

DEVELOP 10 TIMES FASTER



In which order should these guides be read?

WEBDEV is a powerful tool for developing Web sites that includes all the tools required for developing and maintaining sites.

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

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

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

Note: If a difference exists between the guides and the online help, follow the instructions given in the online help.

We hope you enjoy getting started with WEBDEV.

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

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

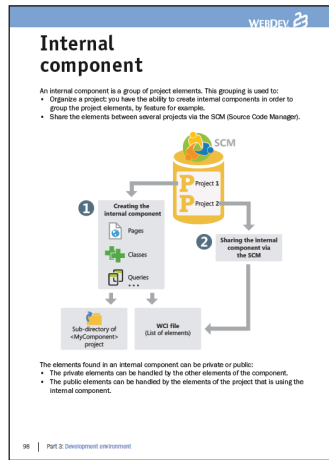
This documentation is not contractually binding. Modifications may have been made to the software since this guide was published. Check the **online help**.

All product names or other trademarks mentioned in this publication are registered trademarks of their respective owners.
© PC SOFT 2018: This publication may not be reproduced in part or in full without the express consent of PC SOFT.

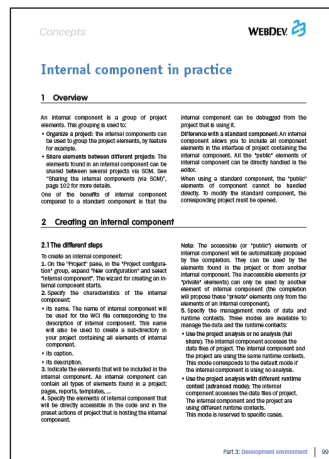
Organization of the guide

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

Summary

PART 1 - INTERNET: MAIN CONCEPTS

- What is a Web site? 11
- Static, semi-dynamic or dynamic site? 12
- The different types of dynamic sites..... 13
- The different types of pages..... 14
- Principle for displaying a dynamic WEBDEV site 18
- Principle for displaying 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
- Development cycle of a site 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
- The different types of standard controls..... 50
- Repeating a group of controls: the loopers..... 60
- The two types of code 61
- WLanguage: a simple and powerful language..... 64
- Sequence for running the code of buttons/links..... 65
- Upload: Sending files to the server 67
- Control templates..... 68
- The reports 69
- Report templates 70
- Different print modes 71
- Displaying and sequencing pages 72
- Performing a process on several successive pages..... 76
- Style sheets: to simplify the layout 77
- Site centered or anchored in width?..... 78
- Referencing a site 79

PART 3 - DEVELOPMENT ENVIRONMENT

The WEBDEV editors.....85
 Project dashboard.....89
 WEBDEV, WINDEV, WINDEV Mobile: 100% compatible90
 Project configuration.....91
 Multiple generation.....92
 Source Code Manager (SCM).....93
 Internal component100
 External component.....103
 Generation modes111
 Project Monitoring Center.....114
 Managing requirements115
 Managing tasks.....116
 Managing business rules117

PART 4 - ADVANCED CONCEPTS

RAD RID121
 The UML model122
 User groupware128
 Multilingual sites134
 10 tips for better interfaces ...138
 Overlaying the controls142
 Type of image: dynamic, static, generated, ... ?143
 Customizing the aspect of a site: skin elements145
 Choosing a type of button/link.....147
 Preventing from going back to a page.....148
 Security benefit: managing the "Back" button149
 Cookies: information stored on the computer of Web user155
 Protecting the access to the site: passwords.....156
 TLS/SSL: Encrypting information on the Web.....157
 Secure payment with provider158
 Sending emails.....159

PART 5 - DATABASES

Analysis: Database structure163
 The different types of accessible files172
 HFSQL Classic174
 HFSQL Client/Server.....175
 HFSQL Client/Server clusters176
 HFSQL: the physically created files.....177
 Associating controls with data178

The queries.....180
 The embedded queries.....181
 The Table/Looper control182
 Retrieving data from a site184
 Universal replication189
 3-tier architecture190

PART 6 - RUNNING THE TEST OF A WEB SITE

Running a site test: The elements to test193
 How to run a site test?194
 The debugger196
 When can I use the debugger?197

PART 7 - DEPLOYING A WEB SITE

Deploying a static site.....207
 Deploying a dynamic WEBDEV site.....209
 On-lining a dynamic WEBDEV site.....213

PART 8 - HOSTING WEBDEV SITES

What is the WEBDEV administrator used for?223
 Server configuration.....229
 Dedicated server or shared server?230
 Which type of server to choose?232
 Hosting Control Center233
 Dynamic site on specific configurations234
 Traffic statistics of dynamic sites.....237
 Monitor your sites, servers,238

PART 9 - APPENDICES

WLanguage functions specific to WEBDEV 23241
 Examples and components supplied with WEBDEV.....250

PART 1

Internet: Main concepts

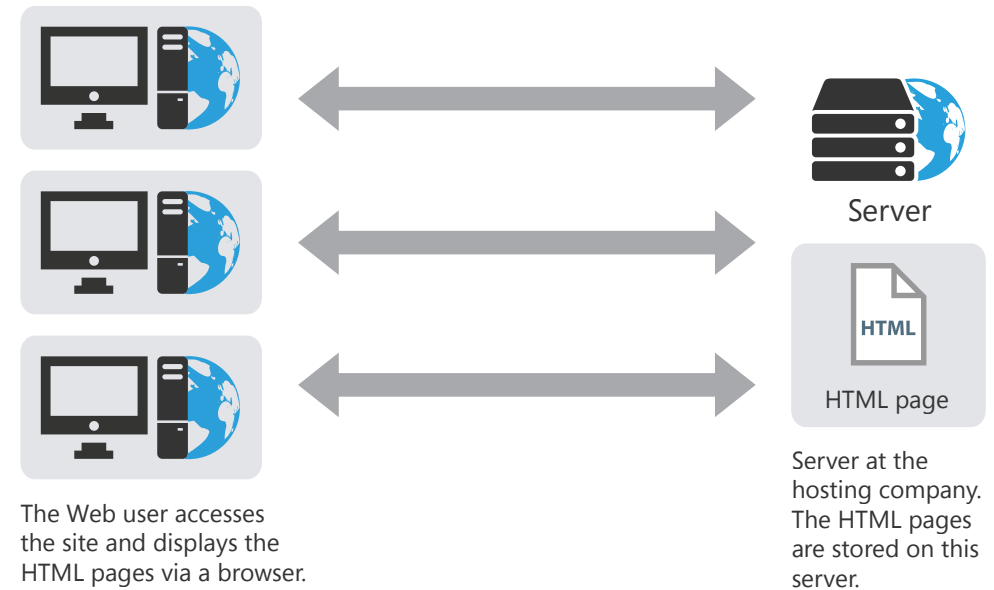


What is a Web site?

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

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

WEBDEV allows you to easily create Web sites that manage (or not) the data.



Static, semi-dynamic or dynamic site?

Several types of sites can be created:

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

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

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

Note: A dynamic site can also include a static section (the presentation of the company, ...).

WEBDEV allows you to create:

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

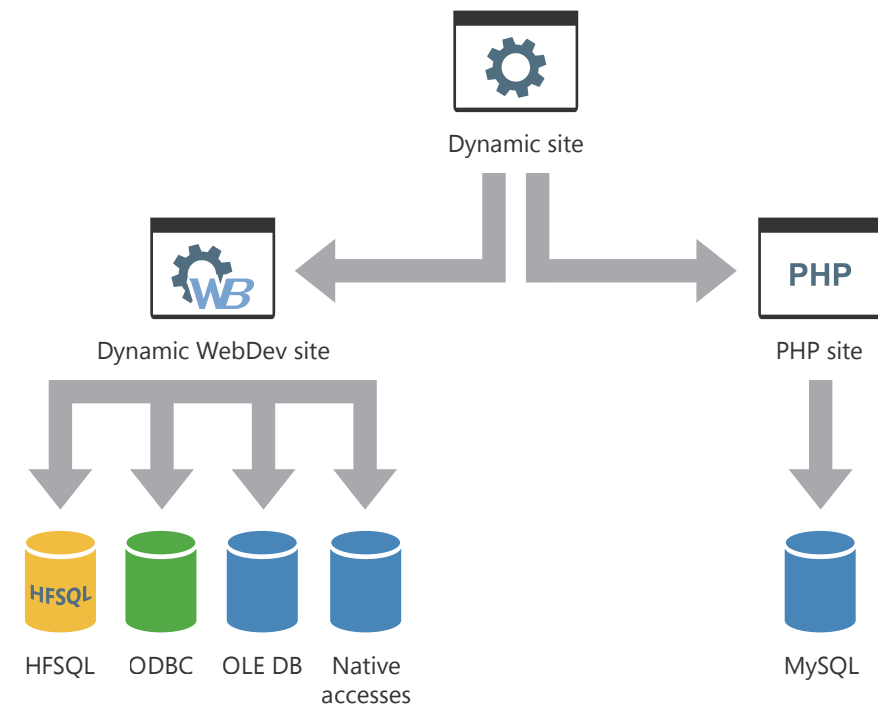
The different types of dynamic sites

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

A dynamic site can be used to manage real-time travel 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.** 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.



- Oracle
- SQL Server
- MySQL
- DB2
- XML
- Informix
- Progress
- Sybase
- Oracle Lite
- AS/400...

