specific tabs are displayed:
  - the "BaaS" tab, which allows you specify the characteristics of the BaaS server, the user accounts and the URLs of the administration sites.
  - the "Email" tab, which allows you to customize the messages to be sent to users (registration validation or password reset emails).

#### BaaS User Groupware: Required configuration:

Two cases may occur:

- **If you are using a PCSCloud platform:** the required elements are installed by default on your platform. Reminder: PCSCloud is a paid, turnkey solution. For more details, see PCSCloud.
- **If you are not using a PCSCloud platform:** install the following elements:
  - an HFSQL server,
  - the reserved server, including the modules for the Mobile User Groupware (WDBaaS).
 For more details, see Configuring a BaaS server (User Groupware and *Private* Store).

#### Starting the User Groupware

Several start modes are available:

- The "Automatic start" option indicates that the User Groupware is started at the same time as the site. The User Groupware login page is started before any other page of the site. The project initialization code is executed once the login page is opened and validated.
- The "Manual start (programmatically)" option allows you to define when the User Groupware should be started. The login page will only be opened if `gpwOpen` (or `gpwLogin`) is used. This option can be used to execute the project initialization code before opening the login page. This option is useful for a multilingual project, for example: a page can be displayed to choose the language of the application.

- The "Manual start via a control template" option is used to integrate the User Groupware login page in the site. This allows you to work with the automatic User Groupware (and to benefit from its updates) while being able to customize the login page. If this option is selected, the `WDGPU_WB_CNX` internal component is added to your project. This component includes the "TPLC\_GPU\_Connection" control template that manages the connection. To add this control template, simply drag and drop it onto the desired pages (or templates). Once it has been added, you can customize it:
  - by overloading the controls and/or the processes for small changes,
  - by directly modifying the control template for more important changes.

#### 3.2 User Groupware options (standard and BaaS mode)

There are multiple User Groupware customization modes:

- **No customization:** all User Groupware programming is automatically integrated into your site. No customization can be done. The project skin is automatically applied to all pages used by the User Groupware. This integration mode automatically follows future changes in the User Groupware.
- **Customize login UI (log in, register):** all User Groupware programming is automatically integrated into your site. The login UI can be modified. The `WDGPU_WB` internal component is added to the project. This component contains all the pages, page templates, control templates and procedures necessary for user authentication. This allows you to fully customize all the login pages used by the User Groupware. This integration mode automatically follows future changes in the User Groupware.
- **Full customization** (only in Standard User Groupware mode): the entire programming of the User Groupware is automatically included in your site via an internal component. The `WDGPU_WB` internal component is added to the project. This component contains an analysis as well as all the pages, queries, classes and procedures required to manage the User Groupware. This allows you to customize all the pages used by the User Groupware.

The "Allow opening windows, pages and reports by default" option allows all users to open windows, pages and reports by default. In this case, simply specify which elements will not be accessible when configuring the rights.

The option "Automatic login for test mode" allows you to specify the username and password to be used when testing a page or project (in GO mode or in "Automated test" mode). This information will be used if an automated test is run on the project.

By default:

- data files are in English. To use data in French, simply check "Data files in French".
- User Groupware data files have a specific password. This password is: "PCSGPW2001". To change this password, enter the new password in the "Options" tab of the User Groupware description window. Characters are shown as bullets. Remark: This password is used when re-indexing or opening data files with WDMAP for example.

For more details, see the online help.

#### 3.3 Options specific to the standard User Groupware

If you use the standard User Groupware, the "Files" tab allows you to configure:

- the type of database used (these options are available in "Automatic start" mode only).
- the type of authentication:
  - Authentication managed by the WEBDEV User Groupware.
  - Authentication from an LDAP directory.

#### Database type

In "Standard User Groupware" mode, the User Groupware uses several specific data files. These data files are mainly used to manage users, groups, rights and elements in the site. The standard User Groupware can be used:

- with HFSQL Classic data files. In this case, you can specify the location of the data files via the "Default directory" link.
- with HFSQL Client/Server data files. In this case, you can define the parameters for accessing the HFSQL Client/Server database as well as the file directory by selecting "Click here to enter the server parameters".

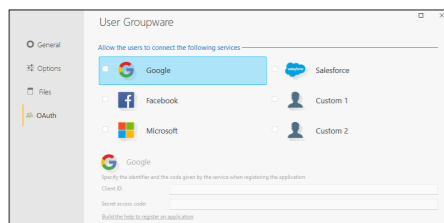
### Type of authentication

The standard User Groupware includes several user authentication modes:

- **Authentication entirely managed by the User Groupware:**  
The username and password are saved in the User Groupware data files.
- **Authentication from an LDAP directory or Active Directory:**  
If the company of your customer uses an LDAP directory, you can enable the User Groupware on this directory. Simply click the link "Click here to enter the server parameters" to specify the characteristics of the LDAP directory (server, port, base DN, etc.). Therefore, the account and the password in the LDAP directory will be automatically requested to the user. In this mode, an additional option is available: **Login window only (no management of rights)**.
  - If this option is checked, no groupware data file will be created. If the application is started in automatic mode, the user will need to log in. If the user is saved in the LDAP directory, the application will start; otherwise, it will be closed. A supervisor cannot configure the rights on the windows.
  - If this option is not checked, only the users in the LDAP directory will be able to log in. You can set the rights on the windows.

For more details, see the online help.

The "OAuth" tab allows you to authorize the use of a specific connection service. For the specified service, it is necessary to specify the corresponding identifiers (provided when registering the application with the selected service).



### 3.4 Options specific to the BaaS User Groupware

When using a BaaS User Groupware, the "BaaS" and "Email" tabs allow defining specific options.

The "BaaS" tab allows you to define:

- the location of the User Groupware application server. You can select a PCSCloud platform or a Network or Internet server,
- the user account management mode,
- the URLs of the administration sites.

The "Email" tab allows you to write the messages for registration confirmation or password recovery.

### 3.5 Testing the site (Standard User Groupware)

When a site that uses the standard User Groupware is tested, the first page that appears is the login page (regardless of the first page defined in your site).

A single user is created by default: the supervisor. To log in as supervisor, simply enter the following information in the login page:

- Name: SUPERVISOR
- Password: SUPERVISOR

You can test your site, or configure the User Groupware.

**Remarks:**

- To set a first page other than the login page in your site, select "Manual start (programmatically)" in the "General" tab of the User Groupware options. Simply use **gpwOpen** to open the login page.
- The User Groupware management pages are executed before the project initialization code.
- The first page of your site defined in the project will be started after the login page (when the user is not the supervisor).
- **To avoid starting the User Groupware when testing the site**, select "Disabled" in the "General" tab of the User Groupware options. If you re-enable the User Groupware again, the data files previously used by the User Groupware will not be deleted.

### 3.6 Installing a site that uses the standard User Groupware

Sites that use the standard User Groupware are installed in the same way as any other site : on the "Project" tab, in the "Generation" group, simply click "Deploy site".

A screen specific to the standard user groupware allows you to set the location of user groupware data files.

**Remarks:**

- If the User Groupware data files have been configured for your client, they must be selected when preparing the setup. Therefore, the list of setup files must be customized.
- If the groupware data files have not been configured, the Supervisor will be the only user in the site the first time it is started.

**Tip:** if you do not configure the different User Groupware use levels for your clients, we recommend that you provide them with a document listing all the controls used in your pages in order to get an optimized configuration.

## 4 Configuring a site that uses the standard User Groupware

When the site is deployed, the standard User Groupware is configured by the site supervisor.

This configuration consists in creating users and groups of users, and granting them specific rights for each control in each page of the site.

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 log in as supervisor:
  - Name: SUPERVISOR
  - Password: SUPERVISOR
2. Select "Configure groupware".
3. The User Groupware management page appears. This page is used to:
  - manage users and groups.
  - manage rights.
  - display statistics.
  - migrate data coming from a User Groupware version 18 and earlier.

### 4.1 Managing users

The user management consists in:

- creating users,
- creating groups,
- associating users with groups.

To create a new user, you must specify:

- the user's last name (mandatory)
- the user's first name
- the username. This username is the identifier used to log in to the application.

- the user's password. This password is optional and can be entered by the user the first time they log in (check the corresponding option).

The user can be defined as supervisor of the site. You can also modify or delete users. When you delete a user, you can delete the entire user or their rights only.

**Users can be grouped.**

A user can be associated with several groups.

**Remarks:**

- It is recommended to change the supervisor's password the first time you use the application.
- If you use an LDAP directory, you can import the users from the directory to manage their rights.

### 4.2 Managing rights

The supervisor can manage the rights granted to each user (or group of users) on the elements of the project pages.

A specific state can be defined for all the elements of a page associated with a user.

The following elements can be handled in pages, internal pages and page templates:

- controls, groups of controls,
- menu options.

The following states are available for each element:

- **Default:** the element behavior corresponds to the default behavior, defined in the application.
- **Inactive:** 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.

### 4.3 Displaying statistics

You can get site usage statistics by user for a specific period.

### 4.4 Migrating data

From version 19, the User Groupware has been optimized. The data files used by the User Groupware have changed.

The User Groupware administrator can migrate the existing data to the new format and therefore automatically retrieve the data used in the earlier versions. This migration can be performed regardless of the data format: HFSQL Classic, HFSQL Client/Server or Native Connector.

## 5 Tips for a site that uses the User Groupware

### 5.1 Using groups of controls

To simplify the configuration of the groupware management according to the users, it is recommended to use groups of controls.

You can create groups of controls in your pages according to the controls to be displayed for a specific 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 of the different elements in your pages:

- in the element description (7-tab window),
- through programming (**State** or **Visible** properties).

When configuring the User Groupware, the supervisor can define different visibility options. The options defined by the supervisor have priority.

For example, a button is used to make a control active. This control was grayed out by the supervisor. Your code will be ignored and the control will not be active.

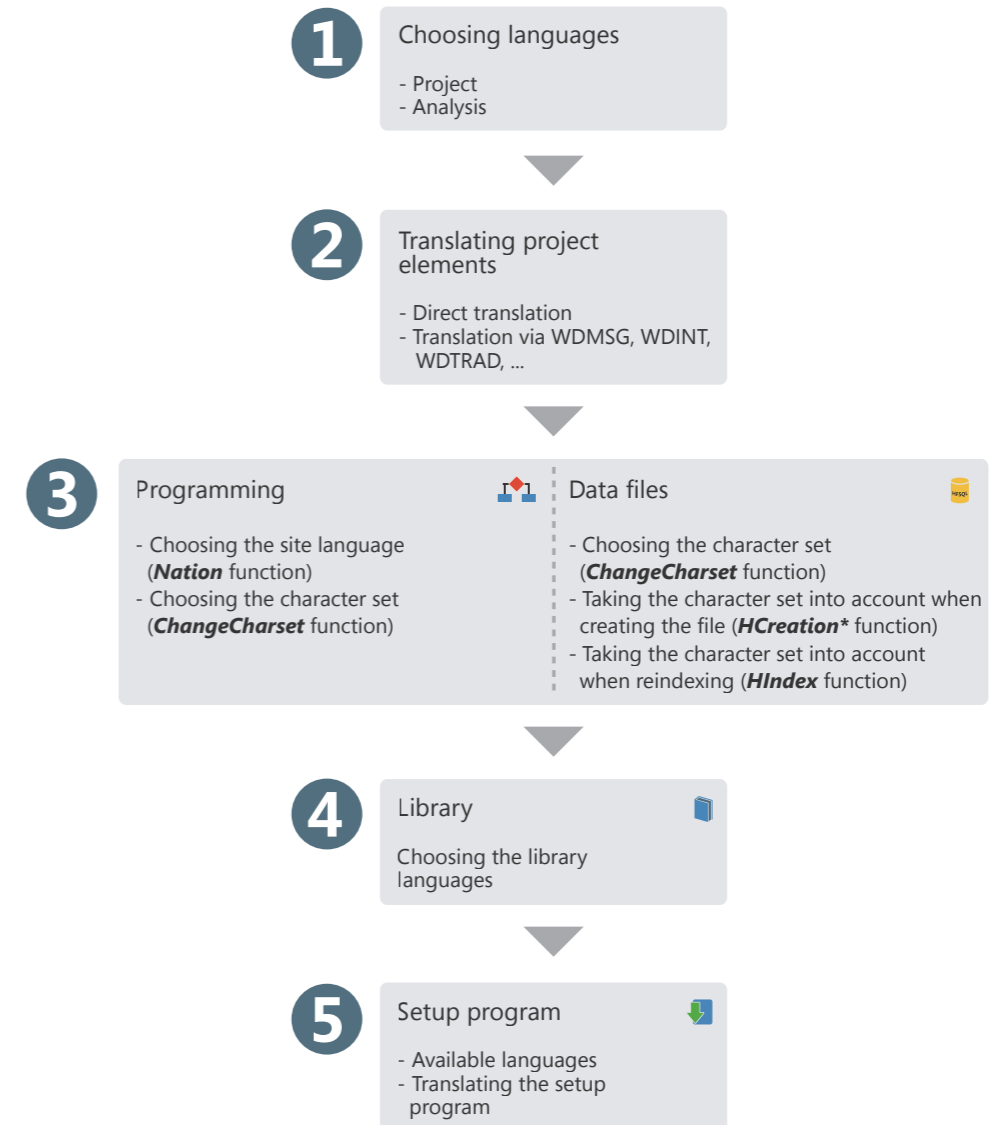
### 5.3 Defining rights

To define the rights corresponding to the features of the site, it is recommended to:

- set the rights according to the site manager's requirements before creating the setup procedure . Simply add the groupware data files when you create the site setup.
- provide a technical documentation that includes the names of controls, groups of controls and options that must be configured according to the usage level defined for the site.

# Multilingual sites

WEBDEV allows you to easily create 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 project elements (pages, code, etc.) in the different project languages.
3. Defining the project language programmatically.
4. Managing specific character sets in the data files.
5. Creating the library and the setup program.

### Remarks:

- If the operating system supports several languages (Hebrew, Arabic, Greek, etc.), the corresponding character set will be automatically used when translating to these languages.
- If your site is multilingual, this feature must also be managed in the User Groupware pages and in the HFSQL automatic error handling pages. For more details, see the online help.
- HFSQL data files and page controls support the Unicode standard.

## 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:

1. On the "Project" tab, in the "Project" group, click "Description". The project description window appears.
2. In the "Languages" tab, you can add or delete the languages supported by the project.

The selected languages will be available for all multilingual resources that can be translated (captions of controls, menu options, help messages associated with a control, etc).

When changing the project language, changes will be automatically taken into account:

- for new elements or objects created in the WEBDEV editor,
- for all elements or objects opened in the WEBDEV editor.

The main language corresponds to the language used by default at runtime.

### Language options

The "Languages" tab of the project description also allows you to define all language-specific options: number, currency, date, etc.

Windows language options are used by default (these options can be found in the Windows control panel).

**When creating an Edit control or a column of a Table control** that displays numeric data (numbers, currencies, dates, times, durations, etc.), the input mask defined in the project language options will be automatically used. This option is available in pages.

**At runtime**, when an Edit control or a column in a Table control has a mask "Defined by the project", the input/display mask will automatically adapt according to the options selected in the project for the language displayed at runtime.

**Remark:** The language options also allow you to define the writing direction and the character set used ("Miscellaneous" option).

## 2.2 Languages supported by the analysis

If your project uses an analysis, the languages supported by the analysis are selected in the data model editor. The same analysis can be shared between different projects that do not use the same languages. Therefore, the analysis might use more languages than the project.

To configure the languages supported by the analysis:

1. Open the project analysis in the editor.
2. Open the analysis description window: on the "Analysis" tab, in the "Current element" group, click "Description".
3. Select the "International" tab.

The different languages set in the analysis will be available:

- when configuring the shared information of the items. The description of the controls bound to the items (options, captions, etc.) can be entered

in the different languages supported by the analysis. When generating a "full RAD" or a page RAD, this information will be automatically taken into account for all languages common to the analysis and the project.

- for the information printed in the technical documentation (notes of data files or items).

### 2.3 Languages supported by the different project elements

By default, the different elements of the project (pages, reports, code, classes, etc.) support the same languages as the project in which they were created.

In some cases, an element may support more languages than the project (when the element is shared between several projects that support different languages for example).

## 3 Translating the interface into multiple languages

When 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, some elements must be taken into account:

- The selected translation mode.
- The supported languages.
- The use of specific pages (User Groupware, automatic management of HFSQL errors).
- The messages displayed through programming.

### 3.1 Translation method

WEBDEV proposes several translation modes:

- Automatic translation of the UI via a specific translation tool, installed on the development computer. This translation is performed in the WEBDEV editor.
- Export all the messages via WDMMSG and reintegrate them once they have been translated.

#### Translating the UI

For each element, multiple multilingual areas are displayed in the different tabs of the description window. These multilingual areas allow you to enter information in the different languages supported by the project.

To translate this information from WEBDEV:

1. Open the WEBDEV options to configure the translation options of the software used: on the "Home" tab, in the "Environment" group, expand "Options" and select "General options of WEBDEV".
2. In the "Translation" tab, choose:
  - the default translation tool.
  - the source and destination languages for the translation.
3. If a translation tool is specified, a "Translate" button will be displayed in each element containing information to translate. This button is used to translate the selected text via the specified translation tool.
4. Select the text to be translated and click "Translate".

#### Exporting and reintegrating the information to translate

WDMMSG is a tool used to extract all the messages of the project (control captions, etc.) and reintegrate them 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 specific character set

If your site supports languages that use specific character sets (Greek, Russian, etc.), you must enter the translation of the different messages using these specific character sets.

WEBDEV allows you to automatically manage the use of specific character sets in the editor.

When the text cursor is located in the input area of a language that uses a specific character set, the corresponding input language (character set used by the keyboard) is automatically enabled.

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, the files corresponding to the desired character sets must be installed in the Windows region settings (control panel).

### 3.3 Translating specific pages

#### Handling HFSQL errors

By default, HFSQL error management pages are provided in English and French. To translate them into another language, you must:

1. Include the default error pages in your project. These pages are provided as example in \Programs\Data\Preset pages\HFSQL - Automatic help pages.
2. Customize the management of errors to use the HFSQL pages for error management (HOnError).
3. Translate the different messages (see previous paragraph).

#### User Groupware

By default, the User Groupware is available in English and French.

## 4 Choosing the language through programming

A multilingual site is an application that can be distributed in several languages. The user will be able to choose the language used by the application at runtime. You can for example:

- prompt the user to choose the runtime language the first time the application is started.

To translate a site that uses the User Groupware, select the "Manual start" mode when integrating the User Groupware into the site. Reminder: To configure the User Groupware, go to the "Project" tab, "Project" group and click "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.


### 3.4 Translating WLanguage code messages

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 typed in the code editor:

1. Select the character string to be translated.
2. Select "Translate" in the context menu of the selection. The window for translating messages is displayed.
3. Type the translation and validate. The line of code is displayed in the code editor:

```
Info("Develop 10 times faster")
```

- The  icon indicates that this string has been translated to other languages. Click the icon to open the translation window.
- To see the existing translations, simply hover over the icon.

**Remark:** To convert all the messages in your code into multilingual messages, go to the "Code" tab, "Languages" group, expand "Translate strings" and select "Convert simple strings into multilingual messages".

- implement an option (menu option or button for example) allowing the user to change the current language of the application.

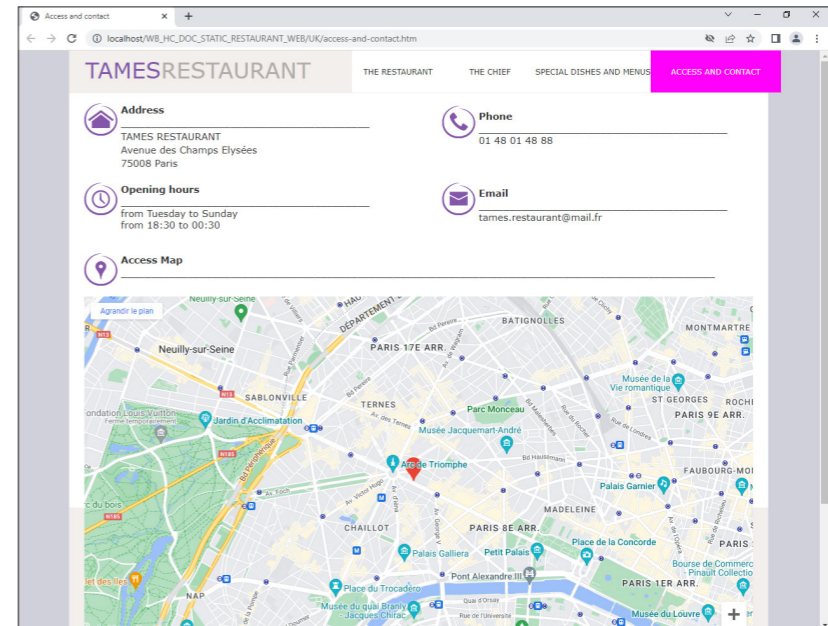
# 10 tips about ergonomics ...

## 1 Enhance the style of your 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). You should pay special attention to its content and presentation. You can read more documentation on how to create user-friendly websites.

## 2 Get in touch

When users visit a site, they must be able to contact the person or the company in charge of the site: provide the information that will allow users to contact you: name, mailing address, email address, etc.



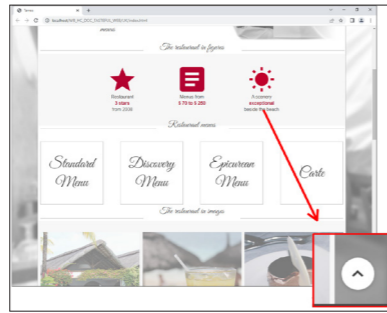
This page shows a map with the location of the headquarters or store, opening times, etc.

### 3 Are your pages very long? Give the Web user ways to go back to the top.

If the content of your pages is too long, 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 button to go back to the top of the page

### 4 Harmonize the style and alignment of your buttons.

Make sure your buttons use the same style, and check their alignment. The page editor includes a snap-on and alignment mechanism to simplify this task.

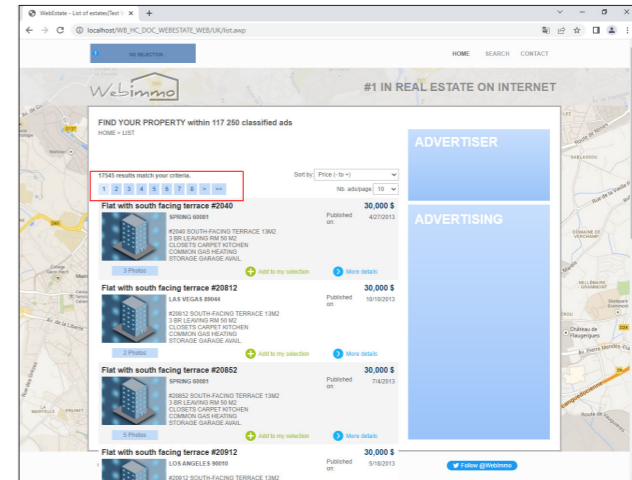


### 5 Avoid using of framesets.

Framesets, even though they seem to be more efficient at first, have specific problems: they include several pages. They are not as SEO-friendly. It is also more difficult for the users to navigate through your website (especially with the back button). Today, it is not recommended to use framesets when developing sites.

### 6 Are there lists containing more than 20 elements? Use several pages to list the elements.

You can use a Pager control to navigate between pages without writing code. This will allow for a cleaner interface. Your website will be faster and more user-friendly.



### 7 Use images displayed at 100% or Homothetic centered.

To keep the proportions of your images, it is recommended to choose:

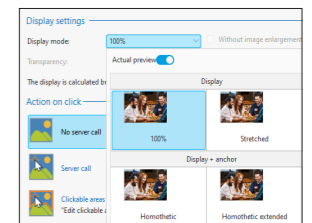
- images at 100% (for static and dynamic images or if the image is the same size as the control).
- images in homothetic centered mode (for generated images or images of different sizes from a database): the size of the image will adapt to the area defined for the image. The proportions will be preserved.



Image at 100 % if the control dimensions correspond to those of the image, otherwise, Homothetic image



Stretched image (not recommended)



Selecting the display mode in the editor (Image control description, "General" tab)

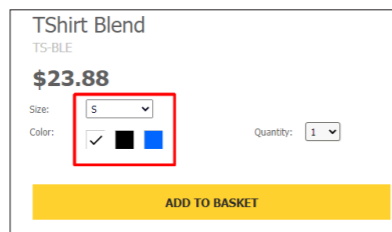
**8** Are you using images? Make sure they are displayed properly in your pages.



If a display problem occurs, check whether:

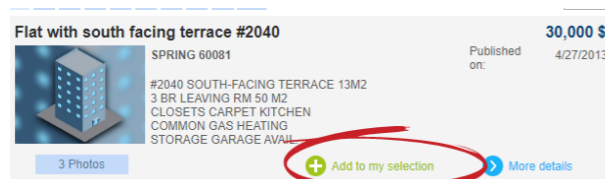
- the image exists,
- the image is saved in a format recognized by the browser (GIF or JPG, for example),
- the image is stored in the subdirectory of the <ProjectName>\_WEB project.

**9** The Web user must type a quantity? Use a combo box to simplify the input of quantities.



**10** Are you proposing a product catalog with the possibility to order?

Specify all the product details (name, reference, price, etc.), including a button or link to order the product, add it to a custom selection, add it to the cart, etc.



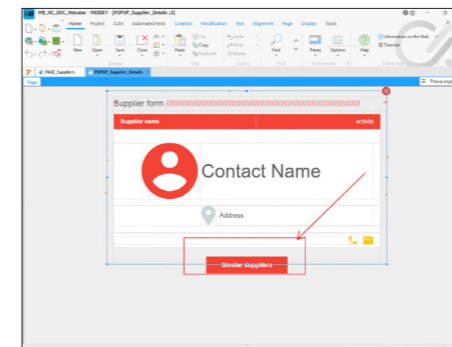
# Overlaying controls

To get special effects (transparency or depth), your controls can be overlaid.

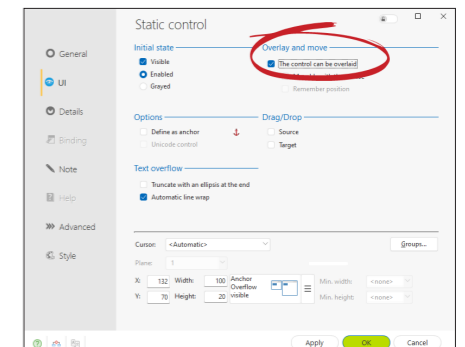
Each control has an "Overlay and move" option ("UI" tab in the control description window). If the option "The control can be overlaid" is selected, the control will no longer be linked to the page positioning tables. The control can be moved and placed "over" any other control on the page.

**Caution:** The control with the option "The control can be overlaid" will be displayed above the other controls. If this option is enabled on multiple controls, the order in which they are overlaid can be defined via the "Bring forward", "Send backward", "Bring to front" and "Send to back" options in the "Modification" tab.

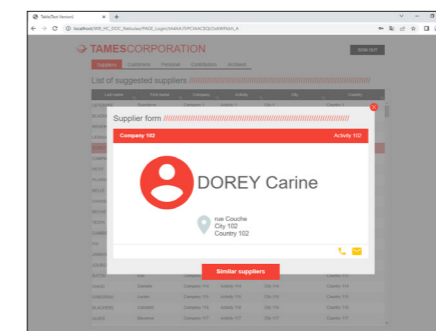
For example, to display a Static control over an Image control, the Static control must be made overlayable and be positioned over the Image control.



In the editor



Overlay option



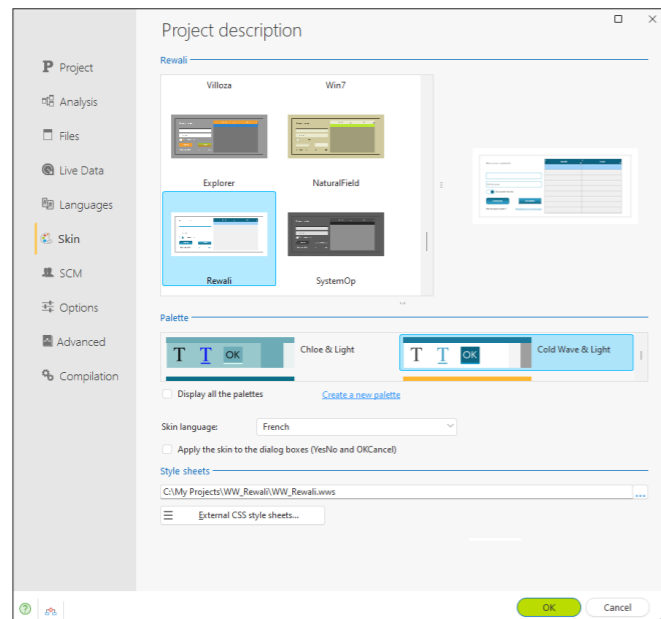
In the browser

# Customizing the aspect of a site: skin elements

The graphic aspect of a Web site is a major element. With WEBDEV, it is very easy to get a good looking site with a professional style at first try. To do so, several tools are available: the skins, the palettes and the preset page templates.

## Skin

The skin of a site can be selected in the project description window ("Project" tab, "Project" group, "Description"). Skins can be configured in the "Skin" tab. Several 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 project reports.

The skins can be swapped at any time to modify the graphic aspect of a site. When changing skin, WEBDEV automatically adapts the visual aspect of the site controls.

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 (e.g., "Promotion" pages of an e-commerce site) must use a specific skin, you can redefine the project skin as a page template. If a particular page must have a different skin, you can choose the skin only for that page.

## Color palettes

Color palettes define a set of colors that are available 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 project skin in order to easily change the visual aspect of a project.

The color palette 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 palette must be defined at project level. If a set of pages (e.g., "Management" pages of a site) must use a specific palette, you can redefine the project palette as a page template. Finally, if a specific page must have a different palette, you can choose the palette only for that page.

## Preset page templates

Preset page templates provide page structures that correspond to a wide variety of sites and promote reusability and graphic harmony between projects.

Several templates are provided with WEBDEV.

You can add the same preset template to a project multiple times to use it with different skins or color palettes.

Once it is added to the project, the preset template can be opened in the editor and modified like any other page template.

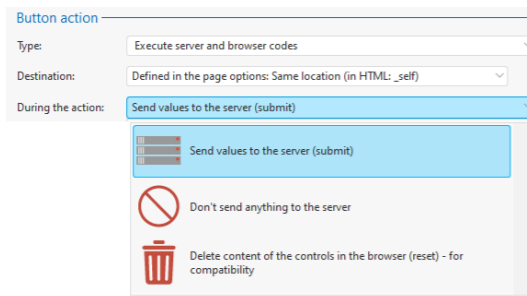
A preset page template can also be used as base template for creating another page template.



# Choosing a type of button/link

You can choose among several button/link actions, depending on the operation to be performed on the page controls.

The action can be chosen in the control description ("General" tab).



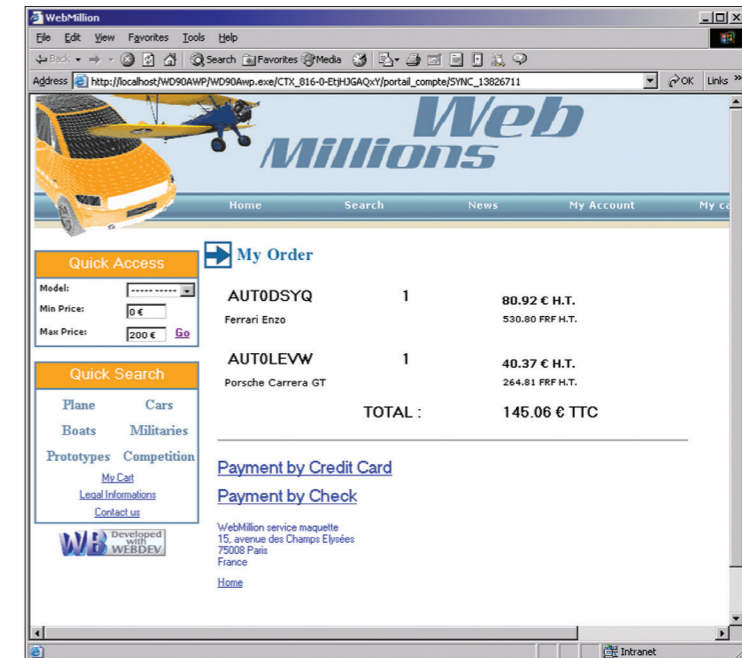
Let's see some examples of "standard" uses of buttons/links in a page:

- Send data of the current page to the server to be processed**  
 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 link with the action "Send values to the server". The page data is sent to the server in order to be processed.
- Perform a process on the server, without retrieving the data of the current page**  
 Example: Menu page with buttons: each button opens a page of the site.  
 To perform this type of process, use a button with the action "Do not send anything to the server".
- Perform a process in browser code**  
 Example: Perform a simple calculation locally, an input check, ...  
 To perform this type of process, use a button with the action "Do not send anything to the server".
- Display a page external to the WEBDEV site**  
 Example: Launch a search engine site via a button.  
 To perform this type of process, use a button with the action "Do not send anything to the server".
- Reinitialize the page controls**  
 Example: Link used to clear the controls in the current page.  
 To perform this type of process, use a link with the action "Delete content of the controls in the browser".

# Preventing from going back to a page

The browser's "Back" button allows the Web user to go back to a previously visited page. However, it may not always be a good idea to go back to a page.

For example, in a business site, if the Web user goes back to the order validation page, he may validate his order twice and therefore place 2 orders.



To prevent from using the browser "Back" button to go back to a page:

1. Display the page in the WEBDEV page editor.
2. On the "Page" tab, click the dialog box launcher of the "Edit" group ( ). The page description window appears.
3. Open the "UI" tab.
4. In the option "Browser "Back" button", select "Forbidden".
5. Validate.

# Security benefit: managing the "Back" button

A browser allows you to navigate through the different pages of a site using the "Next" and "Back" buttons.

A specific page context is created on the server whenever a new Session page is displayed in the browser.

When the same Session page is used to display different data (page with table based on a data file, loops or "Form with browse" page, etc.), the page context on the server changes according to the data displayed.

**When the 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 using a "Back" operation to go from one page to another.

A specific management is required when the same Session page displays different data.

## Why manage the "Back" button? A simple example...

To understand why the "Back" button must be managed, let's take a look at a simple example: a Session page that displays the list of suppliers.

### Step 1: Displaying the list of suppliers

The screenshot shows a web application interface for 'TAMESCORPORATION'. The 'Suppliers' menu item is selected. The page title is 'List of suggested suppliers'. Below the title is a table with the following columns: Last name, First name, Company, Activity, City, and Country. The table contains 17 rows of data, including suppliers like LEFEVRE, BLANCHERE, REVERCHON, etc.

Last name	First name	Company	Activity	City	Country
LEFEVRE	Guillaume	Company 1	Activity 1	City 1	Country 1
BLANCHERE	Emeline	Company 10	Activity 10	City 10	Country 10
REVERCHON	Jean-Baptiste	Company 100	Activity 100	City 100	Country 100
LESALMNER	Marie	Company 101	Activity 101	City 101	Country 101
DOREY	Carine	Company 102	Activity 102	City 102	Country 102
CAMPANA	Albino	Company 103	Activity 103	City 103	Country 103
PETIT	Liliane	Company 104	Activity 104	City 104	Country 104
PLASSARD	Elise	Company 105	Activity 105	City 105	Country 105
BELLE	Barthelemy	Company 106	Activity 106	City 106	Country 106
CHASSARD	Cyrille	Company 107	Activity 107	City 107	Country 107
BECHET	Eloanna	Company 108	Activity 108	City 108	Country 108
TESTA	Jeanne	Company 109	Activity 109	City 109	Country 109
GAMBIER	Adrienne	Company 11	Activity 11	City 11	Country 11
FIX	Monique	Company 110	Activity 110	City 110	Country 110
JANKOWSKI	Eloannette	Company 111	Activity 111	City 111	Country 111
JOLIBOIS	Micheline	Company 112	Activity 112	City 112	Country 112
BATON	Eloi	Company 113	Activity 113	City 113	Country 113
VAUD	Danielle	Company 114	Activity 114	City 114	Country 114
GABORIAU	Lucien	Company 115	Activity 115	City 115	Country 115
BLANCHERE	Constant	Company 116	Activity 116	City 116	Country 116
ALVES	Maurice	Company 117	Activity 117	City 117	Country 117

The context contains the list of suppliers displayed in the Table control.

### Step 2: Clicking the "Customers" link.

The screenshot shows the same web application interface, but the 'Customers' menu item is selected. The page title is 'List of registered customers'. The table structure is identical to the previous screenshot, but the data represents a list of customers.

Last name	First name	Company	Activity	City	Country
LEFEVRE	Guillaume	Company 1	Activity 1	City 1	Country 1
BLANCHERE	Emeline	Company 10	Activity 10	City 10	Country 10
REVERCHON	Jean-Baptiste	Company 100	Activity 100	City 100	Country 100
LESALMNER	Marie	Company 101	Activity 101	City 101	Country 101
DOREY	Carine	Company 102	Activity 102	City 102	Country 102
CAMPANA	Albino	Company 103	Activity 103	City 103	Country 103
PETIT	Liliane	Company 104	Activity 104	City 104	Country 104
PLASSARD	Elise	Company 105	Activity 105	City 105	Country 105
BELLE	Barthelemy	Company 106	Activity 106	City 106	Country 106
CHASSARD	Cyrille	Company 107	Activity 107	City 107	Country 107
BECHET	Eloanna	Company 108	Activity 108	City 108	Country 108
TESTA	Jeanne	Company 109	Activity 109	City 109	Country 109
GAMBIER	Adrienne	Company 11	Activity 11	City 11	Country 11
FIX	Monique	Company 110	Activity 110	City 110	Country 110
JANKOWSKI	Eloannette	Company 111	Activity 111	City 111	Country 111
JOLIBOIS	Micheline	Company 112	Activity 112	City 112	Country 112
BATON	Eloi	Company 113	Activity 113	City 113	Country 113
VAUD	Danielle	Company 114	Activity 114	City 114	Country 114
GABORIAU	Lucien	Company 115	Activity 115	City 115	Country 115
BLANCHERE	Constant	Company 116	Activity 116	City 116	Country 116
ALVES	Maurice	Company 117	Activity 117	City 117	Country 117

The customers are displayed. The page context is updated on the server: the context contains the list of elements displayed in this Table control (list of customers).

### Step 3: Clicking the browser "Back" button.

The screenshot shows the same web application interface as in Step 1, with the 'Suppliers' menu item selected and the 'List of suggested suppliers' table displayed. This represents the state of the browser after clicking the 'Back' button, where the page content is restored but the server context remains that of the customers page.

Last name	First name	Company	Activity	City	Country
LEFEVRE	Guillaume	Company 1	Activity 1	City 1	Country 1
BLANCHERE	Emeline	Company 10	Activity 10	City 10	Country 10
REVERCHON	Jean-Baptiste	Company 100	Activity 100	City 100	Country 100
LESALMNER	Marie	Company 101	Activity 101	City 101	Country 101
DOREY	Carine	Company 102	Activity 102	City 102	Country 102
CAMPANA	Albino	Company 103	Activity 103	City 103	Country 103
PETIT	Liliane	Company 104	Activity 104	City 104	Country 104
PLASSARD	Elise	Company 105	Activity 105	City 105	Country 105
BELLE	Barthelemy	Company 106	Activity 106	City 106	Country 106
CHASSARD	Cyrille	Company 107	Activity 107	City 107	Country 107
BECHET	Eloanna	Company 108	Activity 108	City 108	Country 108
TESTA	Jeanne	Company 109	Activity 109	City 109	Country 109
GAMBIER	Adrienne	Company 11	Activity 11	City 11	Country 11
FIX	Monique	Company 110	Activity 110	City 110	Country 110
JANKOWSKI	Eloannette	Company 111	Activity 111	City 111	Country 111
JOLIBOIS	Micheline	Company 112	Activity 112	City 112	Country 112
BATON	Eloi	Company 113	Activity 113	City 113	Country 113
VAUD	Danielle	Company 114	Activity 114	City 114	Country 114
GABORIAU	Lucien	Company 115	Activity 115	City 115	Country 115
BLANCHERE	Constant	Company 116	Activity 116	City 116	Country 116
ALVES	Maurice	Company 117	Activity 117	City 117	Country 117

The Session page displayed on 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 user. The page context on the server is still the one corresponding to the customers.

If the user selects an element (a supplier) in the Table control, the selected element will be the corresponding element in the context: a customer. **A desynchronization occurs between the dynamic page viewed and the page context found on the server. The WEBDEV engine detects this desynchronization.**

**This is why the Back button must be managed.**

## How to manage the "Back" button?

To manage the "Back" button, you can use:

- the **page synchronization code in Session mode**. This code is called in case of desynchronization.
- a **hidden Edit control** (invisible Edit control) to identify the record displayed in the browser.
- the option **"Call the page synchronization process if the user pressed the browser Back button"** on your buttons and/or links ("Advanced" tab of the button description).

You can choose and combine either of these options according to the type of action that can be performed in the page in Session mode.

## Detailed example of how to manage the "Back" button

A dynamic page is used to browse the different products of a site. The buttons found in the page are as follows:

- **Back to main menu**  
This action requires no specific management of the "Back" button. This button does not handle data files: a 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. It is important that the deleted record corresponds to the record viewed by the user.  
To manage the "Back" button, we recommend that you use:
  - a hidden (invisible) 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 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 "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!

## Managing the "Back" button in practice

### 1 Overview

The browser "Back" button allows the Web users to display the pages that have already been visited.

In a WEBDEV site in Session mode, 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 found on the server can be desynchronized when using the browser "Back" button.

#### 1.1 Two methods can be used to 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:** Prevent from using the browser "Back" button to go back to this page.  
If the browser "Back" button is used to display the previous page, this action will have no effect.
- **Solution 2:** Managing the synchronization (default solution)  
For each action performed in a page from the browser, a synchronization test is automatically run between the HTML page and its context. Two modes can be used to manage the synchronization:
  - default synchronization (mode used by default when creating a new page).
  - programmed synchronization.

#### 1.2 Example of desynchronization

Let's see a site example:

- A browser page contains a browsing Table control linked to the ITEM file and a "Next" link.
- The ITEM file contains a single item, each record includes a letter of the alphabet.
- The page is used to display 6 rows of the Table control, the "Next" link is used to display the next 6 rows.

When opening the page, the Table control displays the 6 first records of the file (from 'A' to 'F'). Let's see the sequence of actions performed by the user:

##### 1. Click on the "Next" link

**Result:** the server is positioned on the next 6 records of the ITEM file and returns their contents to the browser. The browser displays the next page of the Table control with the 6 new contents ('G' to 'L').

##### 2. Click on the browser "Back" button

**Result:** the browser displays the page preceding the first action. The Table control 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 on the "Next" link

**Result:** the server is positioned on the next 6 records of ITEM ('M' to 'R'). The browser is synchronized with the server and it displays the same elements: the Web user has the feeling that some information is missing.

This behavior can have unexpected consequences when modifying a file record (modification of a record other than the one viewed by the Web user for example).

**Reminder:** each action on the browser must trigger a server action: the server sends a response to the browser. The click on the browser "Back" button being a browser action independent of your WEBDEV site, the second condition may not be performed.

## 2 Preventing from using the "Back" button

If the browser "Back" button is used to display the previous page, this action will have no effect.

### 2.1 Operating mode

Disabling the "Previous page" feature of the browser inserts the following JavaScript code into the generated HTML page:

```
<SCRIPT LANGUAGE="JavaScript">
history.forward()
</SCRIPT>
```

When running the page in a browser, it will not be possible to go back to this page by using the "Back" button.

#### Remarks:

- Clicking the browser "Back" button can make the page blink.
- This mechanism can fail if a click of the "STOP" button is run before the forward() statement is run by the browser.

## 3 Managing synchronization

### 3.1 Overview

For each action performed in a page, the mechanism for page synchronization automatically checks the synchronization. This check consists in verifying whether the page displayed in the browser corresponds to the page context found on the server.

Two modes can be used to manage the synchronization:

1. Default synchronization management.
2. Management of the synchronization through programming, in the page synchronization code.

### 2.2 Implementation

**To disable the browser "Back" button for a specific page:**

1. Open the page description window: on the "Page" tab, in the "Description" group, click "Description".
2. In the "UI" tab, for "Browser "Back" button", select "Forbidden".
3. Validate.

**To disable the browser "Back" button for all the project pages:**

1. Open the project description: on the "Project" tab, in the "Project" group, click "Description".
2. Click on the "Advanced" tab.
3. For "Default option for the browser "Back" button", check "Forbidden".
4. Validate. This option will be automatically taken into account for all new site pages.

### 3.2 Default synchronization

The synchronization mechanism is triggered only if the option "Use the mechanism for synchronizing pages" is selected for the page.

If a desynchronization occurs, a warning message informs the user that the requested action has not been performed. The page corresponding to the context on the server is re-displayed. The site can continue to operate.

**To enable synchronization in a page:**

1. Open the page description window: on the "Page" tab, in the "Description" group, click "Description".
2. In the "UI" tab, for "Browser "Back" button", select "Allowed (runs the synchronization code)".
3. Validate. This page will be automatically included in the browser page history: to go back to this page, click the browser "Back" button.

**To enable synchronization in all the pages of the project:**

1. Open the project description: on the "Project" tab, in the "Project" group, click "Description".
2. In the "Advanced" tab, for "Default option for the browser "Back" button", check "Allowed (runs the synchronization code)".
3. Validate. The project pages will be automatically included in the browser page history: to go back to these pages, click the browser "Back" button.

#### Remarks:

- This management mode requires no specific WLanguage code.
- The synchronization mechanism can be disabled for the page controls that do not require a synchronization ("Close" button for example): simply uncheck "Call page synchronization process if the user has used the browser "Previous" button" in the "Advanced" tab of the control description.
- The warning message can be customized (see next paragraph).

## 4 Synchronization through programming

**To set the synchronization through programming:**

1. Open the page description window: on the "Page" tab, in the "Description" group, click "Description".
2. In the "UI" tab, for "Using the browser "Back" button", select "Allowed".
3. 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, simply uncheck "Call the page synchronization process if the user pressed the browser Back button" in the "Advanced" tab of the control description.
4. Enter the code required to set a custom synchronization in the page synchronization code. Use **ChangeAction** in the page synchronization code. This function is used to define the action that will be performed if the page is desynchronized.

#### Remarks:

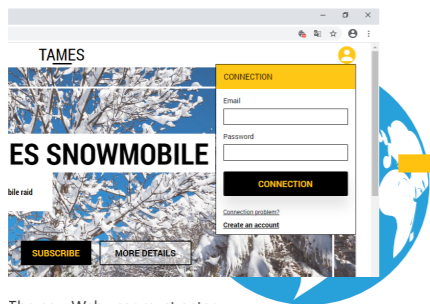
- **ChangeAction** is initialized with "No action" if a WLanguage function that allows displaying or re-displaying a page is used in the page synchronization code.
- To customize the desynchronization message, specify the following in the page synchronization code:
  1. the custom message.
  2. how to re-display of the current page on the server (with **PageRefresh**, for example).
- To perform a synchronization from the information on the user's computer, it is recommended to:
  1. Use a hidden control containing the identifier of the displayed and selected record.
  2. In the synchronization code, find the current record on the browser. This search is performed using the identifier in the hidden control
  3. Refresh the page.

# Cookies: information stored on the end user's computer

A cookie is a file stored on the user's computer by the WEBDEV site. This file can be read at any time from the server or by the browser. Cookies are used to store persistent information on the user's computer.

For example, users must provide their login credentials when logging in to the site for the first time: username, password, ... This information is stored on the end user's computer. Next time the user logs in, the WEBDEV site will read this information in the cookie created during the first visit and will be able to greet the user personally on the first page.

## 1st connection



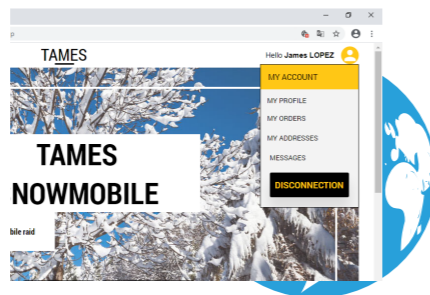
The new Web user must enter his name and password here

The WEBDEV site saves the information given by the Web user on their computer.



## Next connections

The WEBDEV site reads the data previously saved on the Web user computer.



The WEBDEV site displays by default the information read in the cookie

To create cookies in a WEBDEV site, use **CookieWrite**.  
To read the cookies in a WEBDEV site, use **CookieRead**.

# Protecting access to the site: passwords

Protecting the access to a site using passwords is necessary in two cases:

1. Protecting personal information.
2. Restricting the access to a site.

## Protecting personal information

Users can specify a password to access personal information. In a business site, a password can be defined to access the details of previous orders.

In this case, accessing a site requires:

- an authentication:  
When users log in for the first time, they must enter their username and password. Next time they log in, they will be able to access specific information (history, etc.).
- an option to let users recover their passwords.

WEBDEV is provided with all the resources required to manage personal passwords.

## Limiting access to the site

By default, a website is accessible to any user. But a site is not always intended for the general public: some features can be reserved to some types of users (site administrator, etc.).

In this case, the site manager assigns a password to each 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 sales statistics, etc.
- a specific password will be assigned to the members of the marketing team: they will be able to publish 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 end user computer and the Web server is not encrypted. The authenticity of the server identity, the confidentiality and the integrity of exchanged data are not guaranteed.

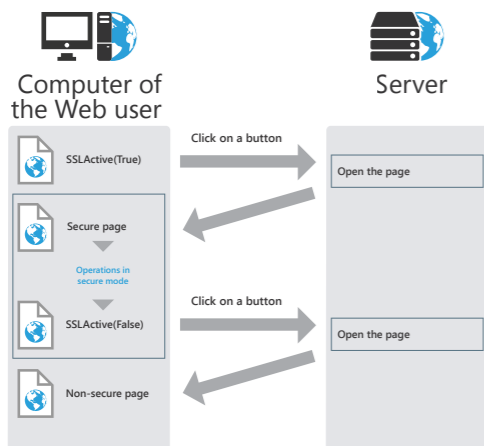
To insure confidentiality, the most used method is the TLS/SSL protocol (Transport Layer Security, formerly Secure Socket Layer). This protocol guarantees the identity of the server to which the Internet accesses and encrypts the data exchanged between the server and the browser.

## Implementing secure transactions via the TLS/SSL protocol

To implement secure 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). For more details, see the WEBDEV online help.

## Transactions secured by TLS/SSL in a WEBDEV site

In most cases, only the transfer of sensitive data must be secured: transferring a credit card number for example. The secure mode is slower. However, a growing number of sites are operating entirely in TLS/SSL thanks to the increase in machine power, which ensures greater confidentiality of user operations.



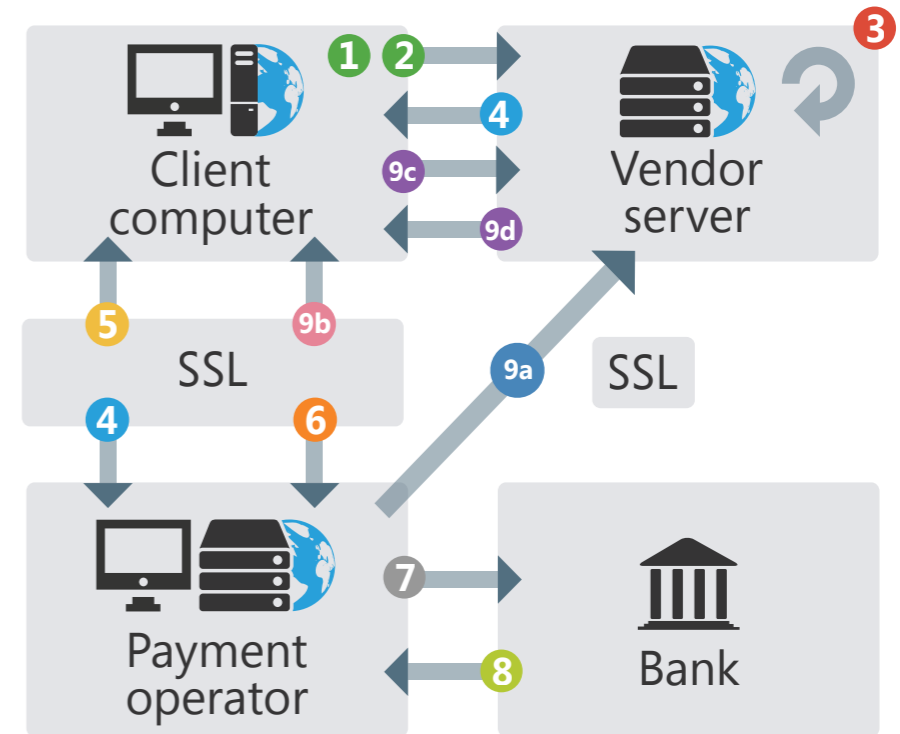
The secured mode is implemented when opening a new page via a button (or link). Simply call **SSLActive** in the browser code of the button (or link). As soon as the secure page is opened, all the actions will be performed in secure (encrypted) mode.