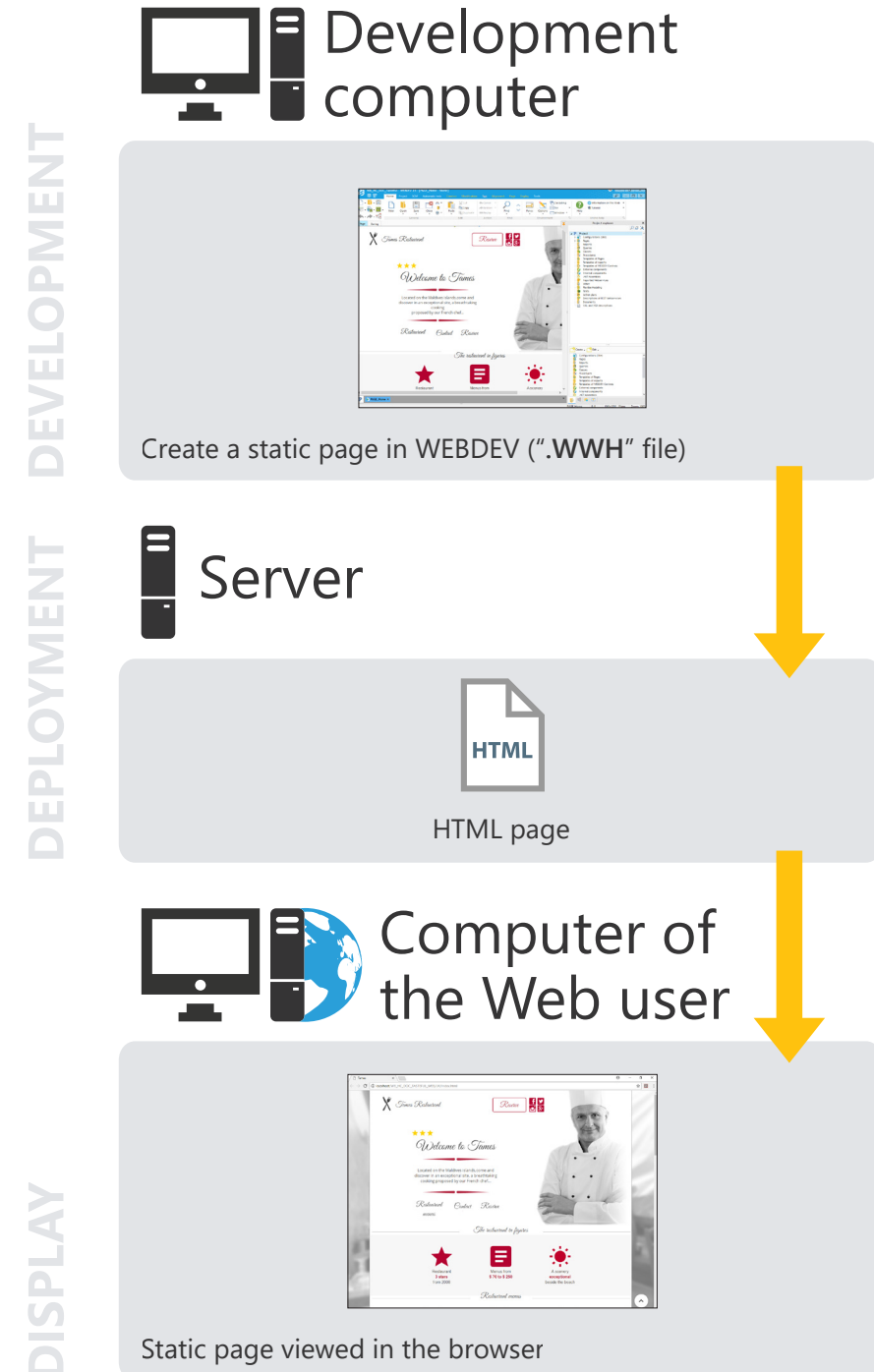
The different types of pages

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

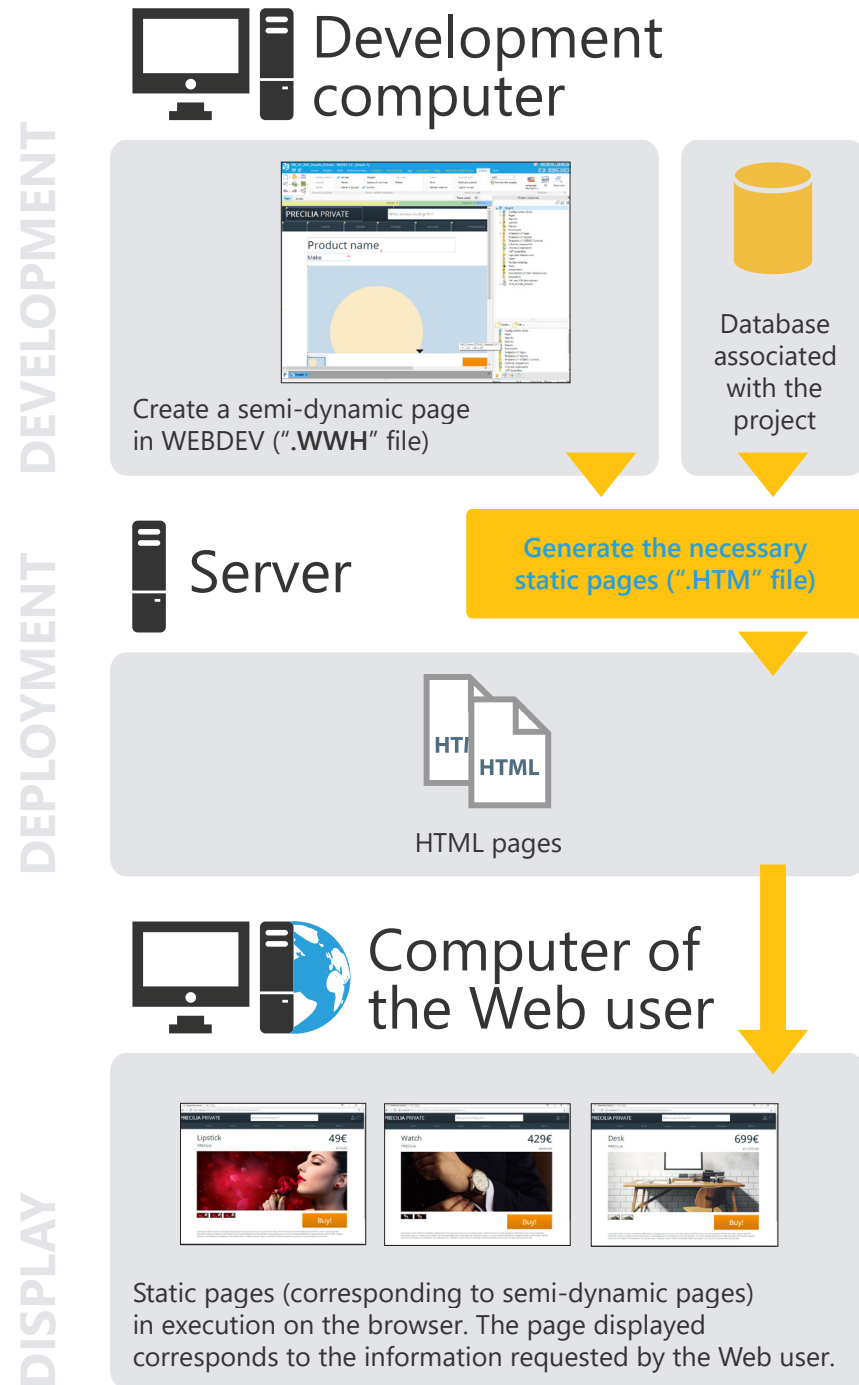
- **The static pages.** This type of page is used to display fixed data.
- **The semi-dynamic pages.** These pages are used to display the data found in a database. These pages and their content are generated on the development computer when creating the site. The page content is defined when the HTML pages corresponding to the dynamic pages are created.
- **The dynamic WEBDEV pages.** These pages are used to dynamically display the information found in a database. The page content depends on the displayed record.
When displaying a dynamic page, an 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.
- **The dynamic WEBDEV AWP pages (Active WEBDEV Page).** These pages are used to dynamically display the data found in a database. The page content depends on the displayed record.
An AWP page is a dynamic WEBDEV page **without persistent context** on the server. The AWP page context is temporary. It is created in a temporary session.
- **The dynamic PHP pages.** These pages are used to dynamically display the data found in a database. These pages can only be used in a PHP site.

The following pages present these different types of pages.

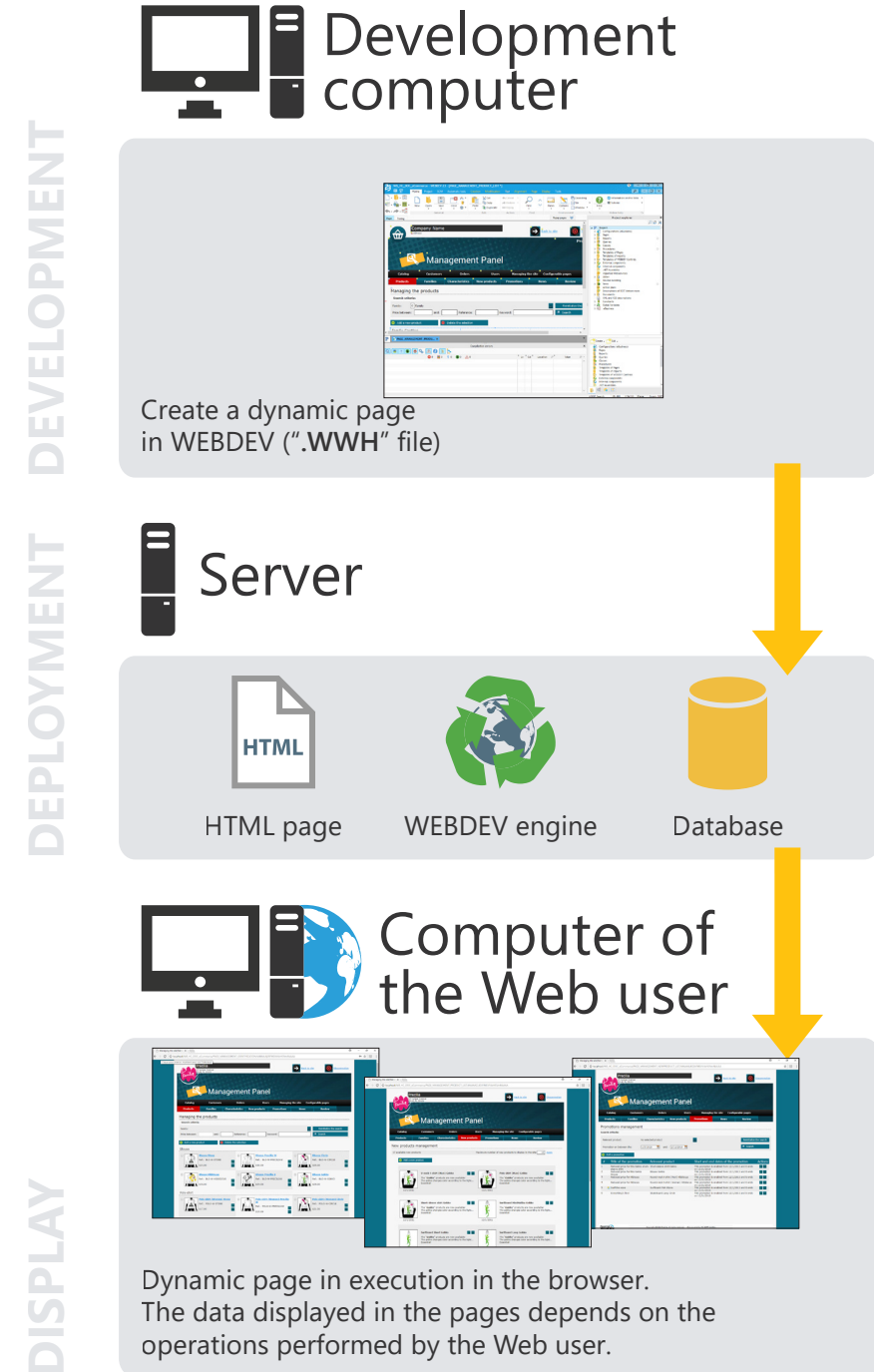
Principle of a static page



Principle of a semi-dynamic page



Principle of a dynamic page



Principle for displaying a dynamic WEBDEV site

The WEBDEV session

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

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

The page contexts

A page context is automatically created on the server whenever a page displayed in the browser. This page context contains all the elements that have been required to build the page viewed by the Web user:

- the global variables,
- the local variables,
- the server processes,
- the connections to the databases,
- the file contexts, ...