To go back to non-secure mode, simply use **SSLActive(False)** in the browser code of a button (or link) used to open a new page.

# Secure payment

Several providers allow you to implement systems for secure payment. For every payment made by credit card, the provider requests an approval from the card center.

The payment steps are as follows:



- 1 View the site  
Fill the basket
- 2 Identification  
(input of personal details)
- 3 Check the order
- 4 Redirection to the secure payment
- 5 Display the payment page
- 6 Enter the credit card number
- 7 Bank authorization
- 8 Response of the bank
- 9a Response for the vendor site
- 9b Response for the customer
- 9c Redirection of the page for order confirmation
- 9d

# Sending emails

In a website, emails make it possible to:

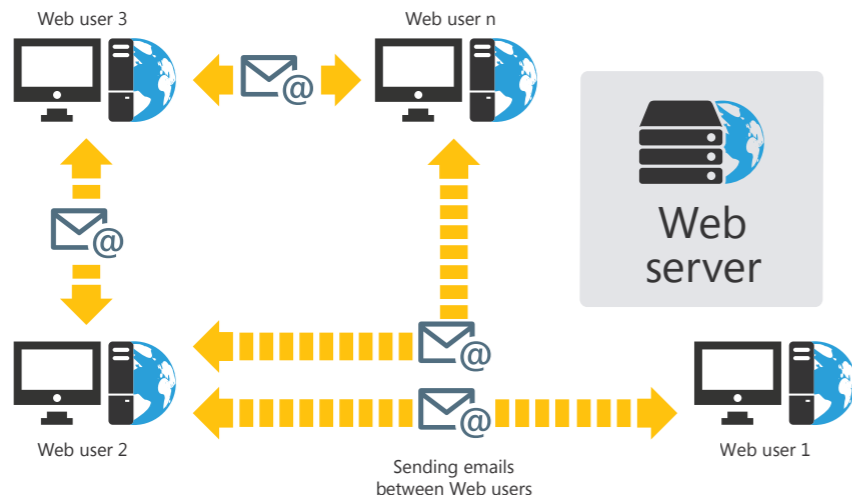
- simplify communication between users: send emails to the website manager, use a mailing list, etc.
- validate an order: in a business website, an email is sent to the customers to validate their orders.
- transfer data from one computer to another: the new orders entered in a business site are transferred by email to the company headquarters, etc.

WEBVED includes two methods to send emails from a website.

## 1 Sending the email from the end user's computer

Use this method when the user must send an email to a specific address: site's webmaster, technical support, sales department, etc.

**EmailOpenMail** opens the user's email program. Some parameters can be set as default: email address of the recipient, etc.



This method allows you to use the user's email program: no specific process is required in the WEBDEV site.

## 2 Sending the email from the server (Session pages only)

In this case, the server manages the emails.

The emails can be sent:

- in the WEBDEV website directly (for an email data transfer, or for an order confirmation email, for example);
- by the email spooler (provided with WEBDEV).



Use this method to customize the sending of emails:

- checking the content of emails sent by the users (mailing website),
- checking the validity of the recipient's address,
- encrypting the information sent by email,
- validating a process, etc.

PART 5

Databases

28

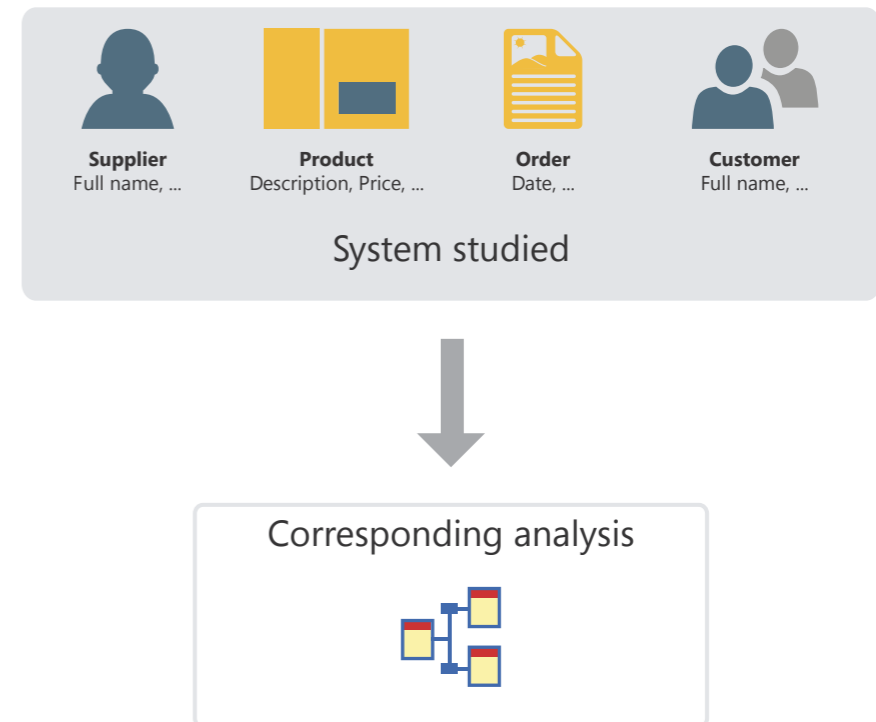
DEVELOP 10 TIMES FASTER



# Analysis: Database structure

When a WINDEV, WEBDEV or WINDEV Mobile project uses data files, it must be associated with an analysis. An analysis is used to describe the data structures (data files, items, etc.) used in your project.

The data model editor allows you to easily create an analysis.



The analysis of a WINDEV, WEBDEV or WINDEV Mobile project corresponds to the LDM (Logical Data Model). The analysis contains the entire structure and data organization: data is grouped into files (data files). Each file contains multiple data fields, called items.

In the analysis, the description of a data file can be linked to a type of file (HFSQL Classic, HFSQL Client/Server, SQLite, etc.).

# Analysis in practice

## 1 Overview

When a project uses data files, it must be associated with an analysis. An analysis describes the structure of the data (data files, items, etc.) used in your project.

There are two methods to describe an analysis:


- **1st method:** Creating the analysis directly (which means a Logical Data Model (LDM)).

## 2 Creating an 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. Click  in the quick access buttons. The creation window appears: click "Data" then "Analysis". The analysis creation wizard starts.
2. Specify the general details of the analysis:
  - **the analysis name and directory.** 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 mode.
  - **the analysis description** to briefly explain what the analysis is about.
3. In the next step, select the type of database:
  - Database HFSQLDrive (paid turnkey solution, on a shared server).
  - HFSQL database in PCSCloud (paid turnkey solution, on a dedicated server).
  - HFSQL database (Client/Server or Classic).
  - Database accessible via a Native Connector.
  - Other types.
4. In the next step, specify if necessary the connection information.
5. Validate the analysis creation wizard. A wizard to create the first data file appears automatically.
6. Create all the elements of the analysis (data files, items and links).

- **2nd method:** Creating the Conceptual Data Model (CDM), then automatically creating the analysis from the CDM. For more details, see the online help.

This chapter presents the first method.

### 2.2 Adding a data file to an analysis

The analysis defines the structure of the data files used by the project. In the analysis, a data file can be:

- A new data file.
- A preset data file, provided with WINDEV, WEBDEV or WINDEV Mobile.
- A data file imported from an existing database (using a specific format, for example).

The following paragraphs explain how to create a data file.

To create a new data file:

1. On the "Analysis" tab, in the "Creation" group, click "New file". The data file creation wizard starts.
2. Select "Create a new data file description".
3. Specify:
  - **the name of the data file.** This name is the logical name of the data file. It will be used to handle the data file.
  - **the caption of the data file** that briefly describes the file subject.
  - **the representation of a record** in the data file. This representation makes the questions asked when describing the links easier to understand. 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 is automatically calculated by WINDEV, WEBDEV or WINDEV Mobile.

- **the type of database associated with the data file:**

- Database HFSQLDrive (paid turnkey solution, on a shared server).
- HFSQL database in PCSCloud (paid turnkey solution, on a dedicated server).
- HFSQL database (Client/Server or Classic).
- Database accessible via a Native Connector.
- Other types.

Depending on the type selected, this data file will be handled by the HFSQL engine, an OLE DB driver or one of WINDEV or WEBDEV's Native Connectors (SQL Server, Oracle, etc.).

4. The new data file becomes the current data file. The description window of the data file items opens automatically. It allows you to describe the file items.

To import a preset description of a data file:

1. On the "Analysis" tab, in the "Creation" group, click "New file". The data file creation wizard starts.
2. Select the option "Select a description among the preset data files".
3. Choose the preset data file to be created (you can select multiple data files). This data file will be imported into the current analysis. This data file can be modified later.
4. Select the items you want to keep. These items can be modified later.
5. The imported data file is automatically inserted into the current analysis.

**Remark:** To modify a data file or the items in a data file:

1. Select the data file.
2. Select "Data file description" or "Description of items" in the context menu.

**Two solutions can be used to import an existing data file description:**

**Solution 1: From the data model editor**

1. On the "Analysis" tab, in the "Creation" group, click "New file". The data file creation wizard starts.
2. Select "Use the data files from an existing database" and select the type of database.
3. Specify the source database containing the descriptions to be imported and the type of database. Depending on the selected type, specify the requested information.
4. Select the tables or data files whose description must be imported and validate.

5. The imported data file is automatically inserted into the current analysis.

**Solution 2: From the Windows explorer**

Drag the data file description from the Windows explorer and drop it onto the data model editor.

### 2.3 Creating an item

To create an item:

1. Double-click the data file where the item will be created. The description window of the data file items opens.
2. Click the first empty row in the table.
3. Specify the name, caption and type of the item.
4. In the right section of the window, indicate the details of the new item (type, size, default value, iteration direction, etc.).
5. Specify the details about the shared information: click "Parameters of the control bound to the selected item (shared information)".
6. Validate the item description.

**Remark:** you can also create an item from the metatypes available in WINDEV. To do so, click the button. The list of available metatypes appears.

### 2.4 Creating a link

Different types of links can be created between data files. For more details on links, see Characteristics of the links defined in an analysis.

To create a link:

1. On the "Analysis" tab, in the "Creation" group, click "New link". The mouse cursor turns into a pen.
2. Select the two data files to be linked. The link description window opens automatically.
3. To define the cardinalities between the two data files:
  - select the desired cardinalities (0, 1; 1, 1; 0, N; 1, N).
  - answer the questions. The cardinalities will be automatically updated.
4. To define the advanced cardinalities, check "Display advanced cardinalities" and answer the questions.
5. Specify the link caption and briefly describe the purpose of the link.
6. Specify the keys to be linked.
7. Define the integrity rules. These rules ensure data integrity when one of the keys of the relationship 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, there is an **owner** data file and a **member** data file:

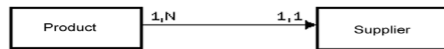
- the owner data file is the owner of the key.
- the member data file is a member of the analysis data files containing a copy of the key.

To set the link between two data files, the key of the owner data file is copied to the member data file.

For example, the key of the Supplier data file is copied to each record of the Product data file. Several records of the Product data file can have the same key of the Supplier data file:

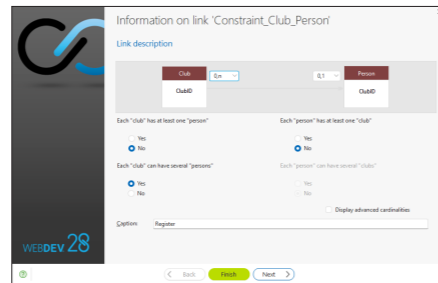
- Supplier is the owner data file,
- Product is the member data file.

This type of link is represented as follows:



The description of cardinalities is essential: it is the basis for respecting the referential integrity of the database.

To put it simply, these two questions can be broken down into four questions. For example, to describe a link between the Club data file and the Person data file:



- each "Club" has at least one "Person": Yes/No?
- each "Club" can have several "Persons": Yes/No?
- each "Person" has at least one "Club": Yes/No?
- each "Person" can have several "Clubs": Yes/No?

#### 3.2 Cardinalities

Cardinalities are used to count the relationships between two data files.

Cardinalities are defined according to the answers to the two following questions:

1. For each record of the data file, what is the **minimum** number of records in the other data file this record 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 of the data file, what is the **maximum** number of records in the other data file this record is linked to?

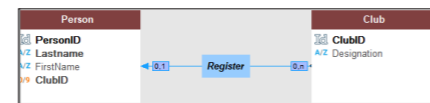
The answer provides the second part of the cardinality (**maximum cardinality**):

- 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

#### Example of cardinalities:

This example presents two different cardinalities:



- **0,1 cardinality:** A person can be a member of a single sport club. This person does not have to belong to a sport club.
- **0,N cardinality:** A club can have no member or several members.

#### 3.3 Advanced cardinalities

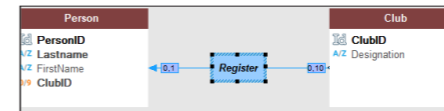
Cardinalities can be defined more precisely. This is referred to as **advanced cardinalities**.

These cardinalities allow you to specify exactly the minimum and maximum cardinality.

To define the advanced cardinalities, check "Display advanced cardinalities" in the link description.

#### Example of advanced cardinalities:

This example presents two different cardinalities:



- **0,1 cardinality:** A person can be a member of a single sport club. This person does not have to belong to a sport club.
- **0,10 cardinality:** A club can have 0 to 10 members.

#### 3.4 Referential integrity

Referential integrity in a database is a set of constraints that maintains consistency of data in a relationship between two data files.

Referential integrity checks that:

- if a record is deleted from the owner data file, the matching records in the member data files are also deleted,
- if a record is added to a member data file, there is a matching record in the owner data file,
- if a record is updated in the owner file, the unique key doesn't change, etc. The referential integrity is checked based on the nature of the link between the data files.

For more details on referential integrity, see the online help.

#### 3.5 Types of links

There are different types of links between data files:

- parallel
- optional
- complement
- shared
- complex

#### Parallel link

In a parallel link, **each record** found in a data file (Product) is linked to **a record** in another data file (Info), and **conversely**.



In a parallel link, the identifier of the Product data file is copied to the Info data file. This identifier is also a unique key in the Info data file.

Product is the owner data file and Info is the member data file.

**Remark:** This type of link is rare because the two data files can be combined into one.

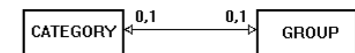
The parallelism of records in the data file is respected if the following operations are run simultaneously on the two data files:

- creation of a record.
- deletion of a record.
- reindexing with compression.

#### Optional link

For an optional link:

- **Each record of a data file** (Category) **has no matching records or has one matching record** in the other 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).

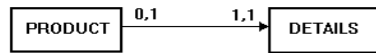


An optional link is established by copying the identifier from one data file to the other.

**Complement link**

For a complement link:

- **Each record** from a data file (Product) **has no matching records or has one matching record** in another data file (Details).
- **Each record** in the other data file (Details) **has a matching record** in the first data file (Product).



Complement links are quite common. They are used when a record can have additional optional information.

In a complement link, the key of the Product data file is copied to the Details data file. To ensure the maximum cardinality of 1, it remains a unique key. Since this key is unique, it is not possible to insert more than one record in the Details data file for a record in Product.

Product is the owner data file and Details is the member data file.

**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).

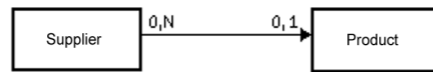
In a shared link, the key of the Supplier data file is copied to the Product data file. It becomes a multiple key to optimize the speed of the integrity check.

Supplier is the owner data file and Product is the member data file.

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.

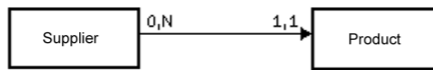
**Shared link with a 0,n - 0,1 cardinality**



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:

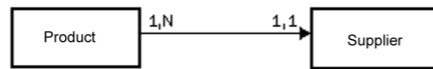
- an owner may have no member (a supplier may have no product).
- each member has a single owner (each product has a single supplier).

**Remark:** This type of link is quite common. WINDEV allows you to automatically create windows that manage data files with a 0,N - 1,1 relationship.

**Shared link with a 1,n - 0,1 cardinality**

With this type of link:

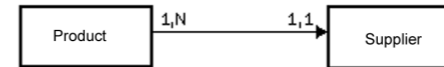
- each owner has at least one member (a supplier has at least one product).
- a member may have no owner (a product may have no supplier).



**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 need to define a **relationship file**.

The relationship file will be automatically created. It will contain a unique key that includes the two keys of the related data files.

The relationship file can also contain information specific to the link.

A complex link includes two shared links.

**Example of complex link**

An order (Orders data file) can contain one or more products. A product (Product data file) can be used in several orders.

In this case, a relationship file is required (OrderLine data file).

The OrderLine data 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 Possible operations in an analysis**

WEBDEV allows you to perform the following operations on an analysis:

- Duplicating/Copying an LDM: Duplicating an LDM allows you to have two identical LDMs with different names.
- Delete an LDM.
- Rename an LDM.
- Associate an LDM with a project.
- Enlarge or reduce a LDM in the editor.
- Move a LDM in the editor.

WEBDEV allows you to perform the following operations on a data file found in the data model editor:

- Duplicate/Copy a data file.
- Delete a data file.
- Rename a data file.

WEBDEV allows you to perform the following operations on an item of a data file found in the data model editor:

- Duplicate/Copy an item.
- Delete an item.
- Rename an item.

For more details, see the online help.

## 5 Generating the analysis (LDM)

The analysis is generated before the programming cycle and after describing the data files. This generation is used to:

- validate the changes made on the analysis (LDM).
- create the modules required for programming.
- automatically update the data files if necessary.

As long as the analysis (the LDM) is not generated, the analysis description (the data files) cannot be used in the project.

To generate the analysis, go to the "Analysis" tab, "Analysis" group and click "Generation".

The generation includes three steps:

- Verification of the changes made and generation of the physical files corresponding to the analysis description.
- Automatic modification of accessible data files (files in the "EXE" directory of the project).
- Project synchronization.

## 6 Managing the analysis versions

To manage the different versions of the analysis:

1. On the "Analysis" tab, in the "Analysis" group, click "Generation".
2. Select "Manage versions".
3. In the wizard, specify whether you want to work on the current analysis or on a specific analysis.

**Remark:** when an analysis is damaged, you can select the damaged analysis to restore one of the earlier versions, for example.

All the versions of the analysis are displayed as a graph. The yellow box indicates the version number of the analysis and the text on the right shows the generation date.

The "small rectangles" displayed between two versions indicate:

- one rectangle: few changes were made between the two versions.
- several rectangles: several changes were made between the two versions.

**To see the details of the changes made between two versions:** Double-click the row containing the "small rectangles" or click [Changes].

The options available in the version manager are as follows:

- Restoring an analysis.
- Undoing the last generation.
- Resetting the version number to 1.

### 6.1 Restoring an analysis

To restore an analysis from the version manager:

1. Select the version to restore.
2. You can:
  - restore the version in 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:** The data files of an analysis whose version number is more recent than the restored version cannot be reopened. In this case, you must also restore the data files of the restored analysis version or delete the existing data files and create them again.

### 6.2 Undoing the last generation

Undoing the last generation of the analysis allows you to restore the analysis to its state before the last generation. All changes made since then are not applied.

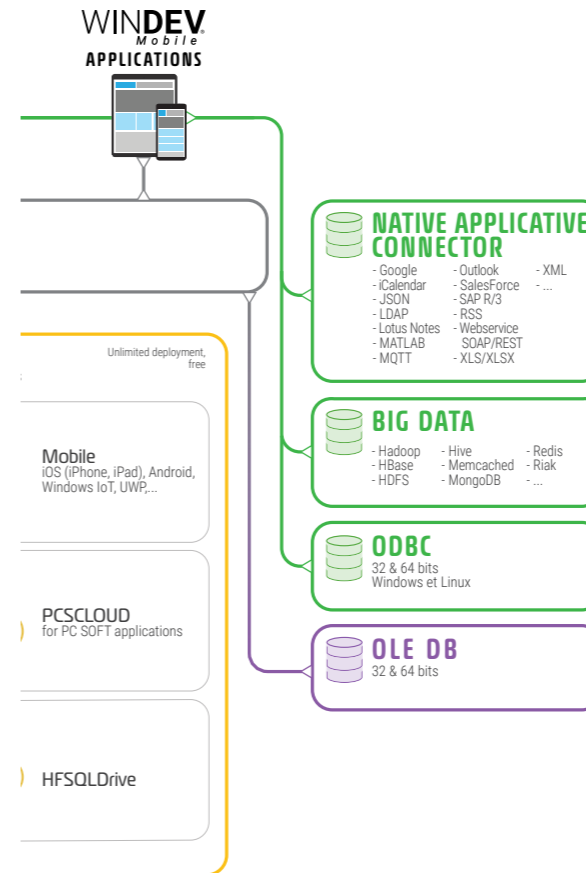
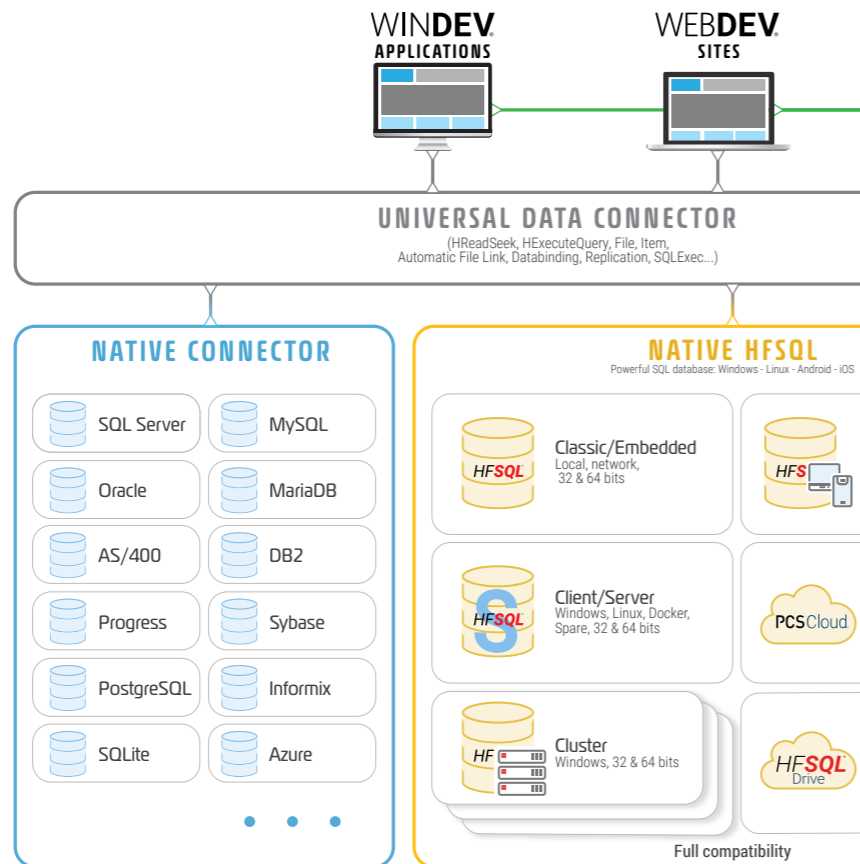
### 6.3 Resetting the version number to 1

The analysis generation number can be reset to "1": on the "Analysis" tab, in the "Analysis" group, click "Generation" and select "Reset analysis version to 1".

In this case, the analysis version number is reset to one. No specific action is performed on the data files.

# Types of accessible files

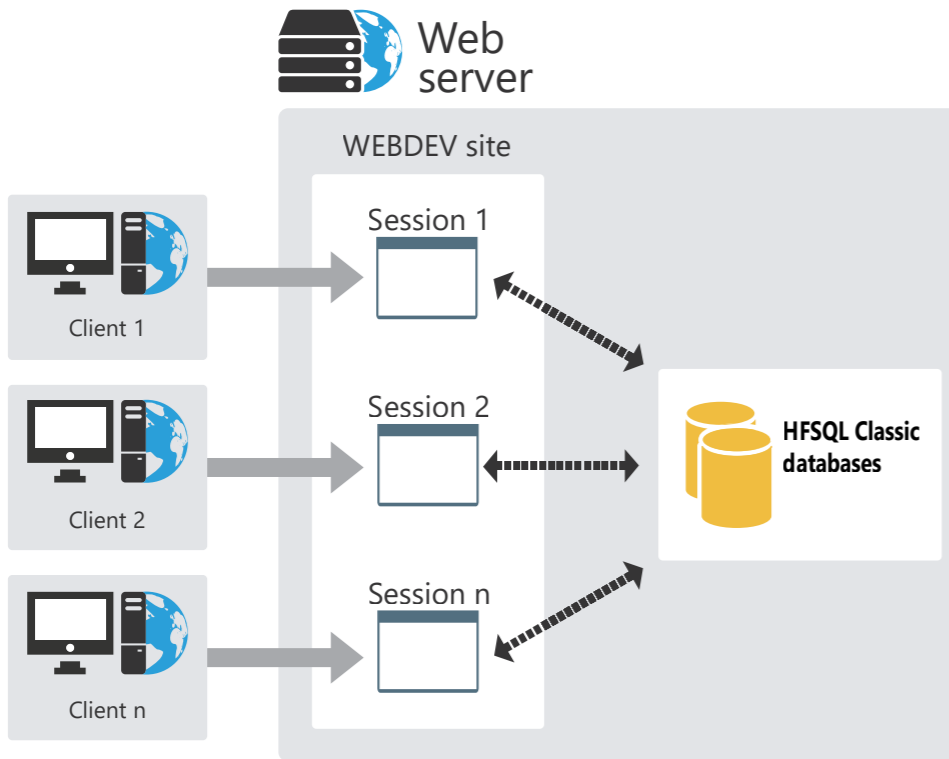
WEBDEV, WINDEV and WINDEV Mobile offer simple access to the majority of databases on the market.