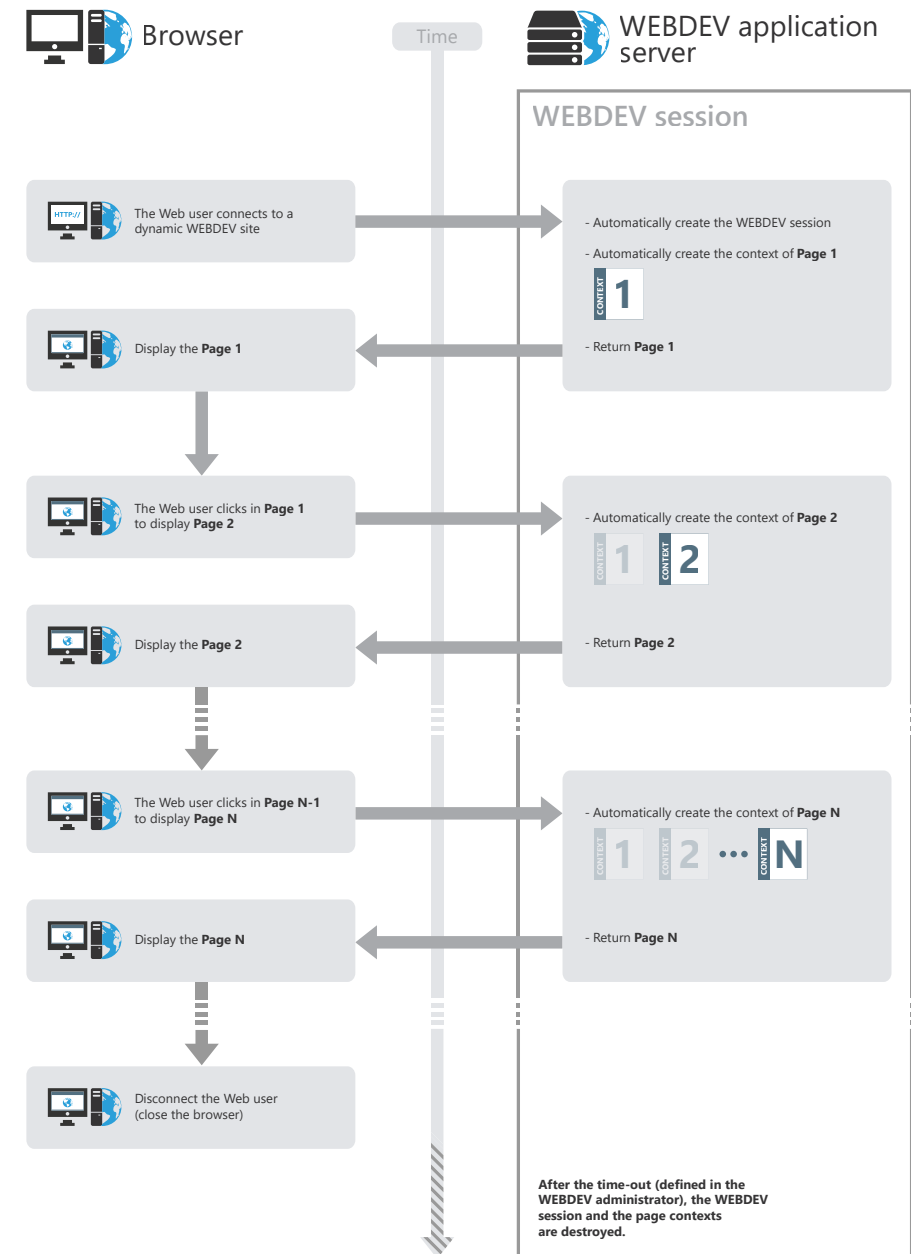
The page contexts remain in memory on the server until the end of WEBDEV session.

If the same page is called several times:

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

Programming

The WEBDEV session and the page contexts are automatically managed by default. There is nothing to program.



Principle for displaying an AWP site

What is an AWP page?

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

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

Operating mode of AWP sites

Are automatically created on the server whenever an AWP page is displayed in the browser:

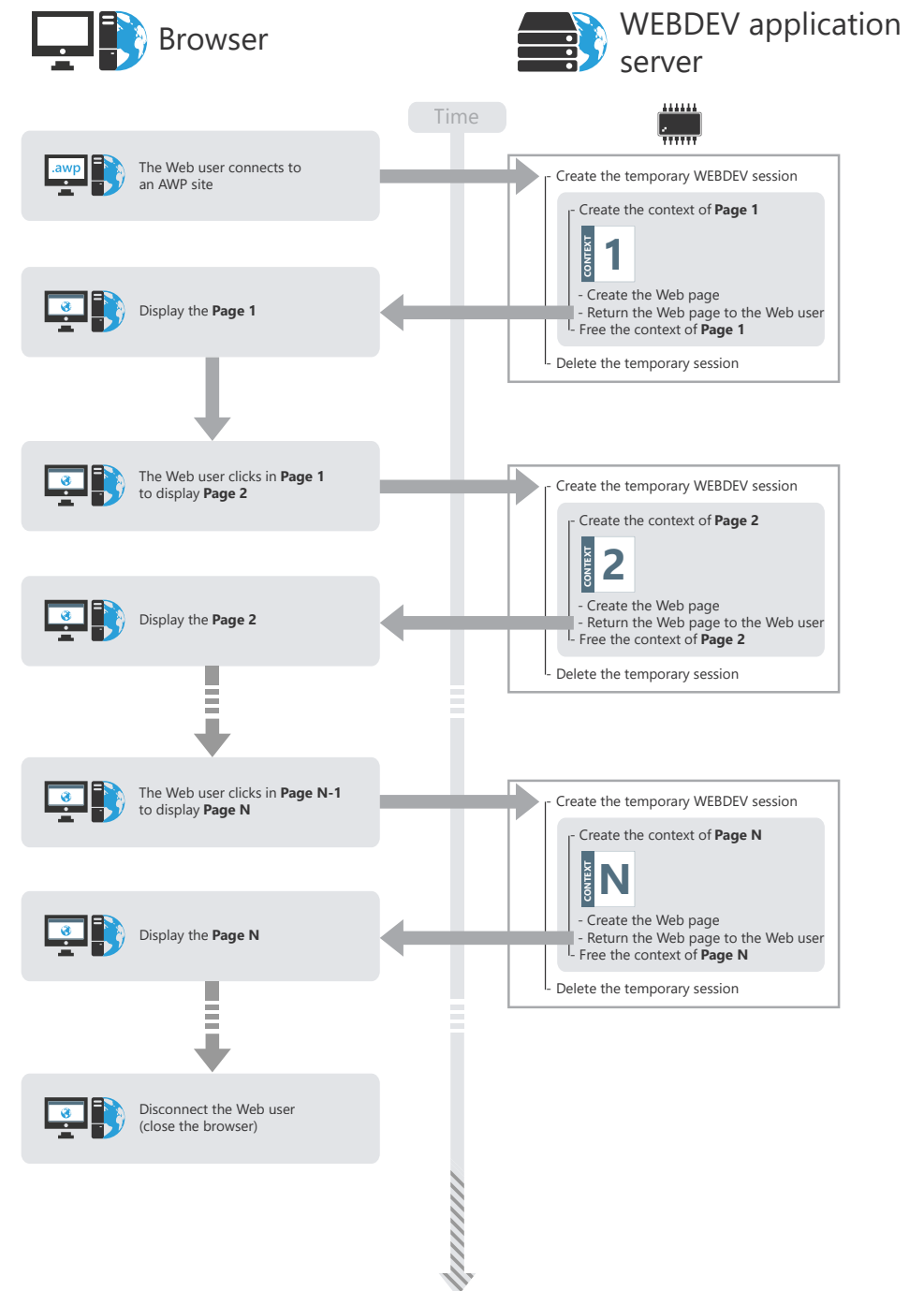
- a temporary session,
- a temporary AWP page context.

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

- the local variables,
- the server processes,
- the connections to the databases,
- the file contexts, ...

When the AWP page was sent to the Web user, these elements are destroyed.



How to share information (values) between AWP pages?

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

- Passing information in the URL. This method allows for a better referencing.
- Saving information in the AWP contexts (by programming).

Passing information (values) between two pages in the URL

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

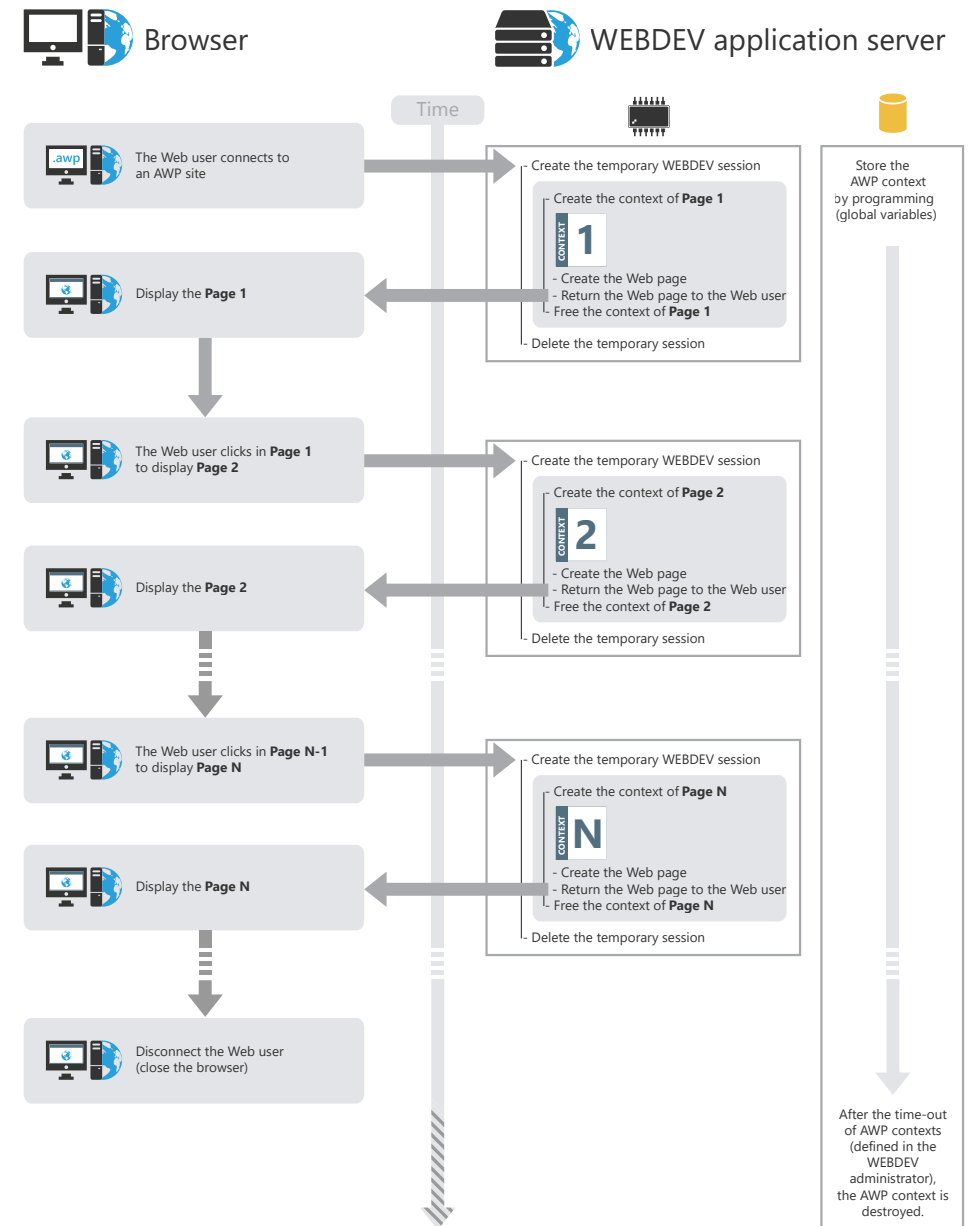
This method allows for a better referencing of the page because the information passed in the URL is visible and analyzed by the referencing robots.

Saving information (values) in the AWP contexts (by programming).

You have the ability to store on the server values common to several AWP pages, via the **AWP contexts**. An AWP context is created on disk on the server. This context is available as long as AWP pages are displayed and as long as the time-out of AWP contexts is not exceeded. The time-out of AWP contexts is defined in the WEBDEV administrator ("Configuration" tab, "Time-out of AWP contexts" option).

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

See the online help for more details.



AJAX technology

The AJAX technology is available in native mode in WEBDEV.

What is AJAX and what are its benefits?