# HFSQL Classic

A WEBDEV HFSQL application can be in Classic mode (HFSQL Classic) or in Client/Server mode. The characteristics of the Classic mode are as follows:

- A site that uses HFSQL Classic is run on different browsers. A session is started on the server for each site run.
- The data files are located in a directory (on the Web server or on another computer) that can be accessed by the sessions of the WEBDEV sites. Each session physically accesses the data files.
- The processes (queries, read or add records in a data file, etc.) are performed by each session.

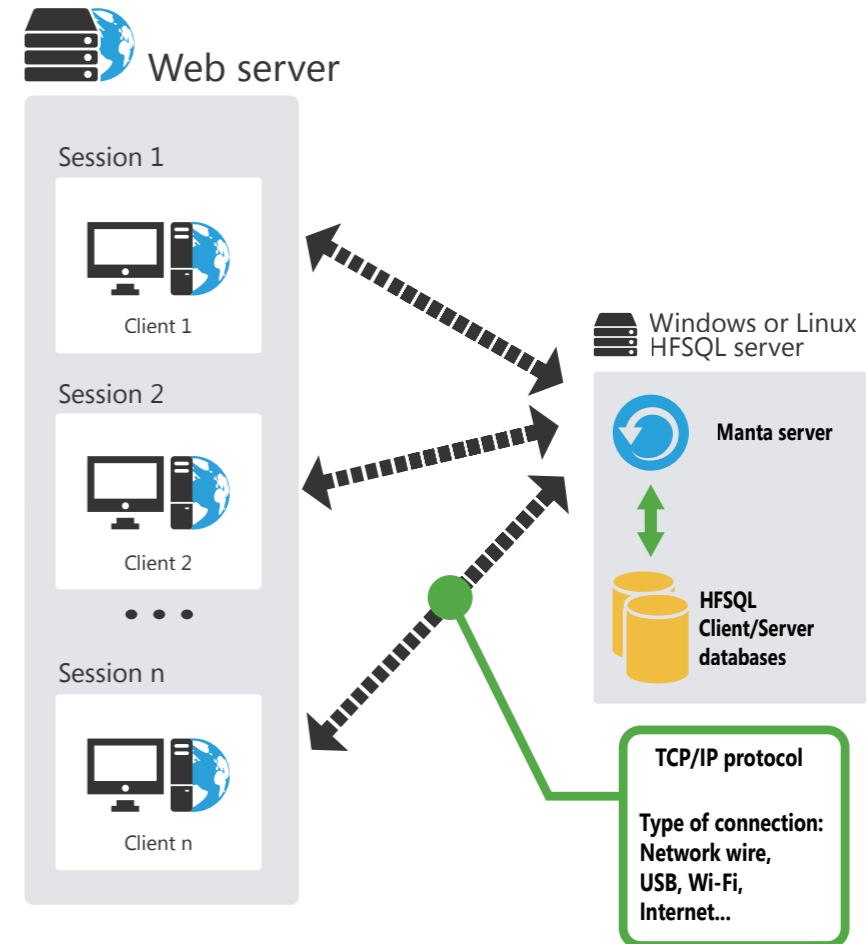


# HFSQL Client/Server

HFSQL 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:

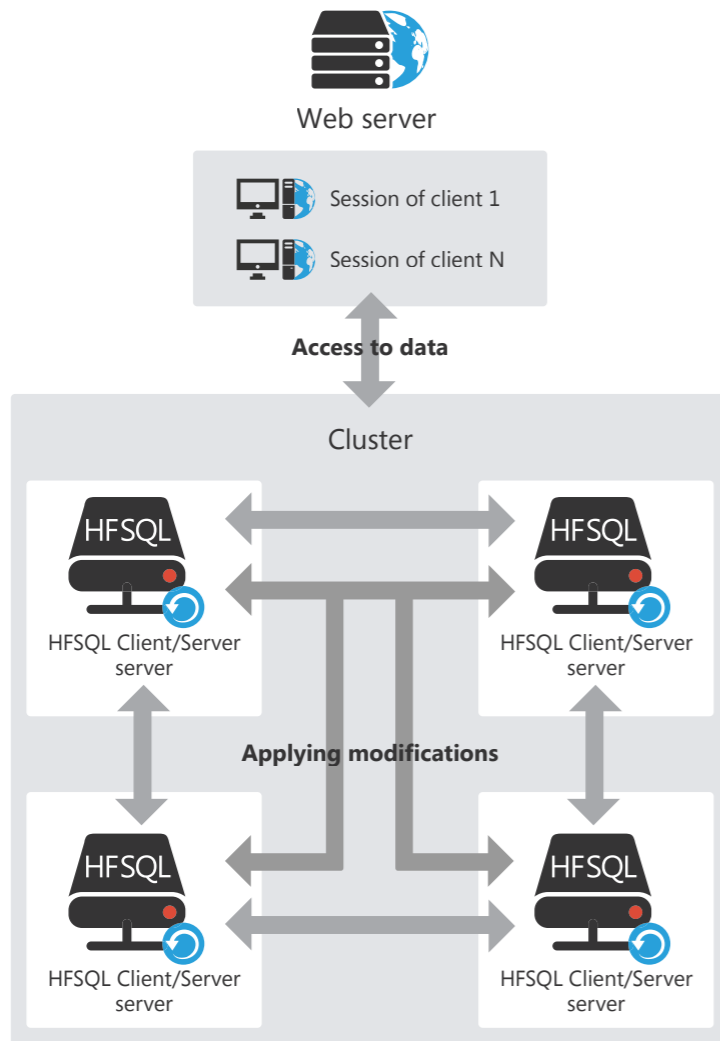
- An HFSQL Client/Server site is run on different client computers. A session is started on the Web server for each site run.
- The data files are located on a server (HFSQL server). Only the server physically accesses the data files.
- All processes (queries, read from and write records to a data file, etc.) are performed on the server.



# HFSQL Client/Server clusters

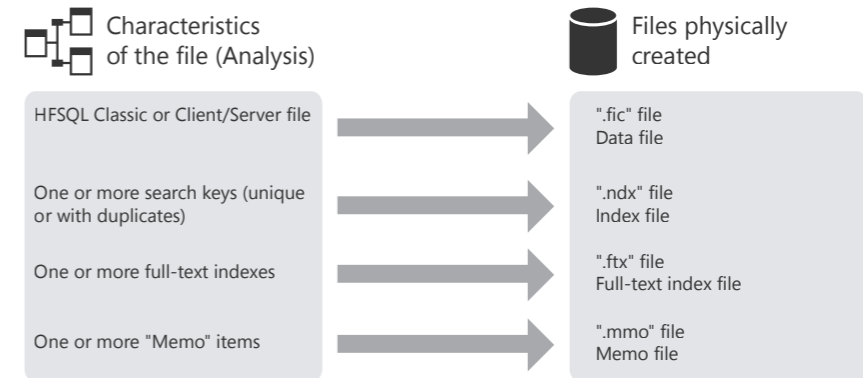
HFSQL Cluster is an extension of the HFSQL Client/Server database model. In a database cluster, all HFSQL servers contain a copy of the databases and are synchronized in real time.

- The read load can be balanced across multiple servers.
- You can make changes to the physical configuration without interruptions on client computers.
- If one of the servers crashes, the client is automatically redirected to an operating server.



# HFSQL: files physically created

The data model editor allows you to define the structure of the data files. Different files are physically created depending on the information typed in the data model editor.



**Note:** This diagram only presents the main created files. Other specific files can be created if the data file uses replication, logs or transactions.



# Binding controls to data

A page can display information from:

- a database: controls are directly bound to the items in the data files or queries available in the database.
- variables in the application's code (variables global to the page or to the project or parameters passed to the page).

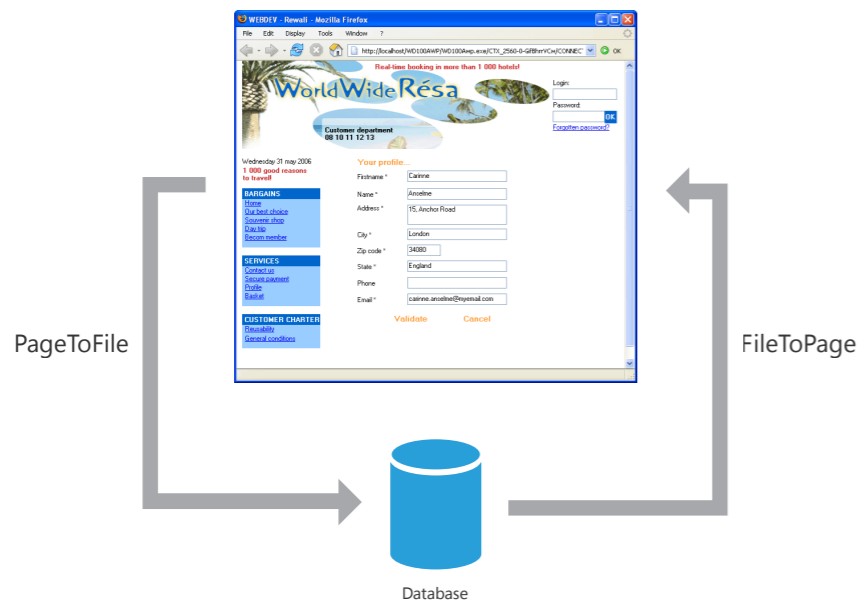
To display this information in a page, the controls of the page must be bound to:

- the different items in the database.
- the available WLanguage variables.

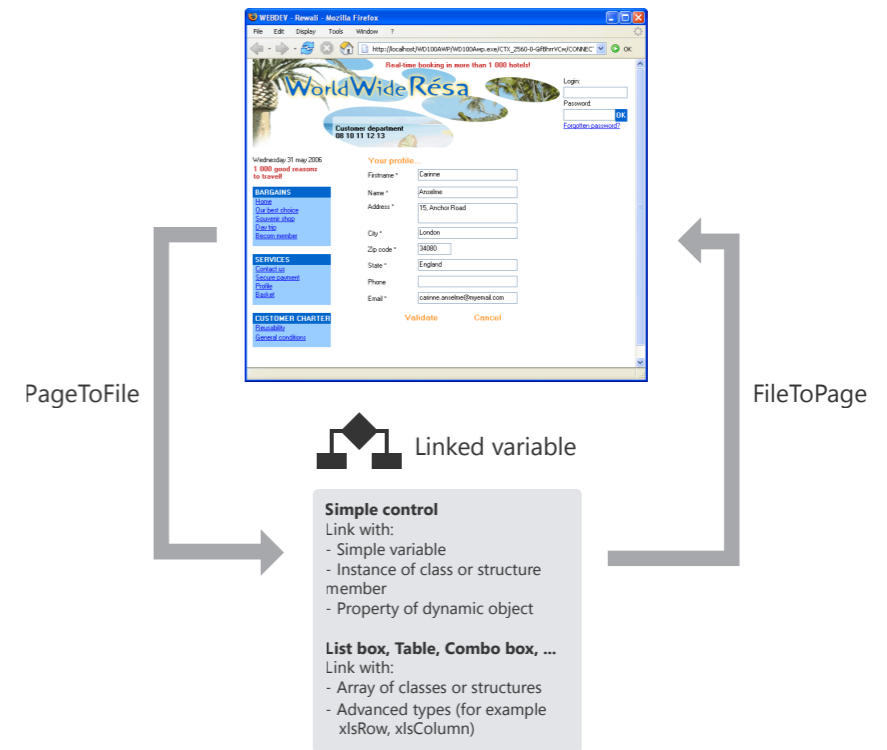
There are multiple methods to display and update values from a data source:

- The binding between a control and an item or variable can be defined in the page editor, in the control description ("Binding" tab).
- **PageToFile** updates the record or variable with the data in the page.
- **FileToPage** updates the data displayed in the page with either the information saved in the data file or variable.

## Control - item binding



## Control - variable binding

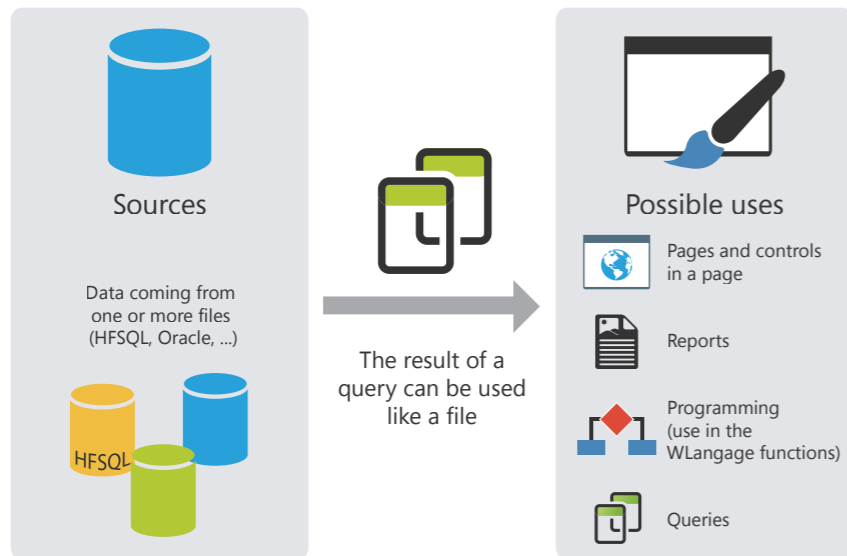


# Queries: concepts

A query is used to request information from a database to view, insert, change or delete data. The query structure defines the data used.

A query can request information from one or more data files.

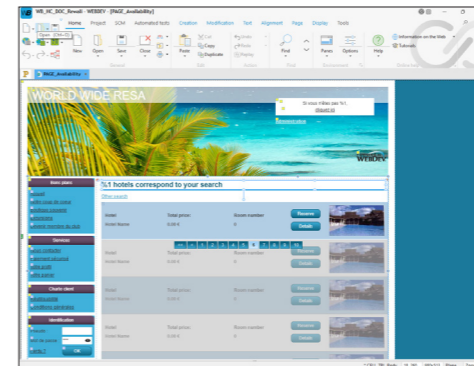
The query editor allows you to easily create queries without programming.



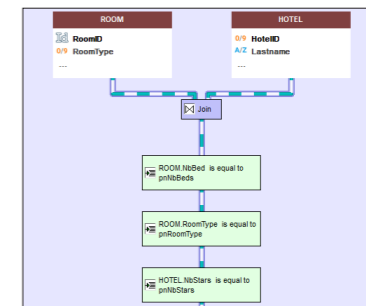
**Remark:** In the code, you can handle queries and data files in the same way. Queries can be associated with controls in the UI (a Table control, for example) to display the desired data.

# Embedded queries

The controls of a page can be linked to a data file or query. These controls can also be linked to a query created when building the control.



Looper control linked to an embedded query



Embedded query : MyPage\_1\$Query

In this case, the query is included in the page. It is in the WWH file corresponding to the page. If the WWH file is copied (to another project, for example), the embedded queries used by the page will also be copied.

# The Table/Looper control

Table/Looper controls can be used to display a set of information (e.g., the content of a data file). The content of these controls can come from three different sources:

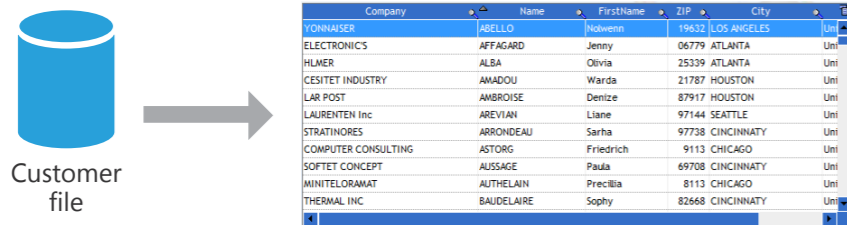
- Table/Looper controls with direct access to the data source.
- Table/Looper controls populated programmatically.
- Table/Looper controls with in-memory data source.

**Remark:** These three modes will be explained in detail for the Table control. However, the same concepts apply to the Looper control.

## Table control with direct access to the data source

A Table control with direct access to the data source directly displays data from a data file or query. The data file structure is described in the data model editor and the data is entered in the site, for example.

When you loop through the data file, the data is displayed in the Table control. The data file is read for each row displayed: the record read is displayed in a row of the Table control.



The data displayed that is not linked to the data file is lost when the Table control is refreshed (when scrolling, for example).

The number of records displayed in the control can be limited by setting a filter (**HFilter** used in the control initialization code).

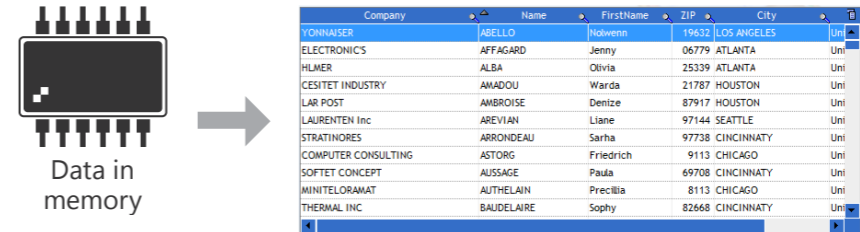
Several WLanguage functions can be used to handle Table controls based on a data file. These functions start with "Table".

**To add a record to the Table control**, add the record to the corresponding HFSQL data file (using **HAdd**) and refresh the Table control with **TableDisplay**.

**To delete a record from the Table control**, delete the record from the corresponding HFSQL data file (using **HDelete**) and refresh the Table control with **TableDisplay**.

## Table control populated programmatically

In this case, the Table control directly displays the data loaded in memory. The data is added to the Table control programmatically (with **TableAddLine**, for example).



When data is in memory, you can perform different actions in the Table control (sort columns, search in columns, etc.).

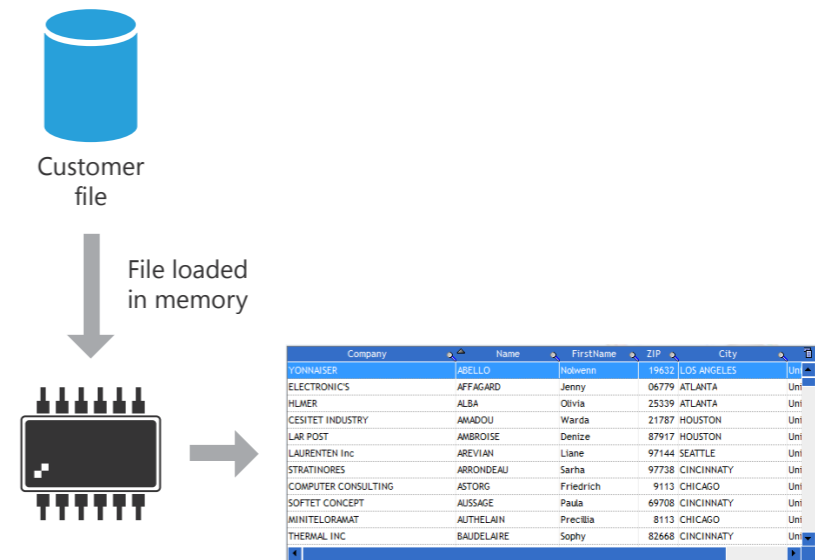
## Table control with in-memory data source

Table controls with in-memory data source combine the advantages of Table controls with direct access to the data source and Table controls populated programmatically.

The Table control is linked to the data file but the entire content of the data file is loaded in memory. Sort and the search features are available for all columns.

The data that is not linked to the data file is not lost when moving the scrollbar (in Check Box columns, for example).

Since the data file records are loaded in memory, this type of control is recommended for data files with less than 100 000 records (to avoid memory overflow).



**Remark:** The different modes to populate the controls (programmatically, with direct access to the data source and with in-memory data source) are available for different types of controls: List Box, Combo Box, Table, Looper, etc.

# Retrieving data from a site

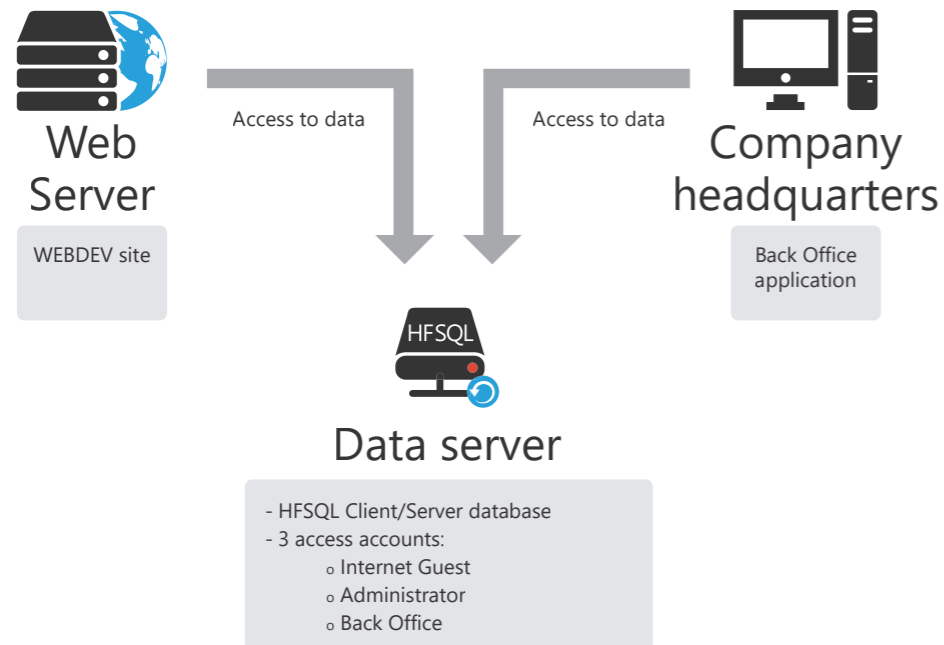
A commercial site allows end users to place orders, make reservations and more, ... Several methods can be used to retrieve this information.

**Tip:** Process the retrieved data in a "Back Office" application developed with WINDEV.

## 1 Sharing data

The data of a WEBDEV site can be directly shared with the data found in the Back Office application used to process the operations performed on the site. Simply use an HFSQL Client/Server database.

Here is how it works:



**Benefit:** The site data is accessible in real time.

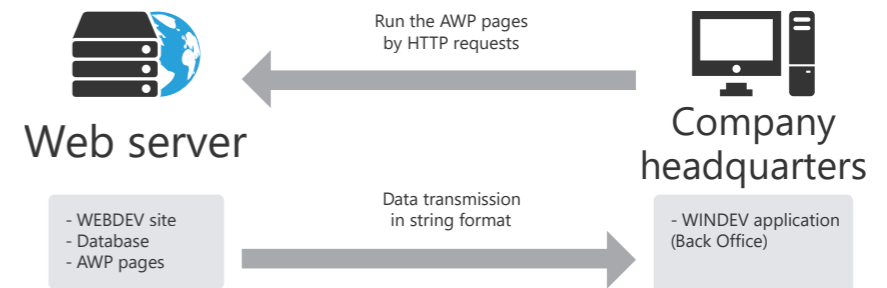
**Drawback:** Need to manage security in an optimal way: use a secured connection to access data, open a specific port to access data (port 4900 by default), configure the firewall, etc.

*This solution is recommended for a site installed on a dedicated server.*

## 2 Retrieving data from a site>

### 2.1 Via a query Active WEBDEV Page

WEBDEV allows you to create Active WEBDEV Pages. WP pages are independent pages that can run processes from the parameters passed as command lines to the page. Active WEBDEV Pages can be used to retrieve data entered on the site.



Active WEBDEV Pages on the Web server receive regular HTTP requests run by the Back Office application. The data is returned in string format and processed by the WINDEV application.

**Benefits:**

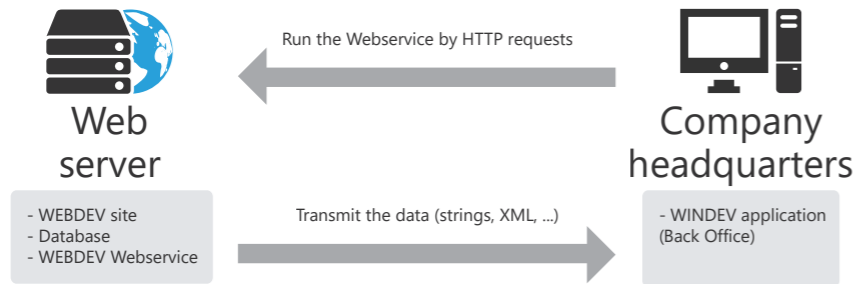
- Solution that can be easily implemented.
- Equivalent to a custom Web service.
- No additional executable is required on the server. No additional session is run on the server.

**Drawback:** Unidirectional data retrieval: no data update toward the server.

*This solution can be used on a shared server.*

### 2.2 Via a Webservice

WEBDEV allows 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 located on the Web server is queried 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:** Unidirectional data retrieval: no data update toward the server.

*This solution can be used on a shared server.*

### 2.3 Via FTP:

WEBDEV allows you to regularly save the data files of the WEBDEV site. 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:**

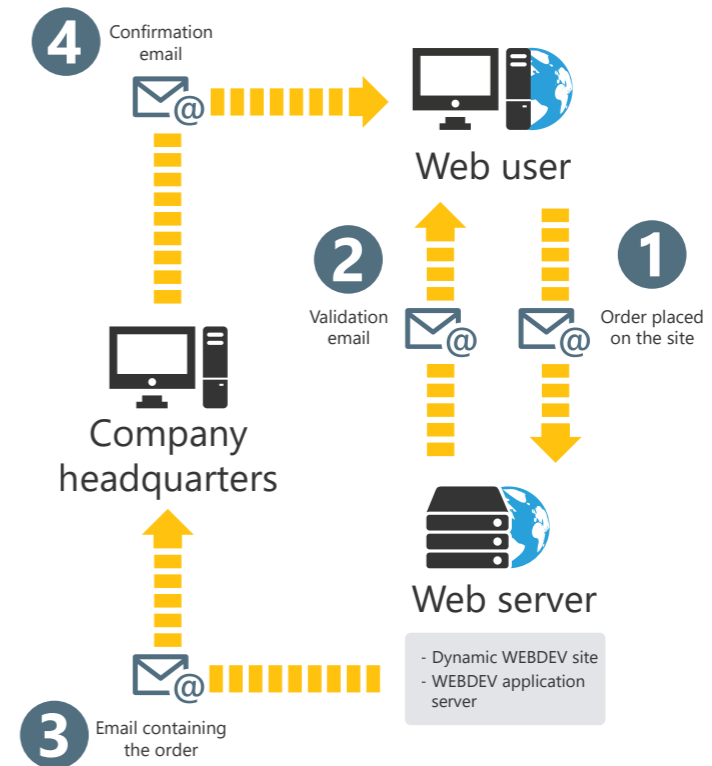
- Solution that can be easily implemented.
- No additional executable is required at the hosting company.