AJAX (Asynchronous Javascript and XML) is used to refresh the data modified in an HTML page without having to redisplay the entire page. For example, if some elements found in the page displayed (content of the basket, characteristics of a product, list of cities, map, ...) are modified, only these elements will be refreshed. The server does not have to send the entire page onto the computer of the Web user.

This technology presents several benefits:

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

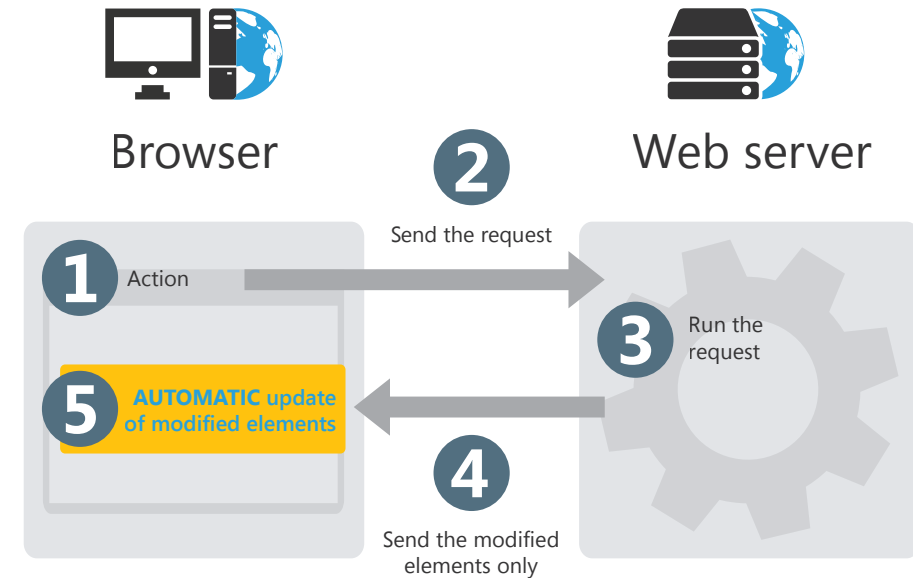
AJAX can be used at two different levels in a WEBDEV site:

- Automatic and immediate AJAX: a simple click allows you to access the AJAX features. The code remains the same.
- Programmed AJAX: functions for AJAX management allow you to write complex processes.

Note: Only the recent browsers support the AJAX technology. **AJAXAvailable** is used to find out whether the current browser supports the 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 and immediate AJAX

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

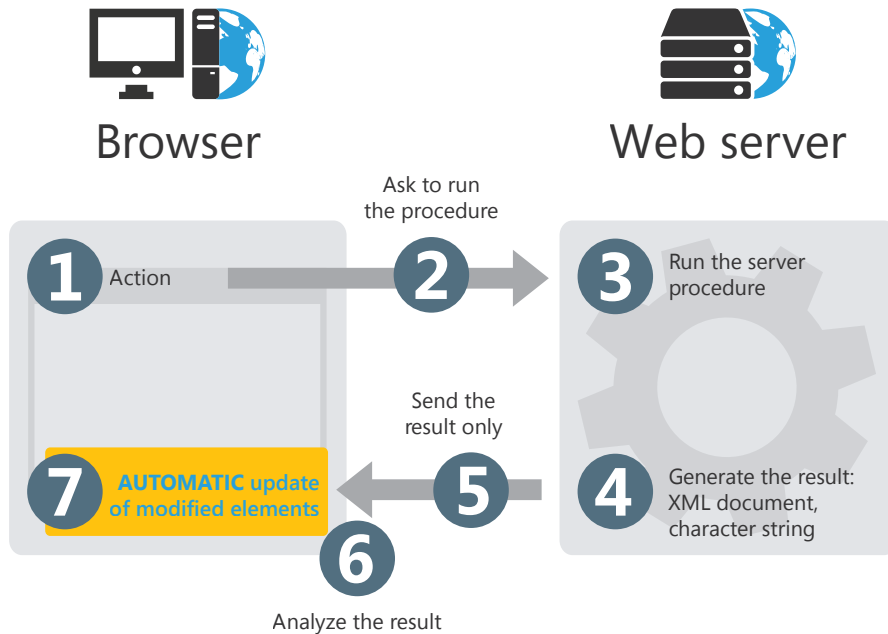


For example, a site page is used to find out the different characteristics of a country (capital city, currency, flag, location, ...). The corresponding information is displayed according to the country selected by the Web user.

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

Programmed AJAX

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



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

If you are already familiar with WINDEV

The main differences between WEBDEV and WINDEV are as follows:

- WEBDEV is used to create pages while WINDEV is used to create windows.
- In WEBDEV, different types of code can be typed: a WLanguage code run on the server, a WLanguage code run on the browser and a Javascript code run on the browser. a PHP code run on the server.
- New WLanguage functions specific to the Web are available.
- Some WLanguage functions not relevant for the Web do not exist anymore.
- Some WLanguage functions can only be used in an executable process in server code.
- Some WLanguage functions can only be used in an executable process in browser code.
- New types of controls specifically designed for Web applications are available:
 - Formatted display control
 - Java Applet control
 - Scrolling Banner control
 - Navigation Bar control
 - Captcha control
 - Cell control
 - Site Map Path control
 - Web Component control
 - Flash control, Flex control
 - Clickable Image control (Map Area)
 - IFrame control
 - Link control
 - Horizontal Rule 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
 - Text Area control
- Some types of controls without purpose on Internet are not available anymore: scrollbar, ActiveX, OLE object, spin, ...

Hardware and software requirements

The Web user

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

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

The server

- **1 server:** PC
- **1 server operating system** (Windows or Linux): **Windows** 2008 or later.
- **1 Web server software:** IIS, Apache, ...
- **1 WEBDEV Application Server**
or
• **1 PHP engine** (version 4.3.2 or later)

The development computer

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

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

PART 2

Developing a site



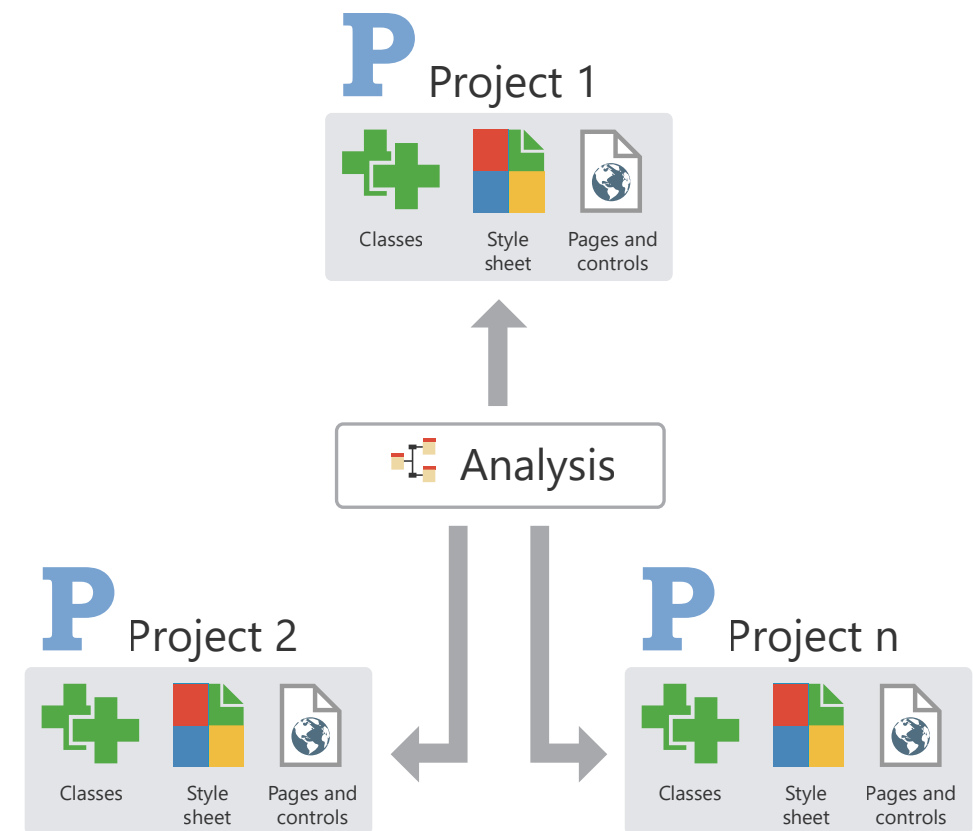
Project and Analysis

Developing a **Web site** 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, ... whose combination allows you to build a Web site.

A **WEBDEV analysis** groups the description of data files found in the site.

A site is built from a project.
 In most cases, a project is associated with an analysis.
 An analysis can be associated with one or more projects.



The project in practice


1 Overview

When describing a site, the first step consists in creating a project.

When creating a project, several questions will be asked in order for your project to match your expectations. All the characteristics specified when creating the project can be modified later.

2 Creating the project

To create a project:

1. Click  among the quick access buttons of WEBDEV ribbon. The window for creating a new element is displayed: click "Project". The wizard for project creation starts.
2. Specify the different project options. The main options are as follows:
 - **the name and location:** these options cannot be modified. The project corresponds to a ".WWP" file. All the objects associated with the project will be created in the specified directory.
 - the **type of project generation:** which means the type of site that will be generated by the project: static site, dynamic WEBDEV site, semi-dynamic site, dynamic PHP site, ...
 - the **creation mode of project:** you have the ability to create a blank project or a project based on the Application RAD.
 - **whether the project will be handled by several developers:** you have the ability to use the developer groupware or the source code manager

(SCM) to share the project.

- **the programming charter:** this charter is used to automatically prefix the variables, the names of controls, the names of pages, ...
 - **the style book.**
 - **the supported languages:** these languages will be proposed by default as soon as an option of a control, page, report, ... can be translated.
3. Specify **whether the project is associated with a database or not.** If yes, it can be an existing database or a new one.
 4. Validate the wizard. The created project becomes the current project.

Notes:

- If you have requested the creation of a database, the wizard for analysis creation is automatically started.
- The specified information can be modified in the project description. To display the description window of the project, on the "Project" pane, in the "Project" group, click "Description".

3 Project dashboard and project graph

3.1 Dashboard

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

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

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

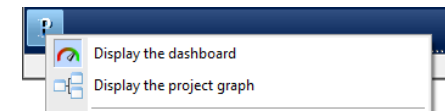
The developer wants to quickly start the most often used project elements, code sections, ...

The dashboard gives an answer to all these wishes. The dashboard proposes different indicators called "Widgets". These Widgets can be configured by the user: you have the ability to add or delete some, and to resize them.

You can for example display the project statistics (number of pages, ...), the result of different audits (static audit, dynamic audit, ...), ...

To display the dashboard of your project:

1. Display the popup menu of project in the bar of opened documents.
2. Select "Display the dashboard".



See "Project dashboard", page 89 for more details.

3.2 Project graph


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

To display the graph of your project:

1. Display the popup menu of project in the bar of opened documents.
2. Select "Display the project graph".

3.3 The project elements

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

To find out the list of project elements, on the "Project" pane, in the "Project" group, click .

This option is used to:

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

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

4 Operations performed on a project

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

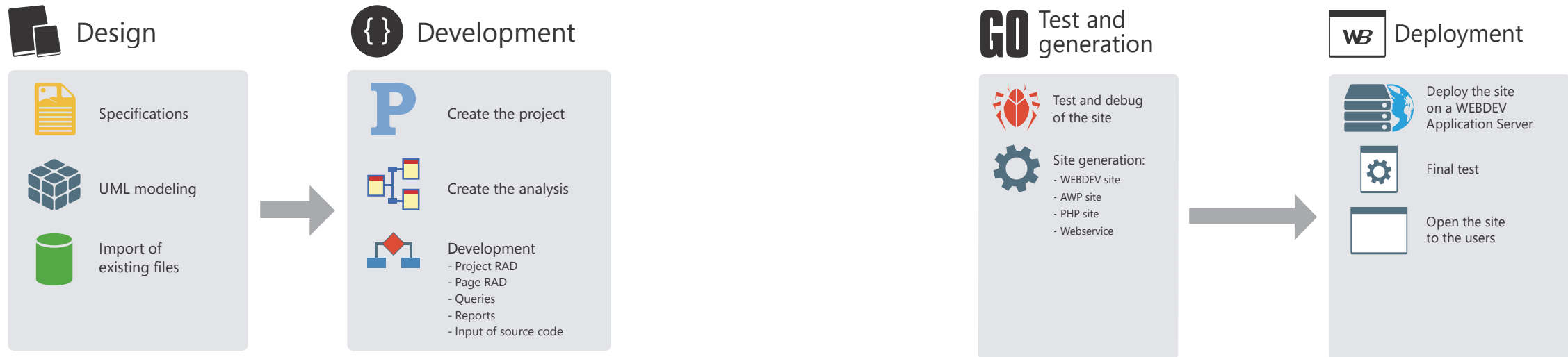
- Archive a project,
- Restore a project,
- Duplicate a project,

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

See the online help for more details.

Development cycle of a site

WEBDEV covers the entire development cycle of a site:



Details of different steps:

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

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

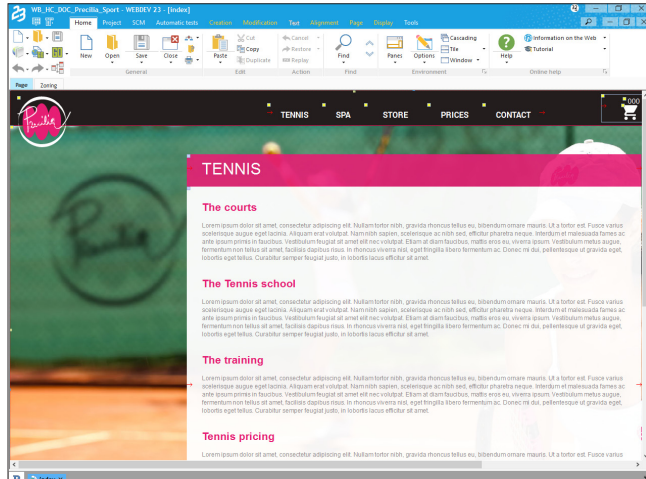
Test and generation step: WEBDEV offers several automated test tools to guarantee the reliability of applications and make sure that no regression occurs during the development step.

Deployment step: Deploying a dynamic WEBDEV site is performed on a WEBDEV Application Server. Once the site is deployed, you have the ability to run the very last tests before on-lining the site.

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

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

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



- 2 Typing code (optional step).

```

// Identification - Initialization of LINK_ForgottenPassword ( CELL_AccountCreat...
// Click (onclick) of LINK_ForgottenPassword ( CELL_AccountCreat...
// Check whether a login is entered
IF EDIT_Existing_EmailAddress -- ** THEN
  ErrorExistingAccount("Enter your email")
  ReturnToCapture(EDIT_Existing_EmailAddress)
END

// Check whether the specified login exists
IF Val(AJAXExecute(bCheckConnection, EDIT_Existing_EmailAddress, ""))
  ErrorExistingAccount("Identifier not found")
  ReturnToCapture(EDIT_Existing_EmailAddress)
END
    
```

Browser code

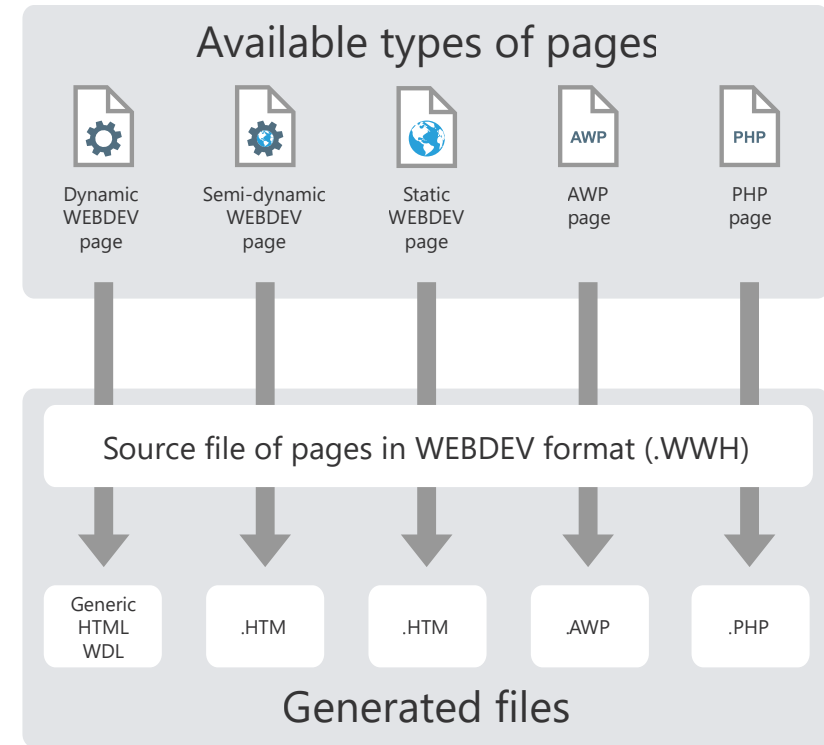
```

// Identification - Click of BTN_ConnectYourself ( CELL_AccountCreation ) (server)
// Read the Customer record
HReadSeekFirst(Customer, Email, EDIT_Existing_EmailAddress)
IF HFound(Customer) THEN
  // Client connection
  Connection(Customer.CustomerID, Customer.Email)
  // Displays the next page (on the "customer_account" page by default)
  PageDisplay($GetAccessPathNextPage("customer_account"))
ELSE
  // Error message
  DisplayError("Unable to find the customer")
END
    
```

Server code

- 3 Saving 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 it remains on the development computer.

.WDL: Project library

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

.Generic HTM

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

.AWP: AWP page

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

.HTM: HTML page of site

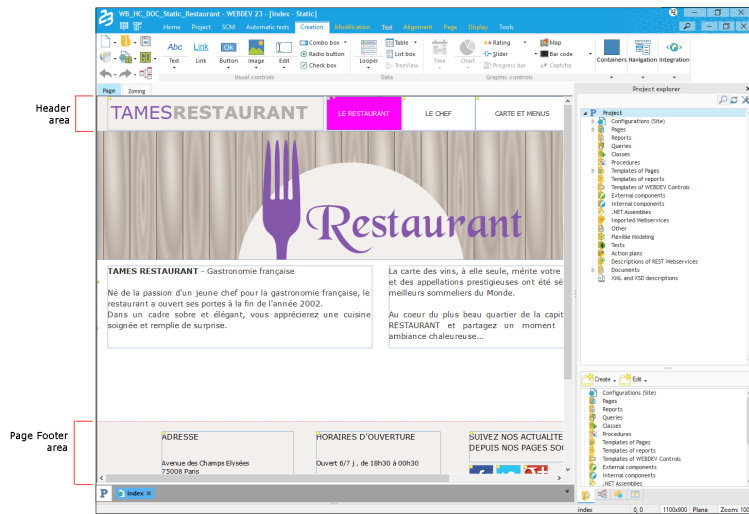
This file is generated from the WWH file and it contains the description of controls in HTML and the browser code (if the code was typed in WLanguage, it is automatically translated into Javascript). This file will be found on the server.

.PHP: PHP page

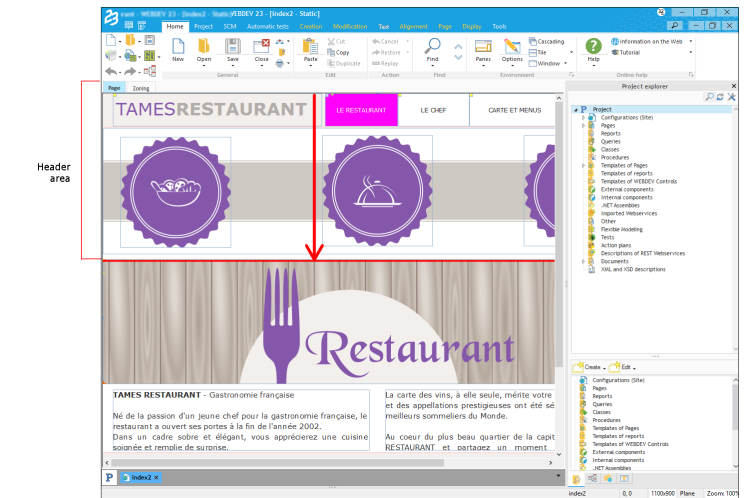
This file is generated from the WWH file and it contains the description of controls, the server code, the browser code in PHP format. This file will be found on the server.

Editing a page: zoning mode

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



During the print, the relative position of controls found in each area is respected as well as their anchor.



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

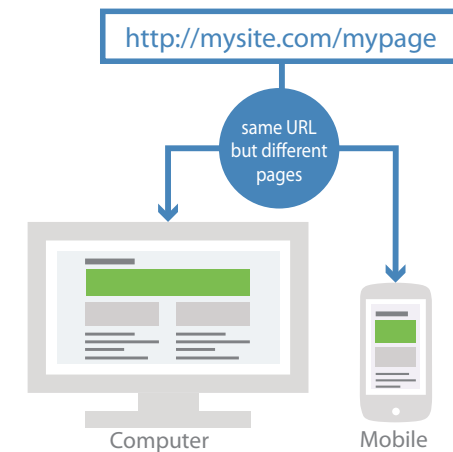
"Mobile friendly" site: Dynamic serving

"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 proper page will be displayed according to the device (PC or mobile) that accesses the page.

This is an alternate solution (even complementary) to "Responsive Web Design" to make a standard site "mobile friendly". With "Dynamic serving", no need to modify the existing pages: all you have to do is add some for the mobile devices.

This technique benefits from a great referencing by Google.



Principle

The site includes 2 sets of pages: the pages for computer and the pages for mobile device. These pages are associated 2 by 2.

When the address of a page 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 the pages opened in the code are automatically redirected to the corresponding page in the current platform.

Notes:

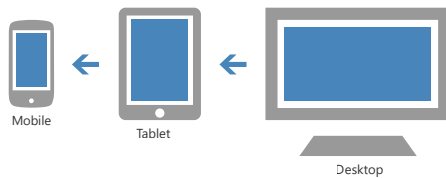
- The detection of the environment is performed according to the Google recommendations: the tablets are not considered as being mobile devices.
- The pages are identified by Google as being in "Dynamic serving" mode. Therefore, the different page versions are referenced. If the mobile pages are developed properly, the site is considered as being "Mobile friendly".

"Mobile friendly" site: Responsive Web Design

The Web sites can be viewed on several platforms: mobiles, tablets, PCs, ... The browser size changes on each device as well as the space used to display data. The "Responsive Web Design" method consists in creating a single page that automatically adapts according to the platform used.

Method used

WEBDEV is using the "Desktop first" method: the developer must create the interface of his Web site in "Desktop" mode. Then, he will create the version in "Tablet" mode and finally he will create the version in "Mobile" mode.