**Drawback:** All data files found in the application are retrieved.

*This solution can be used on a shared server.*

### 2.4 Via email:

How it works:



1. The user places an order on the site.
2. The WEBDEV site informs the user that the order was successfully registered. The user is notified that a confirmation email will be sent.
3. The WEBDEV site sends an email to the company headquarters. This email contains the order placed by the user.
4. The company sends an email to notify the user that the order has been processed.

**Benefits:**

- Solution that can be easily implemented.
- No additional executable is required on the server.

**Drawback:** Unidirectional 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 a 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 (unidirectional replication)
- transmit the online orders to the company headquarters and transmit the stock update to the Internet site (bi-directional replication).

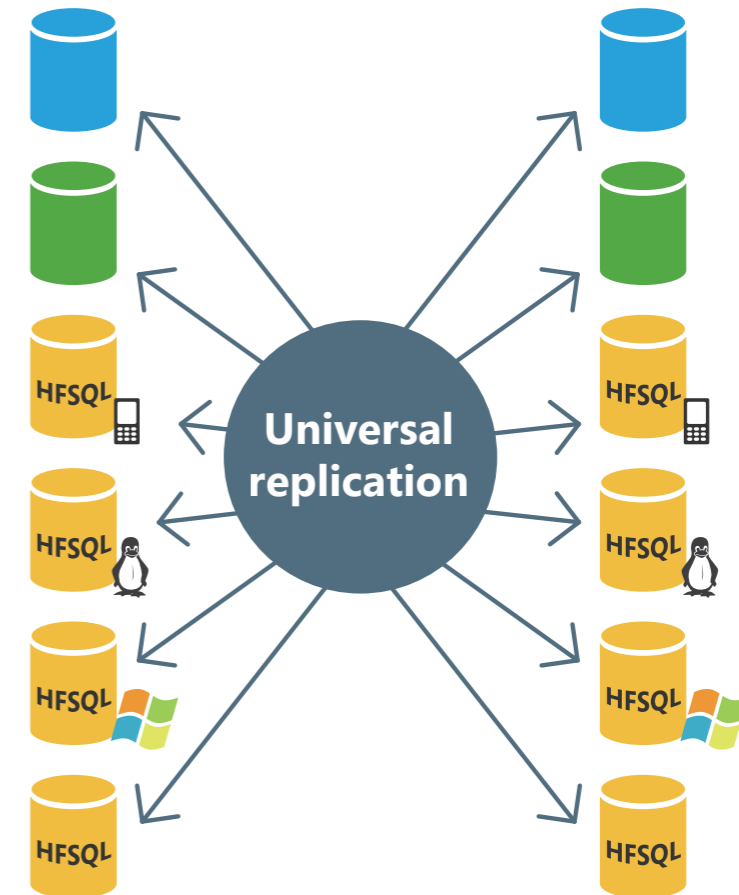
**Caution:** The replication constraints must be taken into account during the project creation. Indeed, several rules must be followed when describing the analysis and the file items.

The replication can be performed by email or via the assisted universal replication. For more details on replications, see the online help.

# Universal replication

Universal replication allows updating databases of similar or different formats (HFSQL Classic, Oracle, SQL Server, etc.). For example, you can synchronize an HFSQL Classic database with an Oracle database.

Universal replication uses a centralized model: all databases are synchronized with a master database. Then, the master database reflect the changes in the other databases.



The synchronization can be adapted to special cases. For example, you can retrieve the records regarding a specific product, or the records created on a specific date, manage conflicts, etc. These adaptations must be done through programming with **HRplFilterProcedure**.

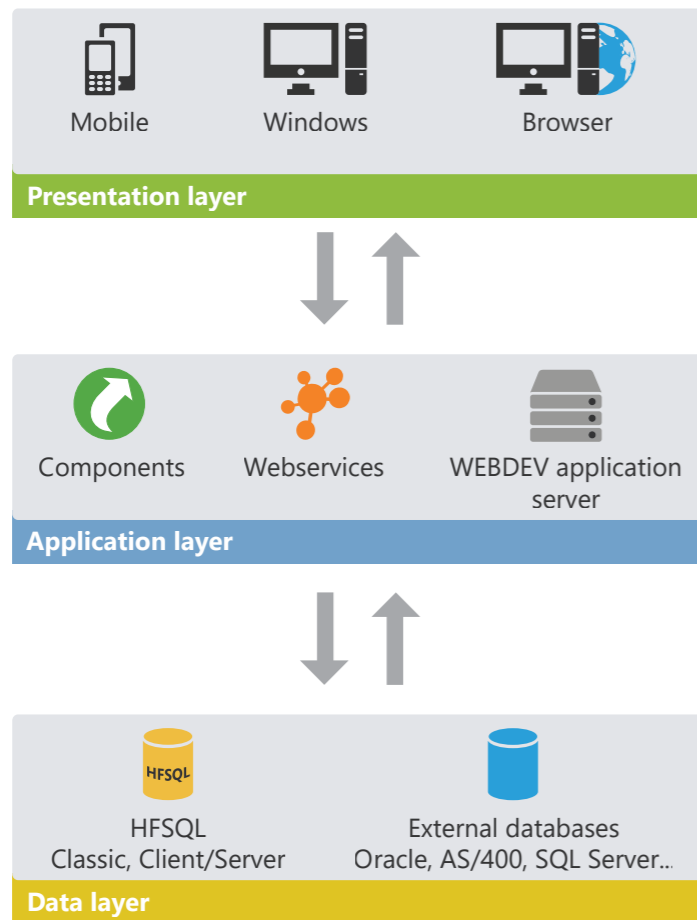
# 3-Tier architecture

3-tier architecture is an application architecture model.

It consists in dividing an application into 3 tiers:

- Presentation tier: the User Interface,
- Application (or processes) tier: business logic of the application,
- Data tier: to access persistent data.

The objective with this separation is to make each tier independent, in order to facilitate maintenance and future developments of the application. This provides better security because only the Application tier allows access to the database. It also optimizes teamwork and cross-platform development.



The 3-tier architecture is fully compatible with the development of applications or sites using WINDEV, WEBDEV or WINDEV Mobile.

# Testing a site: Elements to be tested

Tests play an essential role in the development of a site. A test checks the behavior and usability of the site.

The main points to check in a WEBDEV site are as follows:

- **how the site works:**  
sequence of pages, execution of code typed, etc.
- **appearance of the site:**  
use different browsers (Internet Explorer, Chrome, FireFox, Edge, etc.), use different screen resolutions, resize the browser, etc.
- **Web-specific features:**  
print, cookies, browser "Back" key, etc.
- **access to the site by different users:**  
management of usernames and passwords, management of concurrent accesses to the data files, etc.
- **regression of a site during an update:**  
validation of the changes made in a site and verification to ensure that non-modified features are still accessible.
- **load test** (for a site installed on the server):  
validation of the maximum number of users who can access the site.

**Remark:** To determine the coverage of tests run, WEBDEV includes a "Code coverage" analysis. Code coverage is used to measure the coverage of tests run on a site. Each line of code run is considered as "tested".

Caution: The code coverage does not allow you to know if tests are successful: it only allows you to know whether a line of code has been executed.

The code coverage is used to detect the lines of code that have never been executed.



# How to test a site?

Multiple tools are available to test a WEBDEV site:

- WEBDEV's main editor.
- the WEBDEV administrator.  
This module is required to run a dynamic WEBDEV site (in Session mode or in AWP mode). On the development computer, dynamic sites can be tested using a test page created by the administrator.
- WCTestSite, load test tool.

The table below presents how and when these test modes must be used.

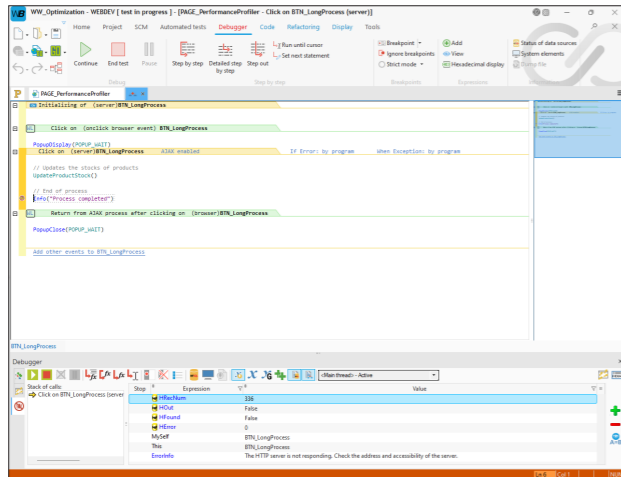
Test mode	Type of test	Running the test
Editor	<p>Running the test from the editor allows you to test:</p> <ul style="list-style-type: none"> <li>• the site features,</li> <li>• the code used in the different events and processes. A powerful debugger allows you to monitor the execution of different events and processes.</li> <li>• the use of the site with different browsers (installed on the development computer).</li> </ul> <p>This type of test:</p> <ul style="list-style-type: none"> <li>• allows you to use the debugger.</li> <li>• is taken into account by the "Code Coverage".</li> </ul>	<ul style="list-style-type: none"> <li>• On the "Project", in the "Test mode" group, expand "Test mode" and select "Debug project".</li> <li>• On the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Debug project from the home page".</li> </ul> <p>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 site features,</li> <li>• the features specific to the Web (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).</p> <p>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>
WCTestSite (dynamic WEBDEV site only)	<p>WCTestSite is used to run stress tests.</p> <p>WCTestSite 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 WCTestSite).</li> <li>2. Install WCTestSite 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>Remarks:</p> <ul style="list-style-type: none"> <li>• Each computer establishes a defined number of connections.</li> <li>• Each connection performs the selected scenario.</li> </ul>
Automatic tests	<p>Automatic tests are used to test procedures and classes of a site at different levels of development.</p> <p>The test scenarios are generated in WLanguage from the procedure test or from the class test. The scenarios can be modified in the code editor.</p> <p>This type of test is taken into account by the "Code Coverage".</p>	<ol style="list-style-type: none"> <li>1. Display the procedure or the class in the "Project explorer" pane.</li> <li>2. Open the popup menu of the element (right click) and select "Create a unit test".</li> </ol>

# Debugger: principle

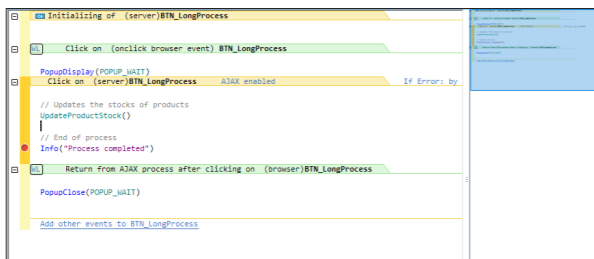
WEBDEV includes a debugger to help you optimize your code.

The debugger allows you to run your dynamic sites step by step, by viewing the WLanguage code run, the content of variables, etc.



To start the debugger, use:

- **breakpoints in the code editor:** the debugger will be automatically started when the line preceded by a breakpoint is run. To insert a breakpoint, simply click to the left of the line of code: a red dot appears.





- **the WLanguage STOP server keyword in your WLanguage code:** the debugger will start automatically when this line of code is run.
- **the "Debug project" option:** the debugger is started when running the test of the dynamic site. On the "Project" tab, in the "Test mode" group, expand Test mode and select "Debug project".

# When to use the debugger?

The debugger can always be used when running a test from the editor. Simply set a breakpoint at the desired location to debug the site step by step.

The different types of tests available from the editor are as follows:

- **"Go" on page:** The page in the editor can be tested via  in the quick access buttons of WEBDEV.
- **"Go" on project:** The project test on the development computer can be run via:
  -  found in the quick access buttons of WEBDEV,
  - on the "Project" tab, in the "Test mode" group, expand "Test mode" and select "Debug project".
- **The deployed project:** The deployed project can be debugged. This feature is very useful to reproduce a problem that occurs under specific conditions. To start the deployed project, on the "Project" tab, in the "Test mode" group, expand "Test mode" and select "Connect to a site and debug the connection". **Remark:** 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 being used. This feature is very useful to reproduce a problem that occurs under specific conditions. Most common case: a problem occurs while you are using the site: you can start the debugger immediately by setting a breakpoint in the code of your project. To use a session of deployed project, go to the "Project" tab, "Test mode" group, expand "Test mode" and select "Debug an existing connection". **Remark:** This option can be used with a development server or with a deployment server. The server must be configured to allow the remote debugging.

# Site test in practice

## 1 Overview

WEBDEV offers several methods for testing your sites:

- test of the entire project,
- test of a single page,
- test of a single query,
- test of a single report,
- step-by-step project execution,
- site performance test,
- regression test/automatic test,
- stress test.

A **test of the entire project** simulates the site startup. This makes it possible to test the entire site, even if its development is not complete yet. If a problem occurs, you can start the debugger to identify and fix the problem.

A **test of a single page** runs the current page only. This allows you to test of your project from a given page or to check the operating mode of a page as soon as its development is complete. Like for the project test, the debugger can be started if a problem occurs.

The **test of a single query** runs the current query only. This allows you to check the operating mode of a query once it has been developed.

The **test of a single report** runs the current report only. This allows you to test a report once it has been developed. Like for the project test, the debugger can be started if a problem occurs.

A **step-by-step project execution** allows you to launch the debugger when starting the site. This solution allows you to closely monitor how the site runs.

A **performance test** allows you to check and optimize the execution time of your site.

A **regression test (or automated test)** is based on the execution of scripts. It allows you to check that, during the execution of your site, ... the existing features are still supported.

A **load test** establishes multiple simultaneous connections to a dynamic WEBDEV site.

In addition to these methods, WEBDEV also includes the "Code coverage" of the site, i.e., the measurement of how much code is executed when the site is tested. For more details, see the online help.

## 2 Testing a WEBDEV project

### 2.1 Testing the project from the editor

Running the test from the editor allows you to test:

- the site features,
- the use of the site with different browsers.

The test of a project can be run regardless of the current element in the editor.

Remark: The test browser used to run the project test can be chosen:


- in the WEBDEV options: on the "Home" tab, in the "Environment" group, expand "Options" and select "General options of WEBDEV". The browser can be selected in the "Web" tab.
- in the test mode options: on the "Project" tab, in the "Test mode" group, expand "Test mode" and select "Test browser".

### Different types of tests

**To run the test of a static site from the editor:**

1. On the "Project" tab, in the "Test mode" group, expand "Test mode" and select "Debug project from the home page".
2. The editor is automatically minimized.
3. The browser specified in the WEBDEV options is opened and the site home page is displayed.

**To run the test of a dynamic site (Session or AWP) from the editor,** several methods are available:

- On the "Project" tab, in the "Test mode" group, expand "Test mode" and select "Debug project" (or press Ctrl + F9).
- Click  in the quick access buttons.

The editor is automatically minimized, the browser specified in the WEBDEV options is opened and the first site page is displayed.

**To test a static + dynamic site (Session or AWP) from the editor:**

- **to test the static part of the site:** perform the operations corresponding to the test of a static site.
- **to test the dynamic part of the site (Session or AWP):** perform the operations corresponding to the test of a dynamic site.

### Dynamic site (Session or AWP mode): Start

The following modules are automatically started during the test of a dynamic WEBDEV site (Session or AWP mode):

- 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 (WD280ADMIN.EXE).  
The administrator is used to manage the connections to the Web server and to configure the WEBDEV sites.  
**Remark:** a project test can be run from the test page of the administrator ("Advanced" tab of WD280ADMIN, "Test page" button).
- The WEBDEV engine (WD280AWP.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.  
**Remark:** The WEBDEV engine is started only if the project contains dynamic pages.
- The Internet browser.  
The Internet browser displays the HTML pages of the WEBDEV site.

### 2.2 Testing the project from the WEBDEV administrator

Running the test from the WEBDEV administrator (WD280Admin) allows you to test:

- the features of the site.
- the features specific to the Web (use of cookies, etc).

**Remark:** The WEBDEV administrator can only be used to test dynamic sites (Session or AWP) or the dynamic part of static + dynamic sites.

Running the test from the WEBDEV administrator is equivalent to starting the dynamic 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: on the "Tools" tab, in the "Web utilities" group, click "WAdmin".
2. In the "Advanced" tab of WEBDEV administrator, click the "Test page" button.

**To stop the test,** go to the WEBDEV administrator (click 28 in the taskbar) and click "Disconnect" ("Connections" tab).

**Remark:** The WEBDEV administrator also allows you to run a project test equivalent to the project test run from the editor:

1. Start the WEBDEV administrator: on the "Tools" tab, in the "Web utilities" group, click "WAdmin".
2. In the "Connection" tab, select the site and click the "Test" button.

### 2.3 Stress test/Regression test

WDTTestSite runs load tests: WDTTestSite is used to start several simultaneous connections to a dynamic WEBDEV site (Session or AWP).

Each connection performs a set of actions in the WEBDEV site (preset scenario).

For more details, see the online help.

### 3 Testing a remote WEBDEV site

Several methods can be used to run a test and to debug a site on the development computer. However, in some cases, you may have to debug the site directly on the 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 of the remote computer is required.

### 4 Testing a page

#### 4.1 Testing the page from the editor

To test a page from the editor:

1. Open the page to be tested.
2. Click  in the quick access buttons of the WEBDEV menu. You can also press F9.
3. The editor is automatically minimized and the page is run.

During the test, all page features can be run. You will have the ability to open other pages for example.

#### 4.2 Stopping the page test

There are multiple methods to stop the test:

- **1st method:**  
Close the page being tested. WEBDEV displays the editor open when the test was started.
- **2nd method:**
  - Go back to the editor via the taskbar or press Alt + Tab.
  - Stop the test. WEBDEV displays the editor open when the test was started.

### 5 Tracing a project

#### 5.1 Debugging principles

Debugging an application consists in:

- checking the operating mode of a process,
- understanding the operating mode of an existing program,
- checking the values of variables,
- checking the operating mode of special cases in an application or in a site.

The debugger is used to perform these operations.

**Remark:** WEBDEV also includes several trace tools (trace window, information box, etc.). For more details, see "Debugging without debugger", page 199.

#### 5.2 Debugger overview

The debugger monitors WLanguage programs so they can be improved.

The executed source code appears on the screen. The processes run are sorted in hierarchical order in the "Debugger" pane.

The value of the variables is displayed:

- individually in the on-hover tooltip of each variable.
- in the "Debugger" pane.

#### 5.3 Debugger features

The debugger is used to:

- view the call stack,
- view the content of variables or expressions,
- execute the code step by step with the possibility to skip code blocks,
- use conditional breakpoints,
- modify the code while continuing the execution,
- ...

#### 5.4 Debugging without debugger

In some cases, running a program with or without debugger may be different.

The debugger sets pauses in the execution of the processes, during which Windows performs multiple tasks.

Therefore, the debugger cannot be used in a procedure called by a timer or in the code of a scrollbar.

**Remark:** For more details on the limitation of the debugger, see the online help.

To debug these applications, you may want to follow the changes of a value, how different procedures are called, etc.

This information can be:

- displayed on the screen
- stored in a trace file.

**Caution:** If the information is displayed on the screen, it must be displayed during the application tests only.

##### Displaying information

Two tools can be used to display information:

- **information boxes:** WLanguage **Info** function.  
Caution: When displayed, dialog boxes block the application.
- **the trace window:** WLanguage **Trace** function.  
The trace window appears in the upper-left corner of the screen, without interrupting the program.

##### Displaying the debug information

Displaying the debug information on the screen is useful in test mode only.

Any unsuitable display must be removed before distributing an application.

To avoid any oversight, it is recommended to manage how the debug information is displayed 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 appears during the application test only.

This procedure prevents trace windows from being displayed on end-user computers, even if they are called in the code of the application.

The call to this trace procedure is identical to the use of **Trace**:

```
MyTrace ("Customer: " + ...
        Customer.CustomerNum)
```

##### Creating a trace file

During long processes (batch processing, etc.), to check the operating mode of the program, you must keep a physical trace of the processes run (a text file, for example).

The following procedure allows you to define how the trace will be displayed:

- 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 = DateTimeTrace.Date
        TimeTrace is Time
        TimeTrace = DateTimeTrace.Time
        fWriteLine (FileNum, ...
            DateToString (DateTrace) + ...
            " - " +
            TimeToString (TimeTrace))
        fWriteLine (FileNum, ...
            StringToTrace)
        fWriteLine (FileNum, " ")
        fClose (FileNum)
    END
END
```

Remarks:

- The trace file is created by default in the user data directory. The name of this file is the same as the project name. This file contains the information to trace during the program execution. The information is completed by the date and time of each "Trace". This allows you to detect a potential problem during the process.

- Example of trace file content:

```
01/12/2015 - 10:53:25:20
Customer name: Montgomery
```

## 6 Performance test

### 6.1 Overview

The performance profiler allows you to check and optimize the execution time of your site.

The principle is straightforward:

- You test your site.
- During the 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 tested site, sorted by duration (from the longest to the shortest action).

You can select a process to analyze its processing time and optimize it.

### 6.2 Starting the performance profiler

To start the performance profiler, go to the "Project" tab, "Audit and performance" group, expand "Analyze performance" and select "Analyze 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.

**Remark:** The performance profiler should be used to optimize your site (before it is distributed for example).

### 6.3 Reading the performance profiler results

The performance profiler presents the result of the analysis in several tabs:

- the "Summary" tab shows the ten longest processes.
- the "Mapping" tab shows a graphical view of the main processes.
- the "Details" tab shows all the processes run during the test of the application (from the slowest to the fastest).
- the "Calls" tab shows the details of the operations performed in a process.

The following information is displayed for each process:

Function	Function, process or procedure run.
Total time	Function execution window.
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.

**Remark:**

- The "Full execution" caption represents the total amount of time for running the site test 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 processes to be optimized are chosen based on several criteria:

- **the process execution time.** The longest processes must be optimized.

- **the percentage of time used by the function or procedure.** The higher this percentage is, the greater the number of processes that can be optimized in the code.

**Remark:** 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 (project Go or page Go) is used to immediately check a modification performed in your site.
- WDTTestSite that is used to run different tests on 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 allow you to easily check all the features available in your applications.

### 7.2 Automated tests

Each test contains a scenario that can be directly edited in the development environment. 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 tested:

- sets of procedures
- 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. Therefore, the number of tests has no impact on the size of the site available to end users.

For more details, see the online help (keyword: "Automated test").

### 7.3 WDTTestSite

WDTTestSite is used to run different tests on a WEBDEV site.

The following tests can be run by WDTTestSite:

- **Load 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.
- **Regression test:**  
The regression test consists in checking the operating mode of a WEBDEV site between two updates. The regression test consists in checking whether a scenario performed with an earlier site version 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 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.
- **Comparison of different servers:**  
WDTTestSite is used to 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 developed in WLanguage:**  
WDTTestSite is used to compare the execution time of a scenario before and after the WLanguage code was optimized.

For more details, see the online help (keyword: "WDTTestSite").

**PART 7**

**Deploying a  
Web site**

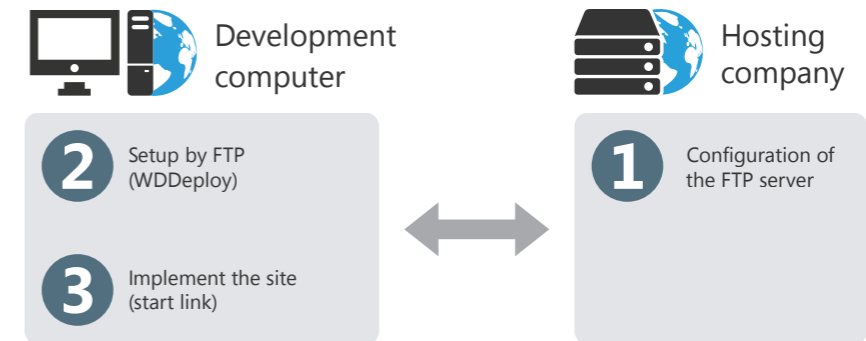
28

DEVELOP 10 TIMES FASTER

# Deploying a static site

## Principle

Static sites are 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 performing a deployment by FTP**, the hosting company must communicate the following information to the developer:

- Name of Web server where the setup will be performed (or its IP address).
- User name and password defined in the FTP server.

**When deploying by FTP**, the developer must specify in WDDeploy:

- the information provided by the hosting company.
- the files to install.

## Deploying a static site or a PHP site

### 1 Overview

When developing a static site or a PHP 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 accessible.

- **deployment by network:** recommended when the server is directly accessible.

WDDeploy is used to simplify the deployment of your static sites.

**Remark:** To deploy a dynamic site, see Deploying a dynamic site (Session or AWP) in practice.

### 2 Creating the setup

To deploy your static site:

1. On the "Project" tab, in the "Generation" group, click "Deploy site". WDDeploy starts.

2. Create the profile corresponding to your site.

This profile contains the following information:

- Location of local files ("`<ProjectName>_WEB`" subdirectory of your site).
- Address of WEBDEV site.
- Location of files of deployed site (network or FTP server).

- Characteristics of FTP server used to update the site. These characteristics are supplied 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.

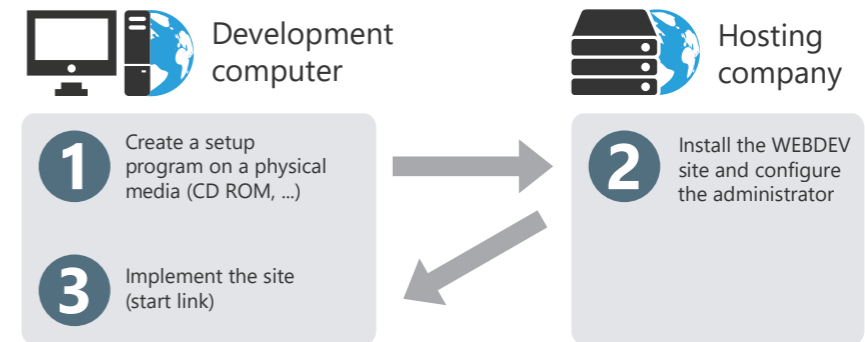
For more details, see the online help.

# Deploying a dynamic WEBDEV site

## The different types of deployment

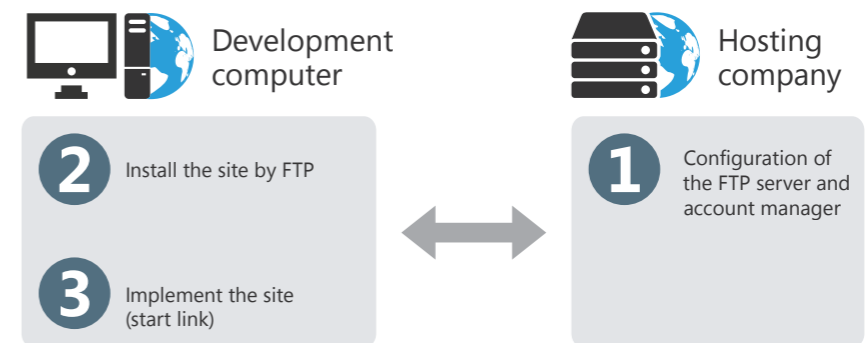
WEBDEV offers three methods for deploying a dynamic site (Session site or AWP):

- **deployment by physical media (CD-ROM, etc.)** of the WEBDEV site, with creation of a setup version supplied on CD-ROM to the hosting company.



**Remark:** a WEBDEV Application Server can be included in the setup supplied by physical media.

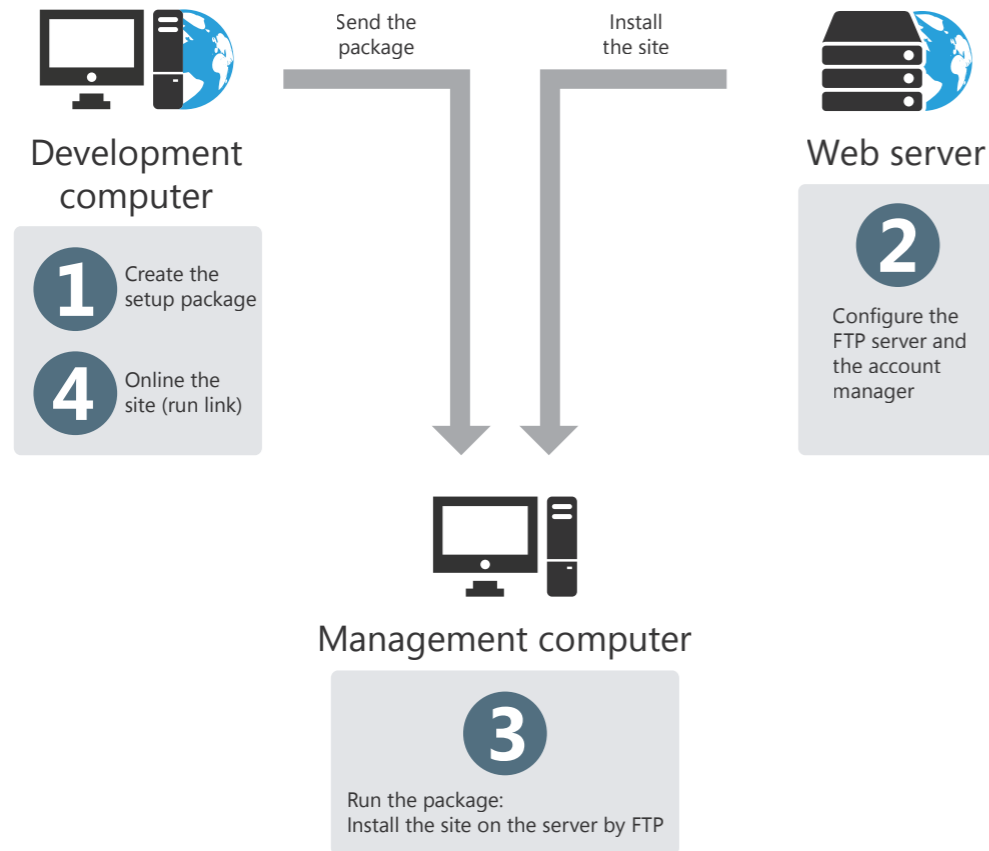
- **remote deployment directly from the development computer (by file transfer via Internet in HTTP mode).**



**Remark:** It is also possible to perform a deployment by file transmission in FTP mode. In this case, the hosting company must configure an FTP server.



- **remote deployment from a management computer (file transfer through Internet, FTP)**



**Note:** It is also possible to perform this deployment by file transmission in FTP mode. In this case, an FTP server must be configured on the Web server.

### Choosing a deployment method

The choice of deployment method mainly depends on the constraints imposed by the hosting company and on the location of 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 example).

**The remote setup (by HTTP) from the development computer is recommended** if the deployment computer cannot be easily accessed.

Before performing a remote setup, the hosting company must configure the WEBDEV account manager.

**The remote setup (by HTTP) from a management 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 server settings (address, etc.) are specified only when the package is run on the management computer.

### Deployment via physical media: necessary information

When deploying by physical media, the information about the setup and the site settings is both in the setup creation wizard and in the setup wizard.

**When preparing the setup,** you must supply:

- the setup directory,
- the files to install,
- the directory for generating the setup.

**During the setup,** you must supply:

- the directory of data files,
- the directory of site files.

**After setup,** the characteristics of the site (connection timeout, number of connections, etc.) must be configured in the WEBDEV administrator (deployment version).

### Deployment via HTTP directly from the development computer: necessary information

No setup can be run without this information. Ask the provider to get this information.

**Before performing an HTTP deployment directly from the development computer,** the host company must communicate the following information to the developer:

- Name of the Web server where the setup will be performed (or its IP address).
- Username and associated password defined in the WEBDEV account manager.
- The OS account used to run WEBDEV sites (it is associated with the WEBDEV account in the account manager).

When deploying by HTTP, the developer must specify in the wizard:

- the information provided by the hosting company,
- the files to install,
- the name of the subdirectory of data files,
- the different parameters for connecting to the site.

**Remarks:**

- A delayed setup can be performed at the specified date and time.
- The setup and the update by HTTP are secured by password and data encryption.

**Deployment via HTTP directly from a management computer: necessary information**

When creating the setup program, no specific information is required.

When running the setup program of the site (from a management computer), the hosting company will have to specify:

- Name of Web server where the setup will be performed (or its IP address).
- Username and associated password defined in the WEBDEV account manager.
- The OS account used to run WEBDEV sites (it is associated with the WEBDEV account in the account manager).

**Remarks:**

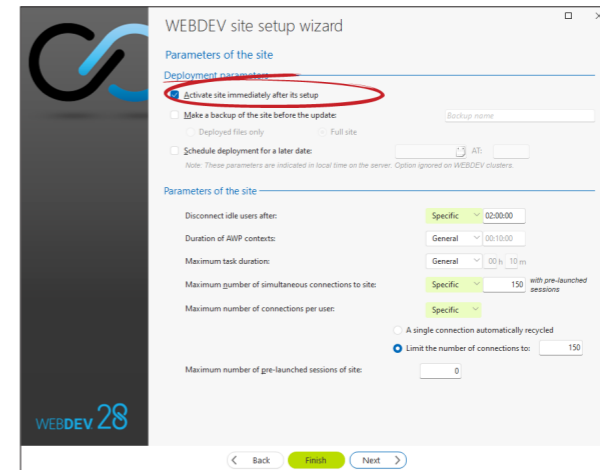
- A delayed setup can be performed at the specified date and time.
- The setup and the update by HTTP are secured by password and data encryption.

# Activating a dynamic WEBDEV site

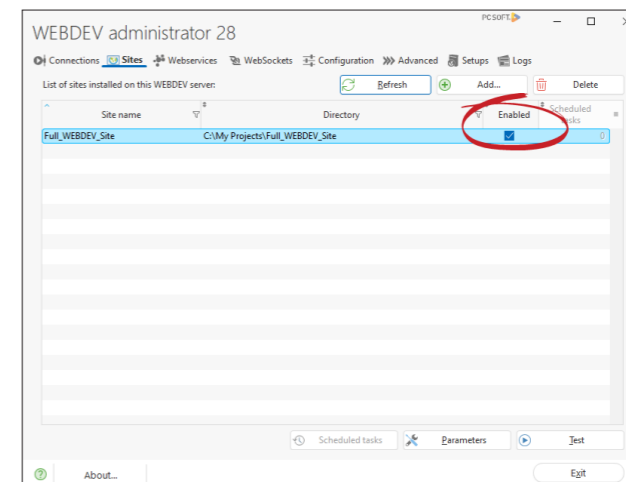
Your dynamic WEBDEV site (in Session or AWP mode) can be accessed as soon as it is on-lined. On-lining a WEBDEV site consists in activating the site.

Several methods can be used to activate a WEBDEV site:

- when performing the setup by HTTP: a check box can be used to automatically activate the site after being installed.



- with the remote administrator.
- with the WEBDEV administrator found on the deployment computer ("Locked" option in the "Sites" tab).



## Deploying a dynamic site in practice

When developing a dynamic site (in Session or AWP), one of the important phases consists of deploying this site on a Web server (at a hosting company or Intranet)

Several deployment methods are available:

- **deployment to PCSCloud:** Paid turnkey solution. For more details, see <http://www.pcsccloud.net>.
- **remote deployment (via HTTP) on a WEBDEV Application Server or WEBDEV Cluster:** recommended method when the server is not directly accessible.  
Remark: for backward compatibility, it is also possible to perform a remote deployment via FTP.

- **deployment by package:** recommended method when the server is not directly accessible and when its characteristics are unknown.
- **deployment by physical media:** recommended method when the server is easily accessible.
- **deployment via a Docker image.**

### Remarks:

- For each Web server that hosts WEBDEV sites, the hosting company must own a license for the WEBDEV Application Server.
- To create the setup of a site, go to the "Project" tab, "Generation" group, expand "Deploy site" and select "Generate and deploy site".

To deploy a static site, see Deploying a static site or a PHP site.

## 1 Generating a library

Before performing a setup, all the objects found in your dynamic site (in Session or AWP mode) must be included in a library.. A library is a file that contains all the objects created during development (database description, page description, compiled source code, etc.). 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:

- **deployment to PC SOFT Cloud:** PC SOFT offers a hosting platform based on the Cloud Computing model: you deploy without worrying about hardware constraints and you are billed based on your applications' real usage. A PC SOFT Cloud account is required to perform this type of setup. The wizard prompts you to create this Cloud account if necessary.
- **remote deployment (via HTTP or FTP) on a WEBDEV Application Server or WEBDEV Cluster:** Before performing a remote setup, the hosting company must have configured the web server (and the WEBDEV Cluster if necessary) and sent the parameters needed to install the site.

- **a remote deployment package:** A deployment package is an executable containing all the elements required to deploy a WEBDEV site. This package can then be executed from any computer to install the website on a remote WEBDEV Application Server or a WEBDEV Cluster (via HTTP or FTP). To create this type of setup, there is no need to know the characteristics of the WEBDEV accounts on the Web server.
- **a setup by physical media:** The wizard allows you to create a setup program for your site. Then, this program will be supplied to the hosting company and it will be used to install your WEBDEV site on the Web server. To install your WEBDEV site, the hosting company must:
  1. Install the WEBDEV Application Server on the Web server (if necessary).
  2. Install your WEBDEV site by running the *Install.exe* executable of your setup.**Remark:** It is also possible to create a setup as 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.

- **a setup via a Docker image.** This option allows you to create a Docker recipe including the site, the WEBDEV Application Server and the web server.
- **a setup via PC SOFT's test hosting service:** PC SOFT proposes to host your test sites. The

setup is easily performed, directly from the environment. Once the site is developed, you can test it directly.

A PC SOFT developer account is required to perform this type of setup. The wizard proposes to create this account if necessary.

## 2 Remote deployment (via HTTP or FTP)

### 2.1 Steps

The deployment steps are as follows:

1. Installing the WEBDEV Application Server at the hosting company (optional step). For more details, see the documentation about the WEBDEV Application Server.
2. Creation of a deployment account by the hosting company.  
Remark: If the installation is done via FTP (optional), an FTP account is created (in the FTP server).
3. The hosting company provides the names and associated passwords as well as the name (IP address) of the server.
4. Creating the remote setup from the editor: on the "Project" tab, in the "Generation" group, expand "Deploy site" and select "Generate and deploy site". In the wizard, select "Deploy to a remote WEBDEV Application Server or to a WEBDEV Cluster". The necessary files are copied into the transfer directory.
5. On-lining the WEBDEV site.

Caution: By default, a WEBDEV site is deployed remotely by HTTP.

### 2.2 Creating the setup

In the wizard, define:

1. **The information regarding the remote computer.** This information must be supplied by the hosting company:
  - **Server address** (name of a computer accessible by network, IP address or Internet address).
  - **Account** (username, password and account of the OS that runs the WEBDEV sites associated with the account).

- **Characteristics of the FTP account** if necessary (username 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, data files, images and HTML pages.

**3. Configuring the setup:** Chose the type of remote setup to perform:

- **Immediate 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.
- **Delayed 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 allows you to update your site during the night (between midnight and 3 AM for example), when the number of Web users is reduced.  
Caution: the specified date and time are the ones of the server. Make sure that they are valid.
- **Activating the site immediately after setup:** The site will be available once it has been installed: 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 activated 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 the analysis was updated, select the **automatic modification of data files.**

**Remark:** if this option is selected, the update will be proposed during the setup.

**Special case:** If the data files found on the server are in Hyper File 5.5 format, they must be migrated to HFSQL Classic format. This migration must be performed during the first site update.

**6.** Specify whether your site uses the **Native AS/400 Connector.**

**Reminder:** if your site uses the Native AS/400 Connector, 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 elements installed as well as their location:**

Site server	Data server
Directory of WEBDEV sites of user N	Data directory of user N
Directory defined by the hosting company. A subdirectory will be created in this directory for each site of user N.	Directory defined by the hosting company. A subdirectory will be created in this directory for the data files of each site of user N.
The HTML pages, the library ("WDL") and the image directory are installed in this subdirectory.	".FIC", ".NDX" and ".MMO" files are installed in this data subdirectory.
Example: The directory of 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 Steps

The deployment steps are as follows:

1. Creating the deployment package from the editor: on the "Project" tab, in the "Generation" group, expand "Deploy site" and select "Generate and deploy site". In the wizard, select "Create a remote deployment package". A setup file is created.
2. Sending the deployment package (via the Internet) to the web server administrator (hosting company, for example)

3. Installing the WEBDEV Application Server at the hosting company (optional step). For more details, see the documentation about the WEBDEV Application Server.

4. The hosting company creates a deployment account (via the Hosting Control Center) and an FTP account (in the FTP server) if necessary.
5. Running the deployment package on a computer and installing the WEBDEV site.
6. On-lining the WEBDEV site.

#### 3.2 Creating the setup

In the wizard, define:

**The files to install.** By default, the wizard selects the library, data files, images and HTML pages.

**1.** If the analysis was updated, select the **automatic modification of data files.**

**Remark:** if this option is selected, the update will be proposed during the setup.

**Special case:** If the data files found on the server are in Hyper File 5.5 format, they must be migrated to HFSQL Classic format. This migration must be performed during the first site update.

**2. 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.

**3.** Specify whether your site uses the **Native AS/400 Connector.**

**Reminder:** if your site uses the Native AS/400 Connector, 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, ...).

**4. The default profile of the Web server to use.**

These parameters can be entered or modified by the person who installs the deployment package.

**5. Configuring the setup:** Chose the type of remote setup to perform:

- **Delayed update:**  
The site files are compressed and immediately transferred onto the Web server (the Web server must be accessible from the computer that installs the package). The WEBDEV site will be installed on the server at the specified date and time. This allows you to update your site during the night (between midnight and 3 AM for example), when the number of Web users is reduced.  
Caution: the specified date and time are the ones of the server. Make sure that they are valid.
- **Activating the site immediately after setup:**  
The site will be available once it has been installed : 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 activated 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 server deployment account and FTP account, if necessary).
3. Validate the setup of the site.

### 4 Deployment by physical media

This deployment is performed in three steps:

1. Creating the setup version from the editor: on the "Project" tab, in the "Generation" group, expand "Deploy site" and select "Generate and deploy site". In the wizard, select "Create a setup by physical media".
2. Installing (if necessary) the WEBDEV Application Server on the Web server. For more details, see the documentation about the WEBDEV Application Server.
3. Installing the site at the hosting company.

#### 4.1 Creating the setup

In the wizard, define:

1. **The files of your site.**  
**Remark:** the library, HTML, Java, JavaScript and image files are selected by default.
2. The type of deployment: deployment with or without disconnecting the users.
3. The deployment parameters:
  - **Activating the site immediately after setup:**  
The site will be available once it has been installed: 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 activated via the remote administrator.

- **Delayed update:**

The WEBDEV site will be installed on the server at the specified date and time. This allows you to update your site during the night (between midnight and 3 AM for example), when the number of Web users is reduced. Caution: the specified date and time are the ones of the server. Make sure that they are valid.

**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 the analysis was updated, select the **automatic modification of data files**.

**Remark:** if this option is selected, the update will be proposed during the setup.

**6.** If programs not developed with WEBDEV must access the HFSQL files of your site, choose to install an ODBC driver.

**Remark:** if this option is selected, installing the driver will be proposed during the setup.

**7.** Specify whether your site uses the **Native AS/400 Connector**.

Reminder : if your site uses the Native AS/400 Connector, 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, ...).

**8.** Indicate if the web server should be included in the setup. In this case, you must select the web server to install.

**9.** Specify the setup customization options:

- if the setup should be customized.
- Default setup directory (site and data).
- Setup languages.
- Setup title. This title can be written in each language that can be selected during setup.
- License (txt file). This license will be displayed when setup is launched.

**Remark:** setup will be canceled if the license is not validated by the person trying to complete the setup process.

- Readme: 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.

**10.** Modify (if necessary) the **directory for generating** the setup.

By default, the setup is generated in the "INSTALL" subdirectory of your project.

**11.** Generate the setup.

An information window indicates if there are 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 files.

**12.** When the setup is created, WEBDEV proposes to:

- test the setup.
- open the Windows explorer in the directory for setup generation.

**13.** Perform a backup of your site.

We advise you to keep a backup of 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 (WD280ADMIN.EXE) is used to individually configure each WEBDEV site installed on this server.  
Caution: The memory resources of server are shared by all sites.
- **Install your site on a "shared" 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 operate properly.

**Remark:** For each Web server that is hosting WEBDEV sites, the hosting company must own a license for the WEBDEV application server.

**2.** To install the WEBDEV site, **INSTALL.EXE** (provided 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 step.

**4.** The setup automatically detects whether one or more of the following **Internet servers** are installed on the computer:

- **IIS Version 7 and later (Microsoft Internet Information Server):** Microsoft server.
  - **Apache Version 2.2 and 2.4:** free Web server supplied by the Apache Group.
- Three cases may occur:
- **Case 1 : your Web server is not displayed in this list:** select "Other server". For more details, see the online help about the server configuration.

- **Case 2: your Web server is displayed in the list but it is grayed:** 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 run the configuration later (to install your server later for example).

- **Case 3: your Web sever can be selected directly:** 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 configuration file will be ".reg", ".conf" or something else.

**5.** Select the setup directory:  
<webroot> indicates the root directory of the computer's Web server. Depending on the selected Web server, the root directory is detected and displayed. If the root directory of the Web server does not appear, you can type it in the gray area or find it via the "..." button.

**Caution:** if the area is filled, we advise you not to modify it.

**6. Updating the data files** (optional step, only for a site update).

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 currently used. Use the WEBDEV administrator to check that no current connection is in progress and to stop the sites (check "Site locked" in the "Sites" tab).

**7.** Install an **ODBC driver for HFSQL** if programs not developed with WEBDEV must access the HFSQL files of the site that will be installed (optional step).

**8.** The setup summary is displayed. Confirm your choices.

**9.** The setup is completed:

- Click the address that appears to test your site locally. This option can be used only if the Web server was automatically configured.
- Click "Copy" to retrieve the proposed address. For more details, see "On-lining the dynamic site", page 218.
- Click "Done" to end the setup.

**10. Configuring the Web server:**

If the configuration of the Web server was not performed automatically, two cases may occur:

- Case 1: your server was detected and a configuration file was generated: this file can be viewed, modified (if necessary) and run.
- Case 2: your server was not detected: for more details, see the online help.

**11.** Via the WEBDEV administrator, unlock the sites that may be locked and 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.

Test the WEBDEV site from the administrator via the [test page] option ("Advanced" tab).

**Remarks:**

- The WEBDEV administrator (WD280ADMIN.EXE) must be started and running in the background at all times. It can be installed as service.
- For more details about the WEBDEV administrator, see WEBDEV administrator in practice.

## 5 On-lining the dynamic site

To allow Internet users access to your site installed with a hosting company, create a link in your home page to launch your WEBDEV site.

If the setup was performed by physical media, the address to use was provided at the end of the setup. Replace <localhost> with the address of your site.

**Example used to start a "OURAPP" site from the PC SOFT site:**

```
http://www.windev.com/WD280AWP/  
WD280AWP.EXE/CONNECT/OURAPP
```

To start the site by using its address (prettyeyes.com for example), 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 towards the WEBDEV site, i.e., towards the following address, for example:

```
"http://205.51.231.57/WD280AWP/  
WD280AWP.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://205.51.231.57/  
WD280AWP/WD280AWP.EXE/CONNECT/  
prettyeyes">
```

**Special case: If a home page was defined in your dynamic site, the address for starting this home page is as follows:**