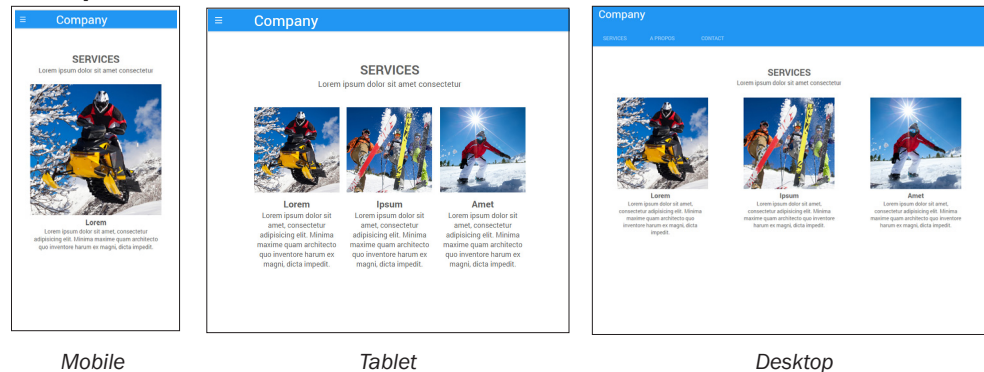


During the design step, it means that a Web site is developed for a "mobile" use first. When reducing the display resolutions (page size), the content and the features are moved, resized or hidden.

In the WEBDEV editor, the representation of different resolutions is performed via blades. A blade represents a browser size for a specific device. WEBDEV includes 3 blades:

- The Mobile blade corresponding to smartphone devices.
- The Tablet blade corresponding to tablet devices.
- The Desktop blade corresponding to devices such as PC, MAC, etc.

Example

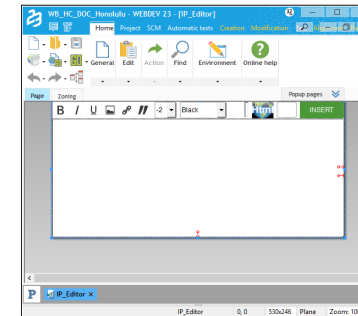


Internal page

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

1. Creating an internal page

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

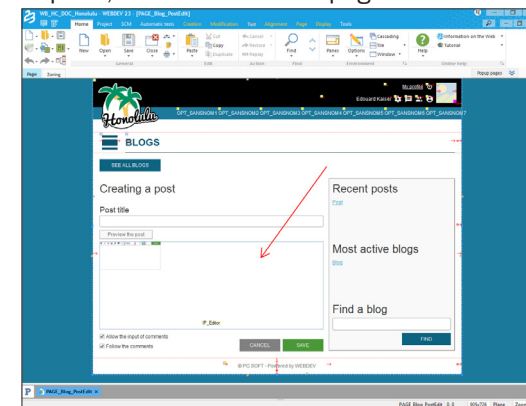


In the example, an internal page is used to manage the basket.

2. Using an internal page

To use an internal page, you must:

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

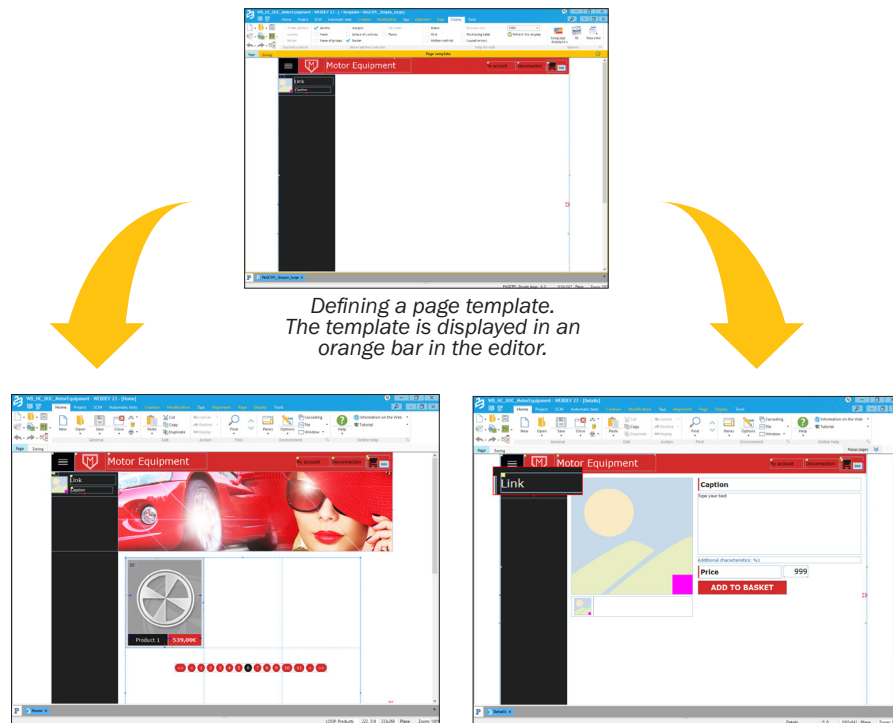


Notes:

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

Page templates

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



Defining a page template. The template is displayed in an orange bar in the editor.

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

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

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

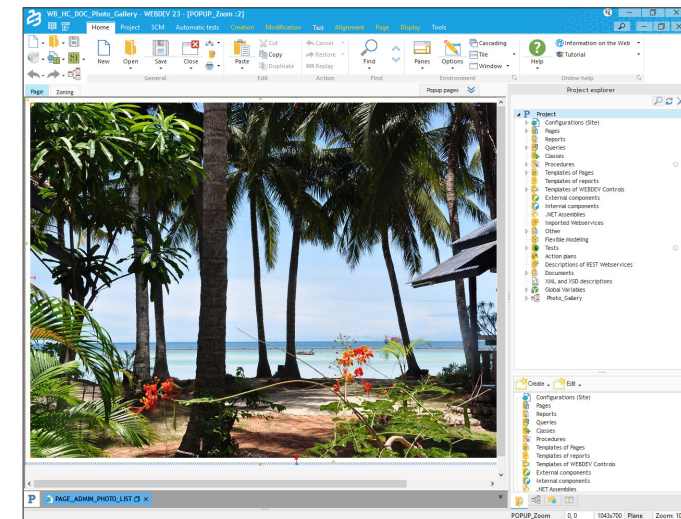
The characteristics of elements can be dissociated from the template. For example, dissociating the position of a template control to position the control somewhere else while keeping the other evolutions performed on the control (code, style, ...). We talk of **inheritance**. In this case, the elements are identified by a blue square.

Popup page

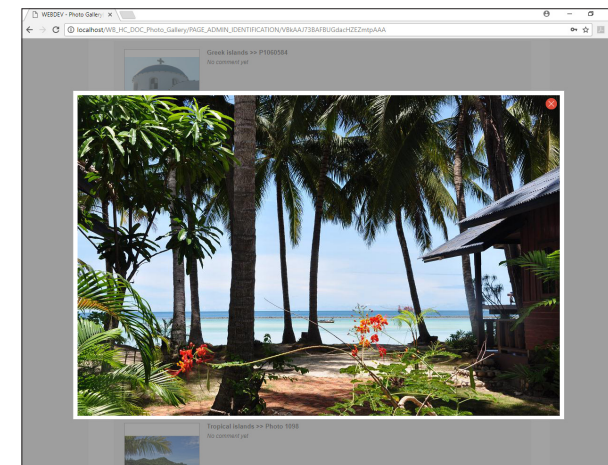
WEBDEV allows you to create Popup pages. The Popup pages are used to communicate with the user in a simplified way.

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 run time:



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, ...

WEBDEV proposes several solutions 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, ...

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

1 Creating a page


WEBDEV proposes several methods for creating a page:

- Creating blank pages, without control.
- Creating preset pages.
- Creating pages via the project RAD.

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

1.1 Creating blank pages

To create a blank page:

1. Click  among the quick access buttons of WEBDEV. The window for creating a new element is displayed: click "Page" then "Page". The wizard for page creation starts.
2. Choose the "Blank page" type.
3. Select the type of blank page to create:
 - Simple layout: Blank page using the zoning.
 - Responsive Web Design: Blank page whose type is Responsive Web Design.
 - HTML stream: Blank page whose type is HTML Stream.
4. Validate the creation of the page. A blank page is created.
5. The backup window is automatically displayed.
6. Specify the page title and its 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.

See "The controls in practice", page 54 for more details.

This chapter presents the following topics:

- "Creating a page", page 44,
- "Simple operations performed on a page", page 45,
- "Main characteristics of a page", page 45,
- "Internal page", page 46,
- "Page templates", page 46,
- "Popup page", page 43,
- "Refreshing a dynamic page", page 47,
- "Opening a page in a WEBDEV site", page 48,
- "Processes associated with pages", page 48,
- "The menus", page 49.

7. Validate the backup window.


8. Create the controls in the page.

Notes:

- The created page is a dynamic page by default. The type of page can be modified in the description window of the page. See "Main characteristics of a page", page 45 for more details.
- The different types of controls are presented in "The different types of standard controls", page 50.

1.2 Creating preset pages

To create a preset page:

1. Click  among the quick access buttons of WEBDEV. The window for creating a new element is displayed: click "Page" then "Page". The wizard for page creation starts.
2. Choose the type of page to create in the list found on the left of window:
 - the "Pages" category is used to create:
 - a blank page containing no control.
 - a blank page based on a page template, if the project contains a page template.
 - an error page. This error page is used to customize the error page displayed in the site.
 - an internal page.

- the "RAD" category is used to create:
 - dynamic RAD pages used to display and type data. This type of page can be based on a data file, on a query, ...
 - semi-dynamic RAD pages used to display data. This type of page can be based on a data file, on a query, ...
 - RID pages used to enter data. Only the interface of the page will be generated. The programming must be done by the developer.
 - the "Preset templates" category is used to create:
 - pages based on preset templates (2 or 3 columns, full, inset on right or left, ...).
 - Responsive Web Design pages based on preset templates.
 - pages specific to the creation of Vista Gadgets.
 - pages specific to the creation of Web sites in Mobile.
3. Depending on the selected type of page, type the requested information in the different wizard steps.
 4. Validate the creation of the page.

5. The backup window is automatically displayed. Specify the page title and its name, then validate.

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

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

See the online help for more details.

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

1. On the "Project" pane, in the "Generation" group, click "Full Application RAD".
- Caution** To use this option, the project must be associated with an analysis that was generated at least one.
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 by RAD.

2 Simple operations performed on a page

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

- Open a page in the editor: All you have to do is double-click the page name displayed in the "Project explorer" pane.

- Save and copy a page.
- Import and export a page.
- Modify the navigation order.

See the online help for more details.

3 Main characteristics of a page

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

- include a **background image**: this image is displayed in page background and the controls are superimposed on this image.
- include a **menu**: this menu allows the Web users to quickly access the main site features. To create a drop-down menu in a page, all you have to do is use a "Menu" control. The first two options are displayed in the page. The options can be handled via the popup menu of options. Then, the menu options can be handled by programming. See "Handling the menu options in the editor", page 49 for more details.
- include a **status bar**: this status bar will display the help message associated with the different

page controls. See the online help for more details.

- 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 (pages shared between several projects for example). See "Multilingual sites", page 134 for more details.

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

4 Internal page

The internal pages are used to dynamically share a section of interface inside one or more sites.

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

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

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

See the online help for more details.

5 Page templates

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

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

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

Using page templates in your sites allows you to:

- simplify the creation of site pages.
- simplify the layout of site pages.

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

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

By default, any modification performed in the template is applied to the pages that is using it. However, special cases can be managed in a specific page by overloading the template elements. See the online help for more details.