```
http://<ServerAddress>/<SiteName  
>_WEB/
```

# PART 8

## Hosting WEBDEV sites

# 28

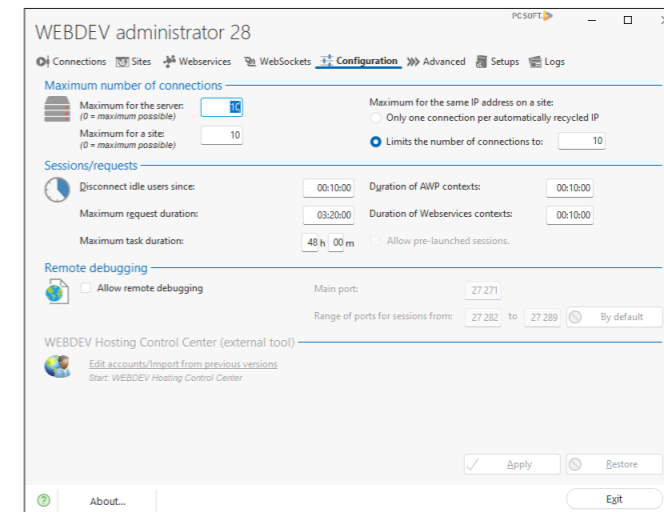
DEVELOP 10 TIMES FASTER

# What is the WEBDEV administrator used 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:

- test dynamic sites via the test page ("Advanced" tab, "Test page" button).
- manage webservices ("Webservices" tab).
- manage WebSockets servers ("WebSockets" tab).
- set the connection parameters to dynamic sites for testing: disconnection time, ... ("Configuration" tab).
- delete the current test connections ("Connections" tab).
- change the Web server used for tests ("Advanced" tab, "Server" option).
- perform a diagnostic if a problem occurs when starting a dynamic WEBDEV site ("Advanced" tab, "Diagnostic" button).
- allow remote debugging ("Configuration" tab).
- ...



In deployment, the WEBDEV administrator allows you to:

- manage the configuration of the different dynamic sites installed on the server: number of authorized connections, ... ("Sites" tab).
- customize error messages displayed in the different dynamic sites installed on the server ("Advanced" tab).
- configure WEBDEV clusters ("Cluster" tab).
- ...

# WEBDEV administrator in practice

## 1 Overview

The WEBDEV administrator is a module used to manage dynamic WEBDEV sites (Session or AWP).

When developing a dynamic WEBDEV site (in Session or AWP mode), the administrator is mainly used to:

- test the site via the test page.
- define the parameters for connecting to the site for tests: disconnection time, etc.
- delete the current test connections.
- change the Web server used for tests.
- perform a diagnosis if a problem occurs when starting a WEBDEV site.

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.
- **the remote WEBDEV administrator** that allows the site manager to manage their dynamic WEBDEV sites remotely from any computer.

## 2 WEBDEV administrator interface

The WEBDEV administrator includes a window containing several tabs:

- "Connections" tab.
- "Sites" tab and "Webservices" tab.
- "WebSockets" tab.
- "Configuration" tab.
- "Advanced" tab.
- "Setups" tab.
- "Logs" tab.

### 2.1 Characteristics of current connections

The "Connections" tab gives various information about the current connections. This information can be consulted at any time on the server. The following information is displayed in the "Connections" tab:

- the **number of current connections** indicates the total number of connections to the dynamic WEBDEV sites managed by the administrator.
- the **table of current connections** indicates for each connection:

This chapter presents only the administrator for WEBDEV Development version.

For more details about deployment tools, see the corresponding help pages.

### Starting the WEBDEV administrator

To start the WEBDEV Development administrator:

- from WEBDEV Development version: on the "Tools" tab, in the "Web utilities" group, click "WAdmin".
- select "Programs .. WEBDEV 28 .. WEBDEV administrator" from the "Start" menu.

The WEBDEV administrator is automatically started during the first test of the site.

The administrator can then be left running in the background.

By default, the WEBDEV administrator allows ten simultaneous connections to the same dynamic WEBDEV site (GO icon).

- the connection identifier,
- the site affected by the connection,
- the identity of the connected client. At runtime, the client is identified by IP address or Internet address,
- the current connection time,
- the idle time of the connection.
- the **Automatic refresh** option is used to automatically refresh the table presenting the current connections.

The "Connections" tab also allows you to:

- Test the sites installed on the computer: Select the site in the drop-down list box and click "Run the test". This mode is used test the options specific to the management of cookies on the development computer.
- Stop a current connection: Select one of the connections and click "Disconnect".
- Stop all current connections and close the administrator: click "Disconnect all".

## 2.2 Installed sites

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 site name.
- the full setup directory of the site on the computer.
- name of the corresponding "project" file.
- the site status (locked or not). You can check the box to directly block access to the site during the update.
- the number of scheduled tasks used by the site.

The "Sites" tab can also be used to:

- **Delete** the site selected in the table. It is used to remove the site from the administrator: the WEBDEV site will still be found on 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 after its setup.
- **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 can also create a specific configuration for the selected site ("Parameters" button). This configuration affects the number of authorized connections, the authorized idle time and whether the site must be locked during an update. For more details, see the "Configuration" tab. To validate the new configuration immediately, click "Apply". You can also manage scheduled tasks associated with the selected site (Scheduled tasks" button). This configuration is available when the site is deployed.
- Directly test the sites installed on the computer (development version only): Select the site and click "Test".

## 2.3 Installed Webservices

The "Webservices" tab gives information about the dynamic WEBDEV Webservices installed on the 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 complete Webservice installation directory on the workstation.
- name of the corresponding "project" file.
- the Webservice status (locked or not). The box can be checked to directly block access to the Webservice during the update.
- number of scheduled tasks used by the Webservice.

The "Webservices" tab also allows you to:

- **Delete** the selected Webservice from the table. The Webservice is deleted from the administrator: the WEBDEV Webservice will still be found on 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 after its setup.
- **Add** a Webservice into the table.
- **Configure** individually the Webservices on the computer. The configuration used by default is the one defined in the "Configuration" tab. You can also create a specific configuration for the selected Webservice ("Parameters" button). This configuration affects the number of authorized connections, the authorized idle time and whether the Webservice must be locked during an update. For more details, see the "Configuration" tab. To validate the new configuration immediately, click "Apply". You can also manage scheduled tasks associated with the selected Webservice (Scheduled tasks" button). This configuration is available when the Webservice is deployed.

## 2.4 WebSocket servers installed

The "WebSockets" tab lists the different WebSocket WEBDEV servers installed on the computer and managed by the WEBDEV administrator. For each WebSocket server, the following information is displayed in a table:

- the server name,
- the full setup directory of the WebSocket server on the computer,
- the WebSocket server status (locked or not). The box can be checked to directly block access to the WebSocket server during the update.



The "WebSockets" tab also allows you to:

- **Delete the selected WebSocket server(s) from the table** ("Delete" button). This only deletes the WebSocket server from the administrator (not the disk).
- **Refresh the display** of webservices found in the table ("Refresh" button). This refresh operation is used to check the presence of a webservice in the administrator after its setup.
- **Add a WebSocket server** in the table ("Add" button).

## 2.5 General configuration

The "Configuration" tab is used to define 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 sessions and requests.
- Managing remote debugging.

The following options can be used to configure the connections:

- **Maximum number of connections for 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 for 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 of a user to a site:** 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, the user who tries to connect simultaneously 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 x simultaneous connections. This number can be modified for each site ("Sites" tab). The "Only one connection per automatically recycled IP" option limits to one the number of connections of a user. The same connection will be reused. To allow multiple connections from the same IP address (users through a router or proxy), select "Limits the number of connections to" and specify the number of connections allowed for the same IP address.

The following options are used to configure sessions and requests:

- **Disconnect idle users since:** Maximum connection time allowed without any action performed by the 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 AWP contexts. As soon as the specified duration is over and if no new request was performed, the context file is deleted. For more details on AWP contexts, see AWP: Management of contexts.
- **Duration of Webservices contexts:** Validity period of webservice contexts. As soon as the specified duration is over and if no new request was performed, the context file is deleted. For more details on the management of webservice contexts, see Generating a webservice.
- **Maximum request duration:** Maximum timeout allowed between the beginning of an action performed by the 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.
- **Maximum task duration:** Used to limit the runtime duration of a WEBDEV scheduled task or delayed task. The task will be automatically stopped when this duration is exceeded. For more details, see WEBDEV: Scheduled tasks and delayed tasks.
- **Allow pre-launched sessions.:** This option is used to manage the pre-launched sessions on the WEBDEV Application Server. This option is used to optimize the connection time to the WEBDEV sites and Webservices found on the server. If this option is checked, you must:
  - modify the project to manage the pre-launched sessions properly. For more details, see Pre-launched sessions.
  - configure the site or the webservice in order to specify the number of necessary pre-launched sessions ("Sites" or "Webservices" tab, "Parameters" button).

## Managing the remote debugging

The WEBDEV administrator can allow (or not) the remote debugging of dynamic WEBDEV sites.

If the remote debugging is allowed, the following parameters must be specified:

- Main port (27 271 by default).
- Range of ports for sessions (between 27 282 and 27 289 by default).

The "Default" button is used to restore the default values.

## 2.6 Advanced options

The "Advanced" tab is used to configure the advanced features at server level:

- Manage the Web server:
  - Choose one of the Web servers installed on the current computer to run the WEBDEV sites.
  - Check the configuration of the current computer.
  - Specify the name or IP address of the current computer.
- Display a test page:
  - Display in the browser a page used to start all WEBDEV sites installed on the current computer.
  - Display in the browser a page used to start all WEBDEV webservices installed on the current computer.
- Perform a search for the expired pages.
- Manage the print jobs.
- Configure error messages displayed in the browser.
- Perform various configurations:
  - Manage the emails in asynchronous mode.
  - Manage the server sockets.
  - Forbid the change of IP while navigating.
  - Don't allow access to AWP context identifiers from JavaScript.
  - Manage the fCopyFileWebFolder and fDeleteFileWebFolder functions.
  - Require a secure connection (https) for remote management sites.

The following options are available for the WEB server:

- **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. For more details about the configuration of specific servers, see the online help.

- **Diagnostic:** "Diagnostic" checks the configuration of the current computer. This diagnosis checks:
  - whether the FTP/IP protocol is installed
  - whether a Web server (HTTP server) is started
  - whether the AWP protocol manager 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. For more details, see the online help.
- **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" opens a page in the browser to start each one of the dynamic WEBDEV sites and dynamic WEBDEV Webservices installed on this computer. For more details on testing a WEBDEV site, see 5. Site test in practice.

In the "Search for expired pages" section, you can search for expired pages. The "Allow searching for expired pages" option allows you to enable and configure the search for expired pages in WEBDEV sites handled by the WEBDEV Application Server.

In "Print jobs", you can choose the default printer (Intranet sties only). The "Print jobs" button is used to select the default printer that will be used when printing on a local printer or on a network printer of Web server.

In "Error messages", you can configure the error messages shown in the browser. The "Errors" button shows the different error messages that can be displayed in the user's browser. You have the ability to customize the error message and the HTML page where the error message is displayed.

The "Miscellaneous" section includes the following options:

- **Email management:** If the "Disable mail spool" option is unchecked, your sites will be able to send emails without blocking the execution of the different processes (asynchronous mode). The asynchronous mode must be enabled when starting the email session (with **EmailStartSMTPSession** or **EmailStartSession**). If the asynchronous mode is enabled, all outgoing emails will be redirected to a "Spooler". Emails are queued up before being sent. Executing Email functions does no longer block the rest of the program. **EmailStatus** is used to determine the status of an email.  
**Remark:** If the WEBDEV administrator is closed, the email spooler is cleared: pending emails are not sent and are removed from the spooler. If "Disable mail spool" is checked while there are pending emails, these emails will not be lost: the administrator continues to send them, but the spooler will not accept any new emails.  
**Caution:** The asynchronous mode can only be used when starting a session on an SMTP server (**EmailStartSMTPSession** to send emails or **EmailStartSession**). This mode is ignored in the other cases.
- **Management of Sockets:** If the "Allow server sockets" option is checked, your sites will be able to handle server sockets (via Socketxxx WLanguage functions).
- If the **Prevent from changing IP while browsing** option is checked, the IP address associated with the session cannot change while browsing. This is used to protect against "session hijack" attacks (attack that consists in pretending to be a legitimate user connected to the server).
- **Don't allow access to AWP context identifiers from JavaScript:** In an AWP site, the site context is stored on the server. The identifier of this context is sent and stored in the browser via a cookie.  
If "Don't allow access to AWP context identifiers from JavaScript" is checked, the type of cookies used will be "HTTPOnly", which cannot be read from JavaScript code. This mode protects against XSS attacks (Cross-Site scripting).  
By default, the access to the AWP context identifiers is not allowed from JavaScript.

- **Managing fCopyFileWebFolder and fDeleteFileWebFolder:** The "Allow fCopyFileWebFolder and fDeleteFileWebFolder." option must be checked if these functions are used in the website.  
These functions are mainly used to include images found in the data directory in the directory of site images (images uploaded then made available to the users, for example).  
Caution: The copy is taken into account by the WEBDEV administrator on the server (WD280ADMIN.EXE). The Windows account running it must have sufficient rights to the target location of the copy.
- The option "Require a secure connection (https) for the remote management sites" allows you to use the remote management sites provided with WEBDEV in secure mode. The affected management sites are:
  - the remote WEBDEV administrator,
  - the remote HFSQL administrator,
  - the remote SaaS administrator.

## 2.7 Setups/Accounts

The "Setups" tab allows you to:

- Configure the server for the site setups and updates.
- Manage the log of setups.

### Setup/Update

These options are available in deployment version. They allow the hosting company to authorize and configure the site setups and updates remotely.

The **"Lock server (for update)"** option simplifies site updates, by preventing any new connections. When new users try to connect to one of the sites on the server, they receive a message indicating that the site is temporarily unavailable and asking them to try again later. Web users who are already connected can continue to use the current WEBDEV site.

The option **"The root of the "http[s]://<server>/" server lists the installed applications"** allows the administrator to authorize the use of the "http[s]://<server>/" syntax to list the sites on the server. In this case, the "http[s]://<server>/" syntax shows a page with a link to each site installed on the server.

## Log file of setups

The log of setups is used to trace the installed elements of a site. This log is very useful before you contact Technical Support if errors occurred during the deployment.

To use the log of setups, simply specify the full path of the corresponding file ("WDInstall\_log.txt" file by default).

## 2.8 Logs

The "Logs" tab is used to:

- see the details of logs for a specific period.
- configure the sending of error reports by email.

**Caution:** To use this feature, the logs must have been enabled for the site. The logs can be enabled:

- in the "Logs" tab to create a general log,
- in the "Sites" tab ("Parameters" button) and in the "Webservices" tab ("Parameters" button) to manage a specific log.

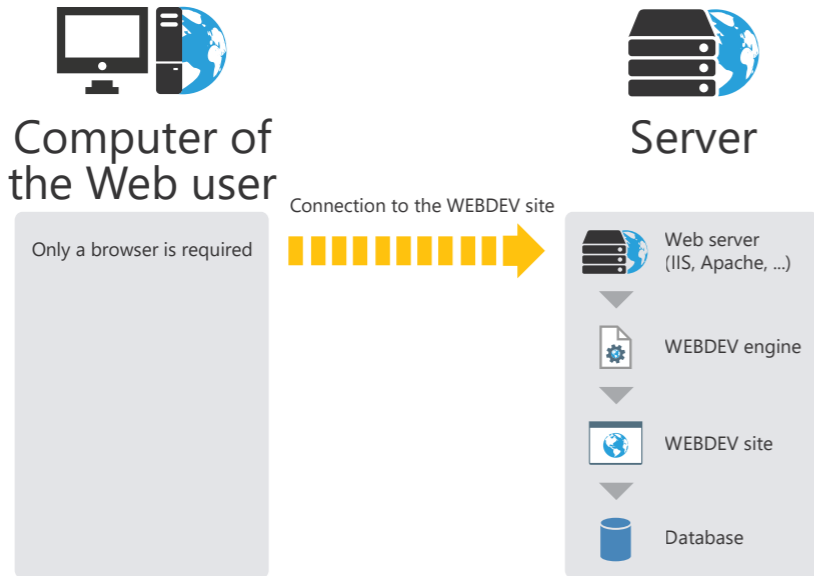
The error log is automatically created when the statistics log is enabled.

The WEBDEV administrator is used to manage a specific log file allowing you to follow the traffic statistics.

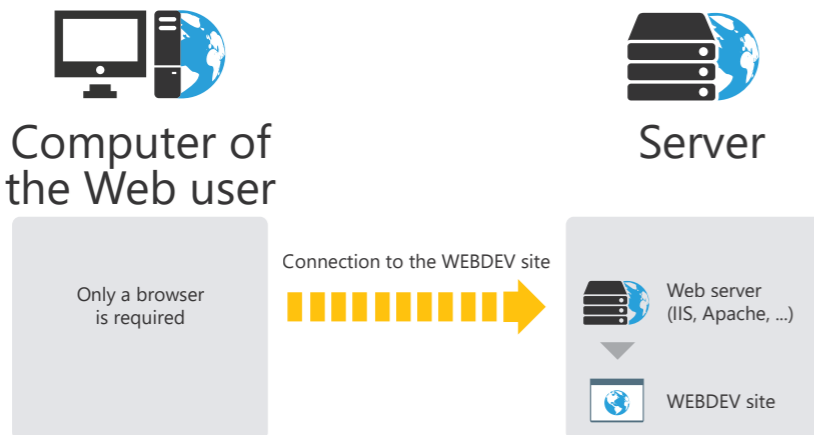
- **Generate a log file (.log):**  
This option allows you to generate a log file (".LOG" extension) used to see the traffic statistics of WEBDEV sites via WStatistic. For more details, see the help about WStatistic. The location of log file can be specified. This option can be modified for each site in the "Sites" tab.
- **Save in the global file and in the file of each application:**  
This option is used to save the statistics in the log files of both the application and the server. Indeed, if the log file is defined both at server level and at application level ("Sites" tab), information may be "lost".

# Configuring the server

## For a dynamic WEBDEV site



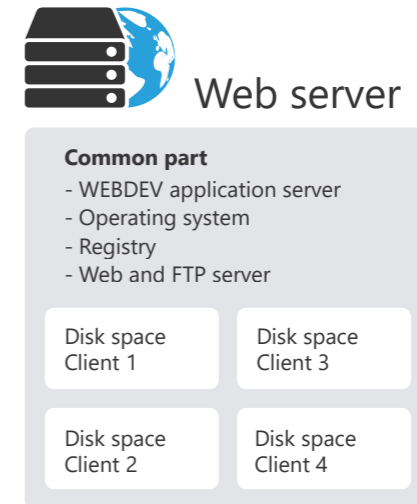
## For a static site:



# Dedicated server or shared server?

Hosting companies propose two types of servers for installing your WEBDEV sites:

- **shared server:**  
server whose resources are shared by several clients (client = hosted company).



### Characteristics:

The hosting company provides a server to several clients.

The hosting company:

- configures the FTP server (required only for FTP deployments).
- defines the maximum number of connections for all the dynamic sites belonging to the same client (via the WEBDEV account manager).

The setup and updates of a dynamic or static site can be performed by FTP.

- **dedicated server:** the server is reserved for the client. The hosting company provides the access to Internet (IP address, wire, etc.). The computer is either rented by the hosting company, or provided by the client.



**Computer dedicated to client**

- Web server and FTP server if necessary
- WEBDEV application server
- Operating system
- WEBDEV sites

**Characteristics:**

The hosting company provides the access to Internet (IP address, wire, etc.).

The client must:

- provide the server (or rent it),
- configure the computer (system, etc.),
- install the Web server,
- install and configure WEBDEV sites (via HTTP, FTP or DVD-ROM).

# Which type of server to choose?

The table below presents the benefits and drawbacks of different types of servers. Regardless of the type of server, a deployment license is required for each server. Each server hosts one or more dynamic WEBDEV sites (in Session or AWP mode).

Server	Benefits / Drawbacks	When to use this server?
Shared	<p><b>Main benefits:</b></p> <ul style="list-style-type: none"> <li>• Lower cost.</li> <li>• Server managed by the hosting company.</li> </ul> <p><b>Main drawbacks:</b></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 data: Unauthorized people can access the data: the hosting company 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>sites with few hits</b> (up to 20 or 30 simultaneous connections)</li> <li>• sites requiring little memory (no large calculations performed on the server),</li> <li>• Sites in startup phase.</li> </ul>
Dedicated	<p><b>Main benefits:</b></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,</li> <li>• WDREPLIC,</li> <li>• copy of files by FTP.</li> </ul> </li> </ul> <p><b>Main drawbacks:</b></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>• sites that require a lot of disk space and memory,</li> <li>• sites that manage sensitive data,</li> <li>• dynamic sites.</li> </ul>

# Hosting Control Center

The Hosting Control Center, which is mainly intended for hosting companies and Webmasters, helps you host WEBDEV sites.

The Control Center manages the WEBDEV accounts as well as the account for the Web server and the rights for the operating system.

A default choice is proposed: it allows you to easily install a server, without specific knowledge.

The configuration of the following elements is centralized:

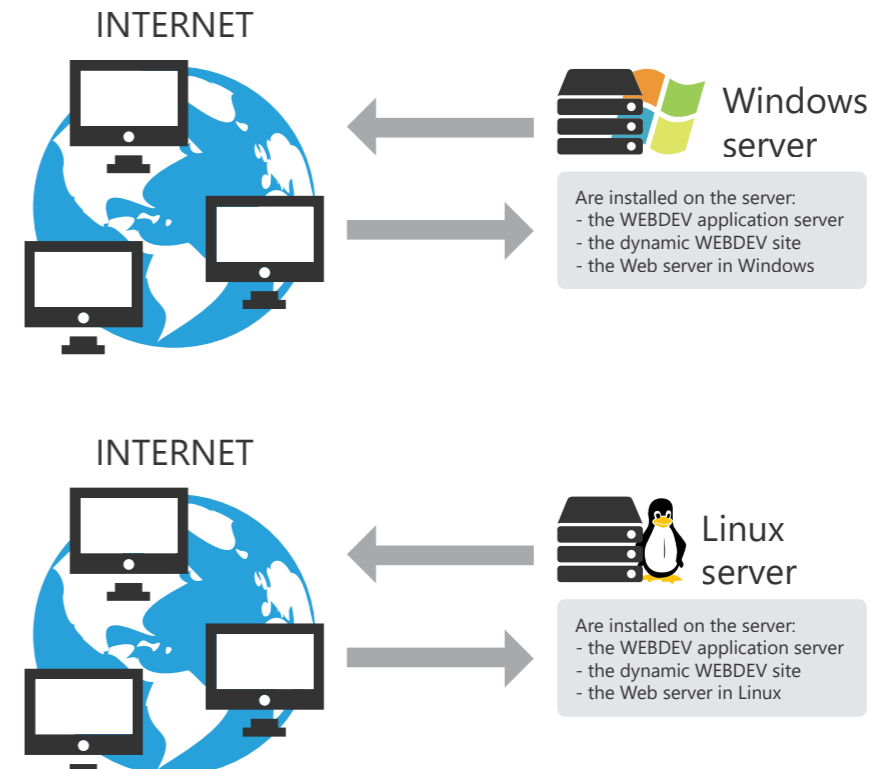
- Windows accounts,
- Groups of FTP users,
- Groups of WEBDEV clients,
- Home Directory,
- FTP alias,
- WEBDEV accounts,
- Data directories,
- Virtual Web sites, ...

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

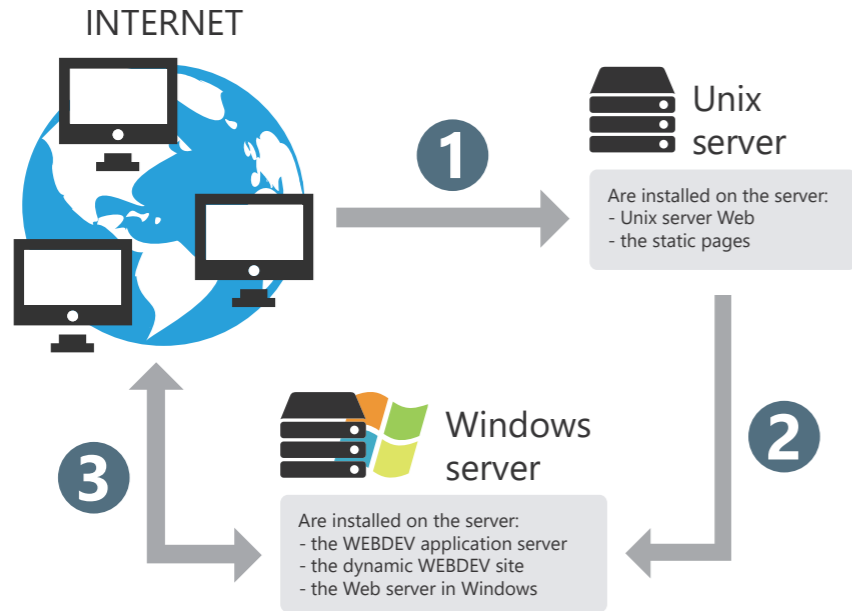
## 1. Setup on a Web server in Windows (2000 or later) or a Linux server

The configuration is as follows:



### 2. Setup on a Windows server with an access to Internet via UNIX

The configuration is as follows:

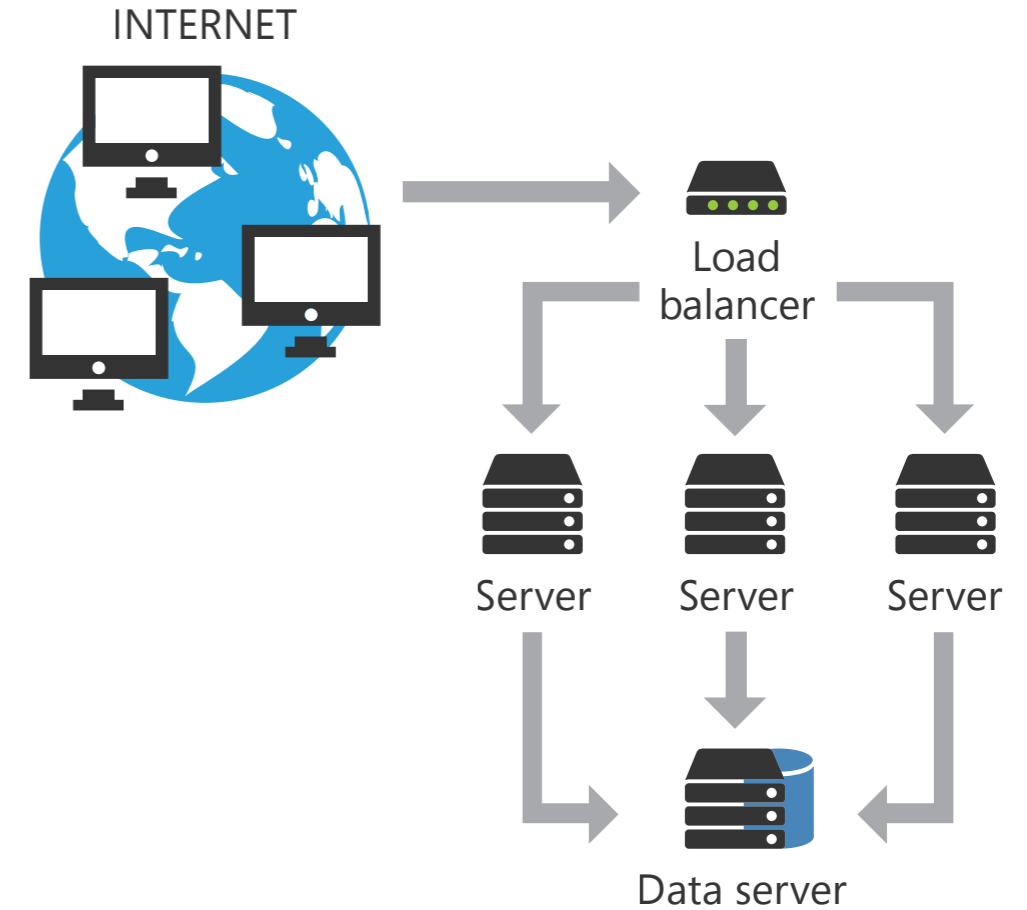


1. Opening a static page. The static page is on the UNIX server.
2. Starting the dynamic WEBDEV site (located on the Windows server) via a link in the static HTML page.
3. Standard operating mode of the WEBDEV site.

**Remark:** the same configuration can be defined on 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.

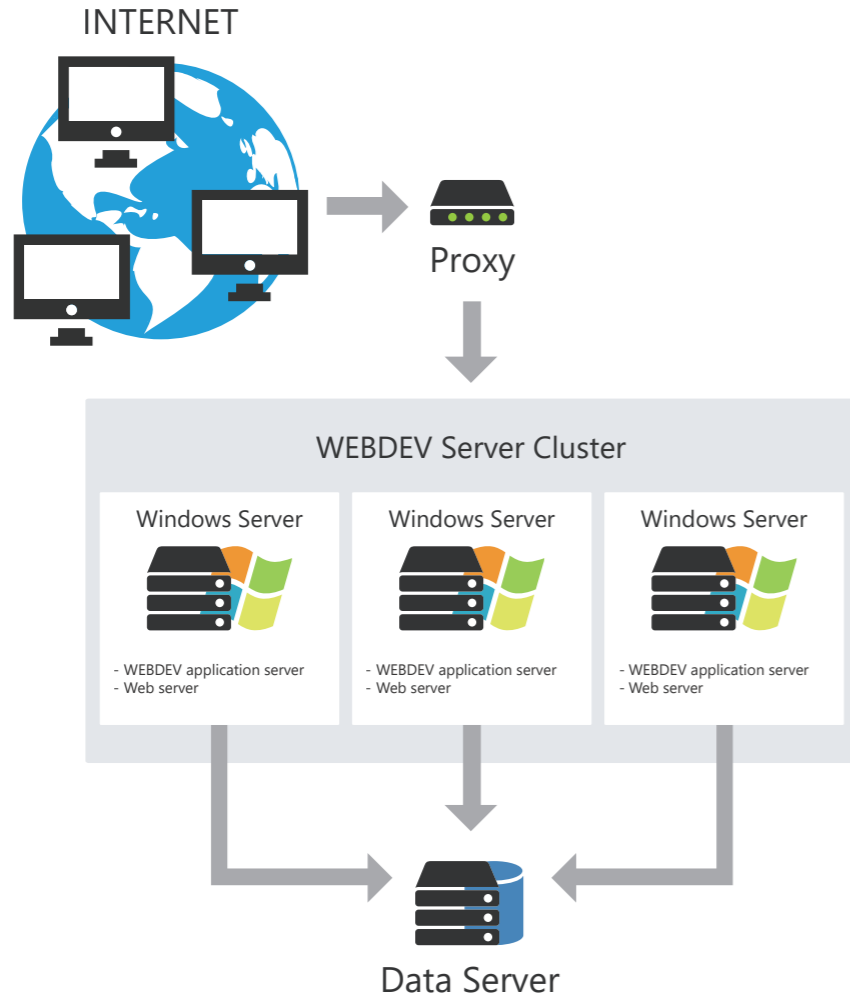
#### Solution 2: Installing the dynamic WEBDEV site on all Web servers.

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.

#### 4. Setup on a WEBDEV cluster

The system of a WEBDEV cluster allows you to distribute the connection load of a dynamic WEBDEV site across different servers in a WEBDEV cluster.



## Traffic statistics of dynamic sites

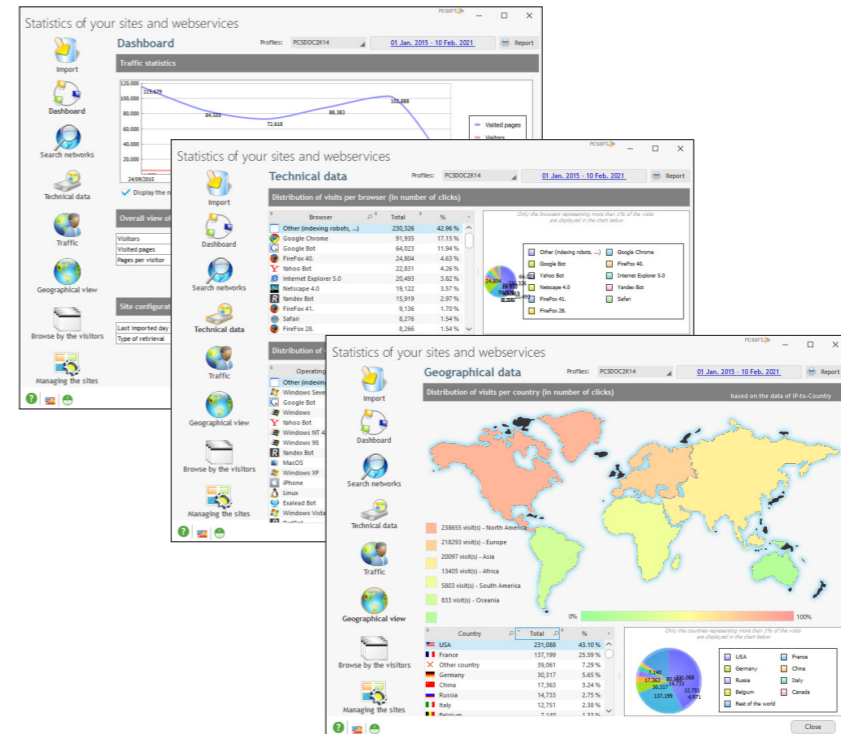
It is often important to get accurate statistics about the site visits.

How to determine who visited your site, which pages were viewed, which applications were used, and which sites the users came from?

WEBDEV includes a statistics tool for dynamic pages: WDStatistics. This tool can be installed on any computer.

Examples of statistics calculated by WDStatistic: number of connections per day, visited pages, operating systems and browsers of Web users, ...

Examples of statistics of dynamic sites:



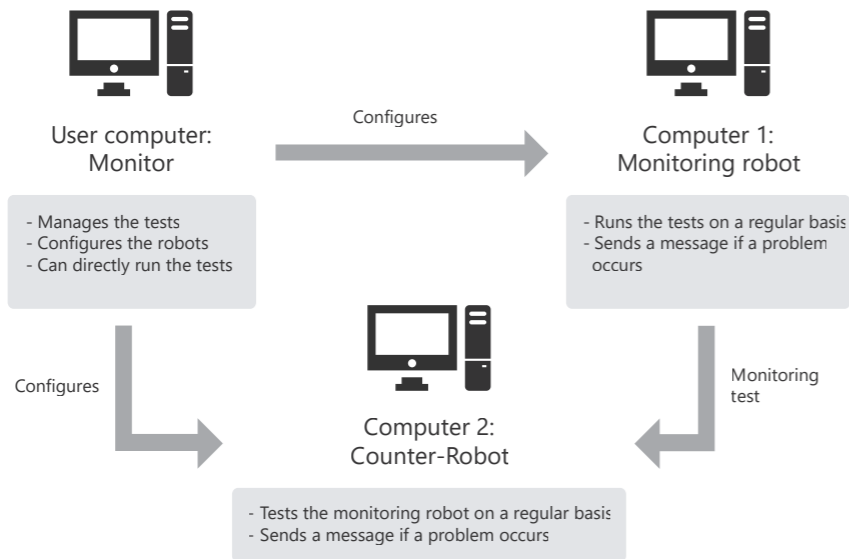
# Monitor your sites, servers, etc.

Most sites rely on many external factors: Web site, HFSQL server, network, ... To optimize the management of incidents, WEBDEV allows you to use a monitoring robot.

The monitoring robot, which is composed of three executables running on different computers, is used to run different tests: Internet tests, network tests, ...

If a problem occurs during a test, the monitoring robot sends you a message using one of the following methods:

- Message sent via PC SOFT's messaging tool (WDBAL).
- Email
- Via a third-party program. Additionally, a sound alert can be set on the monitor.



### Available types of tests

Internet test (test of a Web site)	Test of HFSQL Client/Server servers	Network test (SNMP)
Test of FTP server	Test of news server	Test of WINDEV application
Custom test (in WLanguage)	...	





# WLanguage functions specific to WEBDEV 28

AddFavorite	Adds an Internet address into the list of favorites for the Web user.
AJAXAsynchronous CallPending	Used to find out whether a server procedure called by <b>AJAXExecuteAsynchronous</b> is currently run.
AJAXAvailable	Used to find out whether the AJAX technology is supported by the current browser.
AJAXCancelAsynchronous Call	Cancels the automatic execution of the browser procedure called by <b>AJAXExecuteAsynchronous</b> .
AJAXExecute	Runs a server procedure without refreshing the page.
AJAXExecuteAsynchronous	Runs a server procedure without refreshing the page.
AppServerAddScheduledTask	Adds a new scheduled task to the application (site or webservice).
AppServerDeleteScheduledTask	Deletes a scheduled task associated with the application (site or Webservice).
AppServerInfoScheduledTask	Reads the description of a scheduled task.
AppServerListScheduledTask	Returns the list of scheduled tasks for the application (site or Webservice).
AppServerModifyScheduledTask	Modifies an existing scheduled task.
AppServerRunScheduledTask	Immediately starts the execution of a scheduled task on a WEBDEV Application Server.
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.
BannerFirst	Displays the first plane of a Sliding Banner control.
BannerNext	Displays the next plane of a Sliding Banner control.
BannerPrevious	Displays the previous plane of a Sliding Banner control.
BannerStopSliding	Stops the automatic slide on a Sliding Banner control.
BreadcrumbAddLink	Adds a link into a Breadcrumb control.

BreadcrumbDeleteAll	Deletes all links from a Breadcrumb control.
BreadcrumbDeleteLink	Deletes a link from a Breadcrumb control.
BreadcrumbInsertLink	Inserts a link into a Breadcrumb control.
BreadcrumbModifyLink	Modifies a link in a Breadcrumb control.
BrowserClose	Closes the current browser window and stops the execution of the current code.
BrowserHeight	Returns the height (in pixels) of the client area where the page is displayed.
BrowserHistoryAdd	Adds an entry into the history of navigation by associating data. This data will be transmitted when going back to this entry.
BrowserHistoryModify	Modifies the data of current entry in the navigation history. This data will be transmitted when going back to the current entry.
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.
BrowserMobile	Allows you to find out whether the browser used by the Web user is started on a mobile device.
BrowserName	Returns the name of the browser of the Web user ("user agent").
BrowserOpen	Opens a new browser window.
BrowserOS	Indicates the operating system installed on the browser of the Web user.
BrowserPlatform	Returns the platform of the browser used by the Web user.
BrowserPrint	Prints the page currently displayed in the browser.
BrowserRunApp	Opens the default Web browser of the current device.
BrowserSlice	Returns the number of the slice displayed in a page in Responsive Web Design mode.
BrowserType	Returns the type of browser used by the Web user.
BrowserVersion	Returns the version of the browser used by the Web user.

BrowserWidth	Returns the width (in pixels) of the client area where the page is displayed.
CancelAWPContext	Deletes from the AWP context a variable that was added by <b>DeclareAWPContext</b> .
CapsLockVerify	Checks whether the CapsLock key is pressed.
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 control previously displayed in the page via <b>CellDisplayDialog</b> .
CellDisplayDialog	Displays a Cell control in a page with a DDW effect (Dim Disabled Windows). This function is used to easily simulate a dialog box in browser code, by using a Cell control in a page.
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 server.
ChangeTarget	Changes the target of a button action.
ColorPalette	Returns a color found in the current palette.
ConfigureAWPContext	Configures the operating mode of AWP context.
ConnectionCount	Returns the number of instances of the WEBDEV site currently run on the server.
ContextClose	Closes a page context.
ContextExist	Allows you to find out whether a page context exists on the server (which means whether the page was opened).
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 Web user.
CookieWrite	Writes a cookie onto the computer of the Web user.
CurrentPage	Returns the name of the page containing the WLanguage code currently run.

DeclareAWPContext	Used to declare a list of variables whose value will be persistent between the successive displays of AWP pages.
DynamicServingConfigure	Indicates the environment that will be used by the "Dynamic Serving" to choose the set of pages to display.
DynamicSiteDisplay	Starts a dynamic WEBDEV site from a static page or or from a dynamic WEBDEV page.
EmailOpenMail	Opens the default messaging software: <ul style="list-style-type: none"> <li>• of Web user on the browser computer.</li> <li>• of the user on the current Windows computer.</li> <li>• on the phone.</li> </ul>
EmailStatus	Returns the status of an email sent via an SMTP session started in asynchronous mode.
ExecuteDelayedProcedure	Runs a delayed procedure (in the WEBDEV application server).
fCopyFileWebFolder	Copies an image file from the data directory of application (or from one of its sub-directories) to the "_WEB" directory of application (or to one of its sub-directories).
fDeleteFileWebFolder	Deletes an image file from the "_WEB" directory of the application (or from one of its sub-directories).
FileDisplay	Displays a file in the browser of Web user.
FileToPage	Automatically updates the controls of a page with: <ul style="list-style-type: none"> <li>• the values of the bound items in the current record (loaded in memory) of the data file described in the data model editor. This operation is performed regardless of the state of the controls (grayed, inactive or invisible).</li> <li>• the values of the bound WLanguage variables. This operation is performed regardless of the state of the controls (grayed, inactive or invisible).</li> </ul>
FolderData	FolderData is kept for backward compatibility.
FolderWeb	Returns the path of the directory containing: <ul style="list-style-type: none"> <li>• the images,</li> <li>• the JavaScript files,</li> <li>• the Java applet files,</li> <li>• the other files accessible from the browser.</li> </ul>
FramesetDisplay	Displays a WEBDEV frameset in the browser of Web user.

FramesetUse	Displays a WEBDEV frameset in the browser of Web user and closes all the current page and frameset contexts.
FreeAWPContext	Frees the AWP context in advance (on disk) to allow the other call on the same AWP context to be processed in parallel.
FullScreenDisable	Disables the "Full screen" mode of browser.
FullScreenEnable	Switches the browser to "Full screen" mode. The browser occupies the entire screen.
fWebDir	Returns the physical name of the directory containing the images, the JavaScript files and the Java files of the WEBDEV site.
GglAnalyticsAddEvent	Adds an event beside Google Analytics
GglAnalyticsAddException	Adds an exception beside Google Analytics.
GglAnalyticsAddPage	Adds a page beside Google Analytics.
GglAnalyticsAddSocial-NetworkAction	Adds a social network action beside Google Analytics.
GglAnalyticsAddTiming	Adds a duration beside Google Analytics.
gpwOpenConfiguration	Opens the user groupware configuration window or page.
grImageSize	Defines the size of the image containing the chart.
HTMLClassAdd	Adds a class into the HTML classes of a control.
HTMLClassDelete	Deletes a class from the HTML classes of a control.
HTMLClassToggle	Toggles a class in the HTML classes of a control: if the class does not exist, it is added; if the class already exists, it is deleted.
IdentifierAWPContext	Returns the identifier of AWP context.
iDirImageHTML	Used to select the directory in which the images must be generated during a print in HTML format.
ImageArea	Returns the number of the image area clicked by the Web user.
ImageFirst	Displays the first image for the scroll.
ImageLast	Displays the last image for the scroll.
ImageNext	Displays the next image for the scroll.

ImageOccurrence	Returns the total number of images taken into account by the scroll.
ImagePrevious	Displays the previous image for the scroll.
ImageScrollingPosition	Returns the displayed image.
ImageStartScrolling	Starts the automatic scroll of images.
ImageStopScrolling	Stops the automatic scroll of images.
ImageXPos	Returns the horizontal position (X) of the mouse cursor in relation to the specified Image control.
ImageYPos	Returns the vertical position (Y) of the mouse cursor in relation to the specified Clickable Image control.
jQuery	Runs a JavaScript method (or several chained methods) of the jQuery library on a page element.
jQueryExecute	Runs a JavaScript method of the jQuery library on a page element.
JSEndEvent	Deletes the association between a WLanguage browser function and an event (implemented by <b>JSEvent</b> ).
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 code execution.
JSInterruptEvent	Interrupts the process of current event.
JSMethod	Used to run a JavaScript method on an element found in the current page.
JSONExecute	Calls a server URL that returns data in JSON format (JavaScript Object Notation).
JSONExecuteExternal	Calls an external server URL that returns data in JSON format (JavaScript Object Notation).
JSProperty	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 or not for the current browser.

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.
MenuAddPopup	Transforms a menu option of a page in order for this option to open a popup.
MenuAddURLOption	Adds a new menu option at the end of a page menu. This menu option is used to display the page corresponding to the specified URL.
PageActivateDDW	Enables or disables the DDW (Dim Disabled Window) when displaying a modal page (the pages used to communicate with the user for example).
PageAddress	Used to find out the Internet address of a page.
PageChangeLayout	Applies a specific layout to a page.
PageCloseDialog	Closes the current page. This page was opened by <b>PageDisplayDialog</b> . A return value can be returned to the calling page.
PageCurrentLayout	Returns the index of the current layout: <ul style="list-style-type: none"> <li>• in the current page.</li> <li>• in the specified page.</li> </ul>
PageDisplay	Displays a site page in the browser of Web user.
PageDisplayDialog	Displays a page in modal mode.
PageExist	Checks whether the specified page is currently displayed in the browser of the Web user.
PageInitialization	Resets to zero (or not) the controls found in the current page and starts the events for initializing the controls and the page.
PageParameter	Returns the value of a parameter passed to the current page.
PagePosition	Scrolls a page up to position a control in the visible section of the page (top) in the browser.
PageRateDDW	Allows you to find out and modify the rate of gray for the pages that use DDW (Dim disabled pages).

PageReadHeaderHTTP	Reads and returns one or more HTTP headers received by the current page (these headers are sent by the client to the server in the current request).
PageRefresh	Refreshes a page displayed in the browser of 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 updates: <ul style="list-style-type: none"> <li>• the value of the items in a data file with the value of the controls in the page.</li> <li>• the value of the WLanguage variables with the value of the page controls.</li> </ul>
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 updates: <ul style="list-style-type: none"> <li>• the value of the items in a data file with the value of the controls in the page.</li> <li>• the value of the WLanguage variables with the value of the page controls.</li> </ul>
PageUse	Displays a WEBDEV page in the user's browser and closes all page contexts currently open on the server.
PageVisible	Indicates whether the page is visible to the user.
PageWriteHeaderHTTP	Adds an additional HTTP header that will be returned to the browser: <ul style="list-style-type: none"> <li>• during the next function for page display (<b>PageDisplay</b>, <b>PageRefresh</b>, ...),</li> <li>• during the automatic return of the current page at the end of the current action.</li> </ul>
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.
PopupAnimate	Displays a popup in a page cell.
PopupClose	Hides a popup displayed in the page via <b>PopupDisplay</b> .
PopupClosePage	Hides a page displayed as a popup via <b>PopupDisplayPage</b> .
PopupDisplay	Displays a popup in a page with a DDW effect (Dim Disabled Windows).
PopupDisplayPage	Displays a WEBDEV page as a popup in the current page with a DDW (Dim Disabled Windows) effect.
PreviousPage	Returns the name of the previous page.
ProgressBarDisable	Stops the periodic refresh of a Progress Bar via Ajax.
ProgressBarEnable	Enables the refresh of a Progress Bar via Ajax.
ProgressBarExecute	Starts a long browser process and fill a Progress Bar via Ajax according to the progress of this process.
rssDisplay	Builds a RSS stream and returns the content of RSS stream to the client.
SaaSCheckService	Checks the access rights of the user to a service of the SaaS site.
SaaSClientConnexion	Returns the characteristics of the connection to the client database.
SaaSConnect	Authenticates a user of a SaaS site beside the SaaS webservice that manages the site.
SaaSConnectedUser	Returns the user currently connected via <b>SaaSConnect</b> .
SaaSDisconnect	Disconnect the user of a SaaS site beside 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 or page (.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.
SessionPrologue	Enables or disables a WLanguage procedure called on each request in a session.
SiteAddress	Returns the Internet address for connecting to a dynamic WEBDEV site (in Session mode) found on the same server.
SourceToPage	Automatically updates the controls of a page with: <ul style="list-style-type: none"> <li>• the values of the bound items in the current record (loaded in memory) of the HFSL data file described in the data model editor.</li> <li>• the values of the bound WLanguage variables.</li> </ul>
SSLActive	Used to enable or disable the secure 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.
ToastDisplayPopup	Displays a popup page during a given duration in order to display a "Toast" message.
UploadCopyFile	Saves on the server a file "uploaded" by the Web user. This upload may have been performed: <ul style="list-style-type: none"> <li>• via an "Upload" edit control.</li> <li>• via a single-file or multi-file Upload control.</li> </ul>
UploadCurrentFile	Indicates the file currently uploaded via the Upload control.
UploadCurrentFileSize	Returns the total size (in bytes) of 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. This upload may have been performed: <ul style="list-style-type: none"> <li>• via an "Upload" edit control.</li> <li>• via a single-file or multi-file Upload control.</li> </ul>
UploadFileSize	Returns the total size (in bytes) of a file found in an Upload control.
UploadSize	Returns the total size (in bytes) of the file currently uploaded via an Upload control.
UploadSizeSent	Returns the total size (in bytes) of files already sent by the current upload via an Upload control.
UploadStart	Starts sending the selected files into an Upload control.

# Examples and components provided with WEBDEV

The examples provided with WEBDEV are intended to help you learn the features of WEBDEV. Their source code is clear and includes comments.

Different types of examples are provided with WEBDEV:

- complete examples: these examples correspond to full sites that can be used without any adaptation.
- training examples: these examples illustrate a specific feature.
- unit examples: these examples include a page that can easily be tested in the current project. This page presents the use of a function, group of functions, control, etc.
- components: -these examples include an internal component, an external component and a use example.

These examples and components can be opened from the WEBDEV home window.

- If the example is not installed on the computer, it will be automatically downloaded (from a PC SOFT server) and opened.
- If the example is installed on the computer, a copy is automatically created in "My Projects\My examples".

The open example corresponds to the copy of the original example. This allows you to work or make changes to this copy. The original example can be kept unchanged. Every time you open the example (via the "Open an example" option), you can work on the copy or on the original version.

You can find additional examples on our website ([www.windev.com](http://www.windev.com)).

## Full examples

eBusiness	Full eCommerce site that can be customized via a Web management interface. <ul style="list-style-type: none"> <li>• The showroom/payment section of the site is developed in AWP in order to get the best possible referencing of products.</li> <li>• The administration section of the site is developed in standard WEBDEV session to guarantee the maximum security.</li> </ul>
Honolulu	Free Intranet portal: messaging, forum, blogs, ...
Photo_Gallery	This example is a photo gallery site and is composed of 2 main parts: <ul style="list-style-type: none"> <li>- the "visitor" part, developed in Active WEBDEV Page mode and SEO-compatible.</li> <li>- the administration part, secured and developed in standard WEBDEV mode.</li> </ul>
WAds	Management of classified ads. Multi-criteria search, addition, modification and deletion of classifieds, as well user management.

webmillion	Uses a data file containing more than 1 million records.
WW_Association_AWP	Association site made of 2 main sections: <ul style="list-style-type: none"> <li>• the "visitor" section, built in AWP mode therefore can be referenced.</li> <li>• the "member space" section, built with WEBDEV in standard mode and therefore secured.</li> </ul> This site presents news, a photo gallery, the association overview, ...
WW_Association_PHP	Association site in PHP WEBDEV made of 2 main sections: <ul style="list-style-type: none"> <li>• the visitor section, accessible to all.</li> <li>• the management section, reserved to the members of association.</li> </ul> This site presents news, a photo gallery, the association overview, ...
WW_Blog_AWP	This example is a site for blog management, based on an AWP generation, allowing the site to be referenced by all search engines. Furthermore, each blog can be exported in RSS.
WW_Blogs_php	This example is a PHP site for blog management.
WW_CMS	This example is a CMS (Content Management System), typically a site for displaying articles.
WW_FAQ	This Responsive Web Design example proposes the main features of a FAQ site.
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_PHP	This example proposes the main features of a user forum (creation of forums, topics and messages, moderation by an administrator, ...).
WW_Loan	This example is used to simulate loans and to calculate: <ul style="list-style-type: none"> <li>• the amount of monthly payments for a given loan.</li> <li>• the amount that can be borrowed for a given monthly payment.</li> <li>• the income of an investment for a given monthly payment.</li> </ul> The amortization table is displayed for each case.
WW_MotorEquipment	This example represents an online store. It contains the "front-office" section (overview of products), the basket section, the secure payment.
WW_Newsletter	Used to create a site for managing newsletters. It allows you to create, write, and manage the sending of mass newsletters to subscribers.
WW_Overview_Mobile_Application	Responsive Web site presenting a Mobile application that includes a static page that is entirely responsive.
WW_Precilia_Sport	This example is a Web site with a store section. The site occupies the full browser width with a background image.
WW_Precilia_Winter	This example is an online site for skiing equipment.

WW_PreciliaTShirts	This example is an online site for selling tee-shirts. This site proposes a home page (with all products), a page for product description (with addition to basket) as well as several other pages (faq, contact form, gsc, etc.).
WW_Rewali	Online travel booking.
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_SchedulingActivities_Club	This example presents a site for a structure proposing activities according to a specific schedule; a sport club for example.
WW_Static_Company	This example is a static site for presenting companies, a “showroom” site.
WW_Static_Crafts	This example is a static site for presenting a crafts workshop or a company working in the crafts field
WW_Static_Restaurant	This example is a static site for presenting a restaurant, a “showroom” site.
WW_TamesShop	This example provides an online store that can be immediately used. The site manages the categories of products, the detailed characteristics, the photos, and it proposes several internal components for managing the customer section, the basket, the payment.
WW_TicketTracker	This example can be used in all areas where it is necessary to follow interventions between several people.
WW_WebDoc	This example is an Electronic Document Management (EDM) WebDoc is using the full-text search to perform searches in the content of documents. The documents are classified by category. This example also explains how to retrieve the content of a PDF document.
WW_WebEstate	This site is Responsive. Its layout fits any browser size automatically.

**Training examples**

WebApp	This example shows how to use planes in WEBDEV.
WW_Auto_Error_Management	This example explains how to use the automatic management of WEBDEV errors.
WW_Auto_Zoom	This example displays an image miniature and when it is clicked, the image is enlarged without disrupting the page layout.

WW_Blog_ResponsiveWebDesign	Simple, responsive blog.
WW_Business_ResponsiveWebDesign	Simple, responsive page.
WW_Chart	This educational example presents different uses of the chart control.
WW_Dashboard	This example presents the use of the Dashboard control.
WW_Drawing_HTML5	Drawing functions in browser code for the browsers that support the HTML5 standard.
WW_Layouts	This example contains a website that can be displayed on mobile devices. The website uses WEBDEV layouts to adapt its UI.
WW_Management_ResponsiveWebDesign	Simple, responsive page.
WW_Mobile_ResponsiveWebDesign	Simple, responsive page.
WW_Organizer	Example for using the Organizer control in WEBDEV.
WW_PayPal_PHP	Explains how to propose a secure payment via the PayPal solution in your WEBDEV PHP sites.
WW_Presentation_ResponsiveWebDesign	Simple, responsive page, portfolio of achievements.
WW_Report	This example shows the different methods for achieving a report: <ul style="list-style-type: none"> <li>• impressions published on different data sources (queries, variables, ...)</li> <li>• prints developed on fields (Table, ...)</li> <li>• printing of composite reports</li> <li>• specific prints (portrait / landscape, state with watermark, state with barcode, ...)</li> </ul>
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_Signature	WW_Signature is an example explaining how to draw in HTML5 and how to save the image in browser code.
WW_Two-factor_Authentication	This example shows how to use two-factor authentication to secure access to user accounts.
WW_Wizard	This example includes a class and a page template that can be re-used to create and manage a Web wizard.



WWBrowserDialog	This example presents a dialog between two distinct browsers.
WW_Websocket_Page	This example shows how to use a WebSocket server to enable communication between multiple clients using a simple static WEBDEV page This example requires you to start WW_WebSocketServer beforehand
WW_WebSocketServer	This example is a WebSocket server for WW_WebSocket_Page.
WW_Wizard	WEBDEV allows you to create your own Web wizards, via the WLanguage PageDisplayDialog function that is used to display a page in a modal way. This allows you to establish a dialog with the user. To go from a wizard plane to another one, all you have to do is use the concept of plane found in WEBDEV.
WWBrowserDialog	This example presents a dialog between two distinct browsers. It is used to open a new page in a new browser and to retrieve the modified values.

## Components

WEBDEV is provided with projects corresponding to components. These projects include:

- a project configuration corresponding to the component.
- a project configuration corresponding to the use of the component.
- an internal component.

Let's see the main components provided with WEBDEV:

WW_CloudTags	Used display a cloud of tags in a dynamic WEBDEV application.
WW_Extraction	Used to extract and store the content of several documents (Open Office, Word 2007, ...) in order for them to be found by the full-text search.
WW_PayBox	Used to install a system for secure payment via PayBox.
WW_PayPal	Used to include PayPal payment buttons in your WEBDEV pages.
WW_SecurePayment	Used to manage a secure online payment. The component manages several solutions for secure payment: Paybox, Ogone E-Commerce, PayPal, Cybermut, SogenActif, E-Transaction, Scellius, Mercanet, Sherlocks, CyberP@iement, SPPlus.

Additional components can be found on our website ([www.windev.com](http://www.windev.com)).