6 Popup page

WEBDEV allows you to create popup pages. These popup pages allow you to communicate with the Web user. A Popup page can be used to:

- request the input of information (SMTP parameters for sending emails, new address, ...),

- ask to confirm a deletion,
- ...

To open a popup page from another page, you have the ability to use the browser function named **PopupDisplay**.

See the online help for more details.

7 Refreshing a dynamic page

7.1 Overview

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

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

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

7.2 Implementation

To refresh a page, all you have to do is use **PageRefresh**.

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

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

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

8 The page contexts

8.1 Overview

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

In a dynamic WEBDEV site, a page context exists on the server for each page displayed on the browser of the Web user.

The page context groups the information relative to the page:

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

8.2 Automatic operating mode

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

- A page context is opened when the dynamic page is displayed in the browser.
- The context of a page is updated according to the information typed by the Web user in the browser. This update is performed during the page validation (via a "Submit" button or via **PageSubmit**).
- The existing page contexts are closed when using **PageUse**. Then, the page context corresponding to the page that will be displayed by this function is opened.

Note: WEBDEV allows you to perform an advanced management of contexts via the ContextXXX functions. See the online help for more details.

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, ...)
- by programming in WLanguage.

9.2 Opening a page from a control description

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

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

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

9.3 Opening a page by programming

Several WLanguage functions can be used to open a page.

- **PageRefresh**
Refreshes the page displayed in relation to its context.

10 Processes associated with pages

10.1 Processes managed by default

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

- **Global declarations (server code):**
Declaration of variables global to the page. This is the first code run when the page is opened.
- **Initialization (server code):**
Run when opening the page. The initialization process of page controls is run before this code.

- **PageDisplay**
Opens and displays a new page in the browser of the Web user.
- **PageDisplayDialog**
Opens and displays a new page in modal mode in the browser of the Web user.
- **PageUse**
Closes all the current pages (and their contexts) and opens a new page.
- **PopupDisplay**
Displays a popup in the page.

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

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

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

9.4 Case of dynamic pages

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

1. Checking the existence of page context on the server.
2. The context is closed if it exists.
3. Opening the page context. All the variables, controls, ... linked to the page are re-initialized.
4. Displaying the requested page in the browser.

- **Loading (onload) the page (browser code):**
Browser code run when the page is displayed in the browser.
- **Unload (onunload) of the page (browser code):**
Browser code run when a new page is displayed in the browser.
- **Synchronizing the page (server code):**
Server code used to manage the page synchronization when using the browser "Back" key.
- **Closing the page (server code):**
Run when closing the page.

- **Assigning the ..Value property (internal page only)**

Run when using the ..Value property in assignment on the Internal Page control.

- **Retrieving the ..Value property (internal page only)**

Run when using the ..Value property in read-only on the Internal Page control.

10.2 Optional processes

Several optional processes can be managed.

To manage an optional process, you must:

1. Display the code window of the page.
2. Click "..." found in the "Other processes" process. The window of optional processes is displayed.
3. Select the optional process to add and validate.
4. The selected process is automatically added after the processes proposed by default.

You have the ability to manage:

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

11 The 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,
- run the code of buttons found in the page,
- ...

WEBDEV allows you to:

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

See the online help for more details.

12 The menu options

12.1 Overview

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

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

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

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

12.2 Handling the menu options in the editor

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

- Add a menu option,
- Add a separator,

- Add a sub-menu,
- Delete a menu option.

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



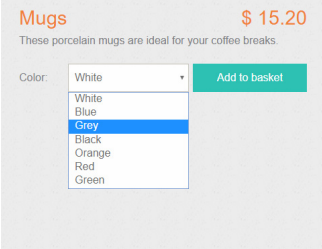
The description window of menu options allows you to:


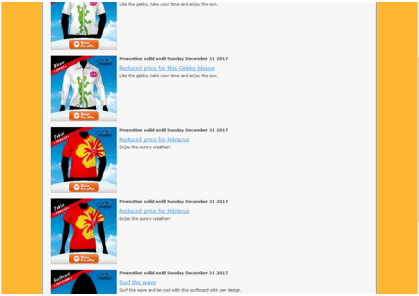
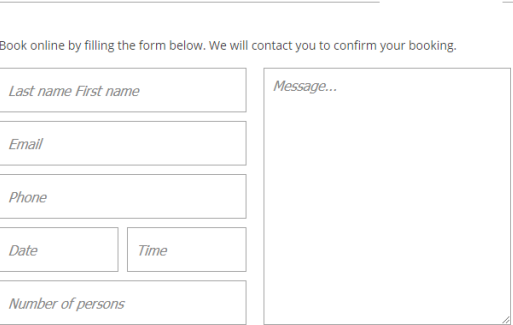
- modify the caption of option,
- check or uncheck a menu option,
- associate an image with a menu option,
- associate a keyboard shortcut with a menu option.

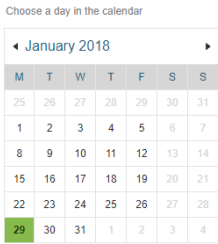
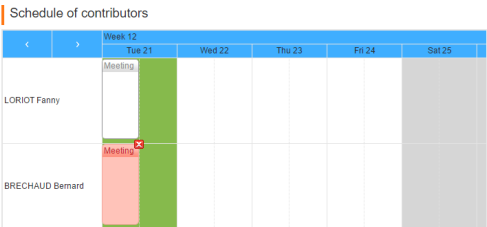
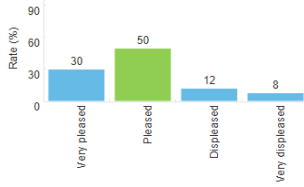
You also have the ability to associate a WLanguage code with a menu option. Only the menu options without sub-menu can start a WLanguage process.



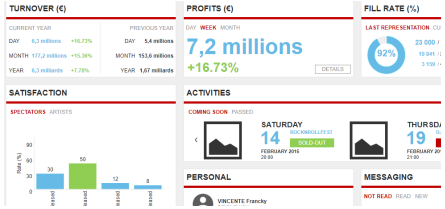
See the online help for more details.

The different types of standard controls

You want to...	Use a control...
Display a text, a title, ... 	Static
Propose a caption with a specific layout (images, links, ...) 	Rich Text Area control
Define a color area	Cell, HTML table
Display a price, a quantity, an address, a date, a time, ...	Formatted display control
Select a value from a list (country, city, color, ...) 	Radio button, Combo box, List box
Select several values from a list (recipients of a message, files to download, ...)	Check box, List box
Display an animation (animated Gif)	Image
Display a page according to the area that was clicked in an image	Clickable image
Display a Flash animation	Flash control

Use an existing HTML code	HTML control
Display a graphic image (photo, ...) 	Image, Clickable image
Use a Java applet (clock, ...)	Java control
Display the file content in a table (list of customers, order details, ...)	Memory table or browsing table, Memory looper or browsing looper
Display an image refreshed on a regular basis	Web camera
Repeat controls in a page (product catalog with photo, ...) 	Looper
Type information 	Edit control
Display information via a hierarchical order (directory content, ...)	TreeView control
Align controls (HTML table)	Cell, HTML table

Program an action in a page (display a page, validate an input, ...)	Button, Link control
Display a video	Video control
Display a page from another site in one of your pages	iFrame
Select and display a date on a calendar	Calendar control
	
Display the appointments in scheduler or organizer format	Scheduler control, Organizer control
	
Display a column chart, a line chart, a pie chart	Chart control
	
Display a page from your site inside one of your pages	Internal page
Display an automatic menu (that is built while the site is browsed).	Site map path

Allow the Web user to give or view a rate.	Rating control
	
Display thumbnails	Thumbnail control
Display a set of images in gallery format	Gallery Looper control
Display Flex files	Flex control
Include SilverLight controls	SilverLight control
Expand/Collapse a display area	HideShow control
Upload one or more files	Upload control
	
Display several contents successively	Scrolling Banner control
Display the dashboard displaying various information in thumbnail format	Dashboard control
	
Display a list of objects as a scrolling horizontal list.	Linear Looper control
Use external resources available on Internet (Angular JS components, Bootstrap snippets, ...)	Web Component control
Create a visual effect in a page by presenting an image partially covered by another one.	Peeling Corner control

The 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" pane of

the WEBDEV ribbon.

All the controls can be handled by programming.

1 Creating a control

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

- Static,
- Text area,
- Formatted display control,
- Button,
- Link,
- Image,
- Clickable image,
- Thumbnail,
- Peeling corner
- Video control
- Web Camera control,
- Edit control,
- Combo box,
- List box,
- Rating,
- Captcha,
- Radio button,
- Check box,
- Slider,
- Looper,
- Linear looper,
- Image Gallery looper,
- Table,
- TreeView table,
- TreeView,
- Dashboard,
- Organizer,
- Scheduler,
- Calendar,
- Chart,
- Bar code,
- Progress bar,
- Cell,
- Popup,
- HideShow,
- Scrolling banner,
- Tab,
- Internal Page control,
- Control Template control,
- Supercontrol,
- HTML table,
- Menu,
- Pager,
- Site map,
- Site map path,
- HTML control,
- IFrame control,
- Flash control,
- Flex control,
- Silverlight control,
- Java Applet control,
- Horizontal rule,
- "Web component" control.
- Layout control.

1.1 Creating a new control

To create a control:

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

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

1.2 Creating a control associated with an item

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

- Button,
- Tab
- Horizontal rule
- HTML control
- Java Applet control
- "Web component" control
- Flash control
- Pager
- Cell and HTML table
- Site map

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

- Drag and Drop from the "Analysis" pane.
- Via the "File" tab in the description window of the control.

2 Characteristics of a control

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.

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

2.1 Displaying characteristics

To display the description window of control:

- double-click the control.
- select "Description" from the popup menu of control (right mouse click).
- select the control and press [Alt]+[Enter].

Notes:

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

2.2 Characteristics by tab

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

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

General tab

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

GUI tab

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

- Initial status of control when opening the page
- Visibility of control
- Size of control
- Anchoring, ...

Details tab

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

- input parameters
- Drag and Drop, ...

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

Link tab

The "Link" tab is used to select the item (from a data file or item) to which the control is linked. Depending on the current record, the content of linked item will be displayed in the control.

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

Content tab

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

The "Content" tab is used to define:

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

Note/Help tab

The "Note/Help" tab is used to:

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

Advanced tab

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

Style tab

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

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

- Create or modify a style.
- Choose a style.

2.3 Dynamic control, static control

WEBDEV proposes the following options for all types of controls:

- **Automatic:** the control will automatically adapt to the type of use.
- **Static:** the control cannot be modified by

programming: the different properties associated with the control will have no effect.

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

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

3 Handling controls in the editor

3.1 The handles of controls

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

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

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

- The **black handles** are used to view and resize a control.
- The **white handles** are used to view a control. This type of handles is also displayed when a page is read-only for the developer: the controls cannot be modified.

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

3.2 Available operations

The page editor allows you to:

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

4 Aligning controls

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

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

4.1 The 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 the controls found in a page.

Display the rulers

To display the rulers, on the "Display" pane, in the "Help for edit" group, check "Rulers". Two types of guides can be used:

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

Note: To view the window with or without its guides, select "Guide visible" from the popup menu of rulers. Visible or not, the guides are always enabled: any control moved toward a marker is automatically snapped by it.

Handling the markers

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

4.2 Snap-on grid

The snap-on grid is used to place 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 presentation rules issued from the Windows standard: alignment of controls, standardization of button sizes, ...

4.5 The alignment options

To align several controls, WEBDEV proposes:

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

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

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

4.6 The configurable zoom

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

To specify a precise zoom factor, you can:

- on the "Display" pane, in the "Help for edit" group, specify the requested zoom factor.
- enter the requested zoom factor in the status bar of the editor (bottom right).
- press the [Ctrl] key and modify the zoom factor with the mouse wheel while keeping the [Ctrl] key down.

5 Options for editing the controls in a page

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

The edit options are used to configure:

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

controls.

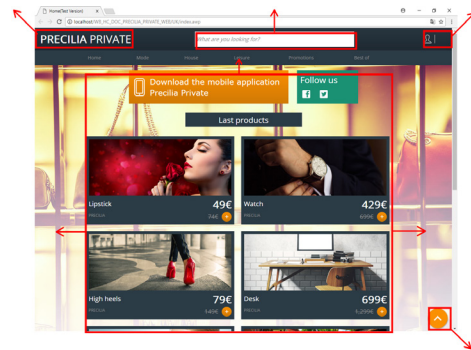
- the options for control selection.

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

To display these options, on the "Display" pane, click the group icon of "Options" group.

6 Anchoring controls

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



The anchoring of a control is managed via several parameters:

- **the anchor of control:** this parameter is used to define the modification that will be applied to the control according to the change of page size. The control can move to left and/or to bottom, it can be enlarged in width and/or in height.
- **the management of width and/or height:** this parameter is used to manage the control behavior when it is stretched. The width or the height can be adapted to the control content or to the browser. You also have the ability to 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 one or more controls to anchor.
2. Define the anchoring of controls:
 - in the "GUI" tab of the control's description window ("Description" from the popup menu).
 - via "Anchor" from the popup menu (right mouse click) of the control.
3. In this window, choose the type of anchoring that will be used. The most common options are represented by icons:



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



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



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



Centered in width: The control remains centered in width in the browser regardless of the browser width.



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



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



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



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



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



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



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



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

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



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



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



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



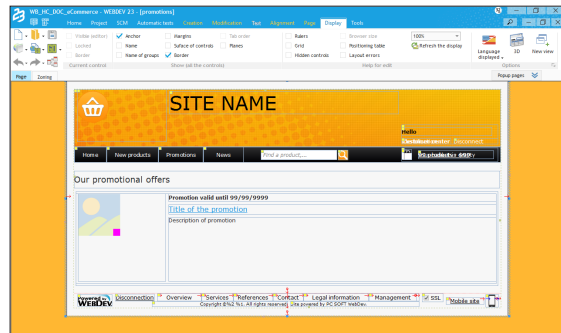
Centered: The control remains centered in height and in width in the browser regardless of the browser size.

4. Validate. If the "Anchors" option is enabled (on the "Display" pane, in the "Show (all the controls)" group), the anchoring signs are automatically displayed in the control (red arrows).

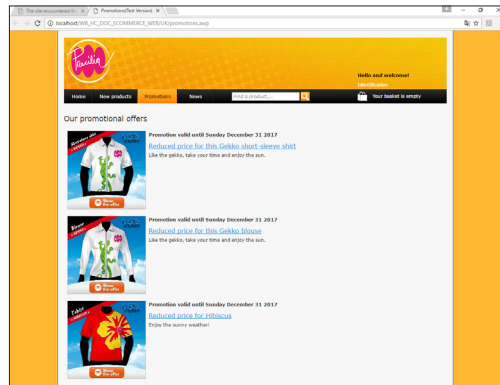
Note: To manage the anchor of controls, you also have the ability to use the positioning tables. See the online help for more details.

Repeating a group of controls: the loopers

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



Creating the looper in the WEBDEV editor



Displaying the looper control in the browser

For each looper control, several characteristics can be modified during the repetition: caption, text color, ...

The repetitions can be performed:

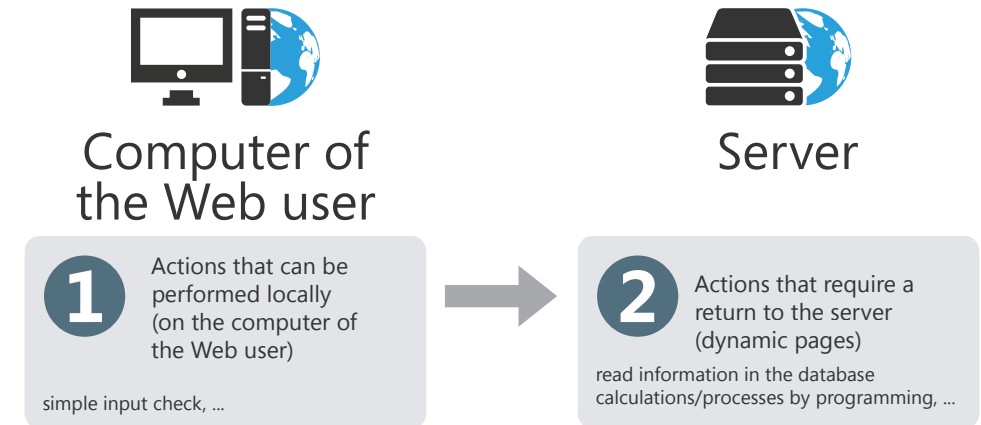
- on a column,
- on several columns. In this case, we talk of "Horizontal looper".

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

The two types of code

Server code or browser code?

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

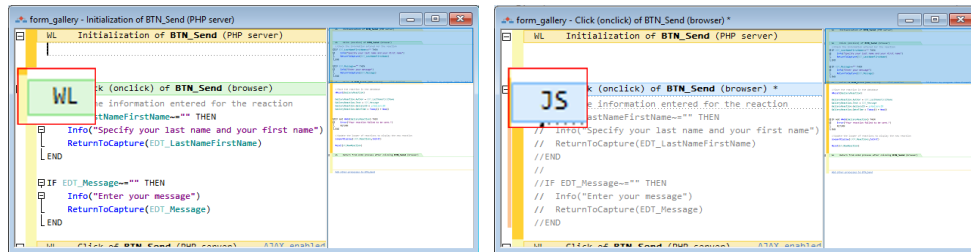


To manage these two types of actions, the code editor of WEBDEV differentiates between two types of code:

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

Browser code: WLanguage code or Javascript code?

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



- green bar = WLanguage: the WL symbol is displayed in front of the code.
- blue bar = Javascript: the JS symbol is displayed in front of the code.

To switch from a green code (WLanguage) to a blue code (Javascript), all you have to do is click the WL symbol found in front of the code caption (and conversely by clicking JS). When the page is saved, the WLanguage code is automatically converted into Javascript.

We recommend that you develop in WLanguage.

Additional events

The last "Other processes" process found in the code window is used to manage additional events. All you have to do is click "..." found in this process to display the list of additional events.

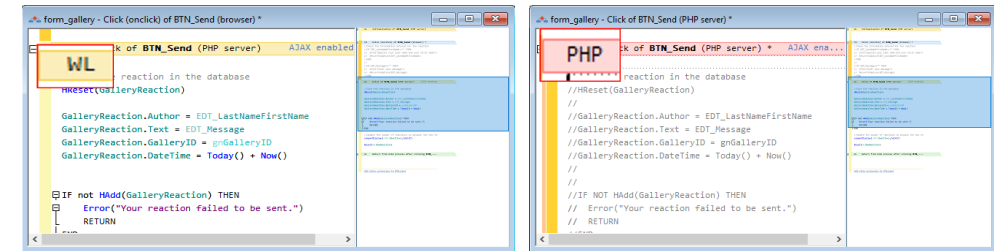
All you have to do is select the requested events to add them to the codes displayed by the code editor.

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

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

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

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



- Yellow bar = WLanguage: the WL symbol is displayed in front of the code.
- Pink bar = PHP: the PHP symbol is displayed in front of the code.

To switch from yellow code (WLanguage) to pink code (PHP), all you have to do is click the WL symbol found in front of the code (and conversely by clicking PHP). When saving the PHP page, the WLanguage code is automatically converted into PHP code.

We recommend that you develop in WLanguage.

WLanguage: a simple and powerful language

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

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

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

In WLanguage, a few lines are sufficient:

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

In Javascript, the equivalent process is as follows:

```
<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>
```

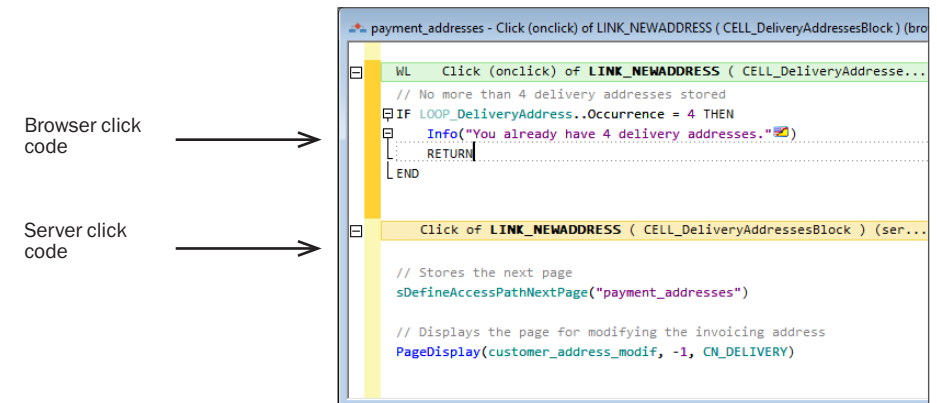
WLanguage code is easier to write, understand and maintain. It is much more reliable. Upgrading is easier.

Use WLanguage in all your processes.

Sequence for running the code of buttons/links

Two click codes are associated with the buttons/links:

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



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

The aspect of button/link and its target do not affect the sequence in which the codes are run.

Tip: To understand and remember the order in which the server and browser codes are run, imagine that you are the Web user: the Web user clicks the button in the browser therefore the browser click is run first.

Upload: Sending files to the server

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

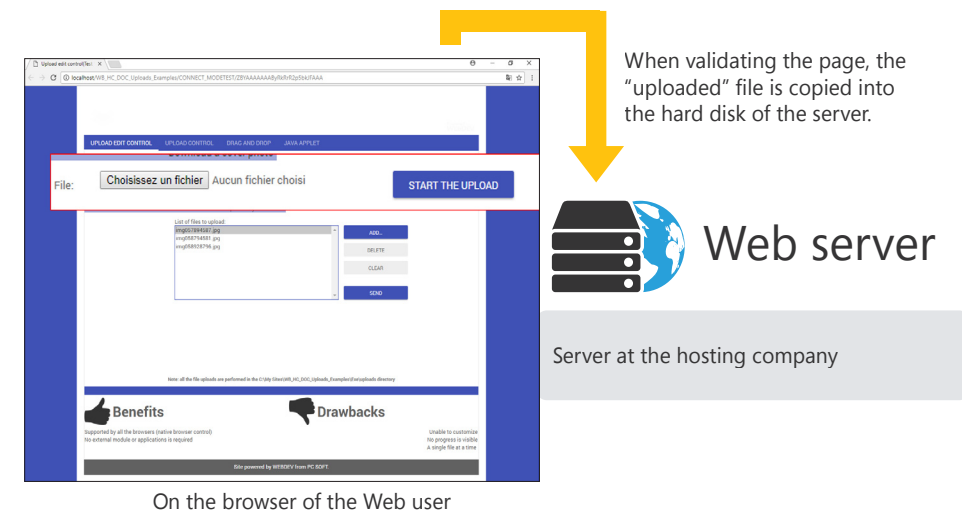
A simple example: Creating a contact in an address book.

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

The UPLOAD control allows the Web user to select the image via a "Browse" button. When the form is validated, the selected image file is sent to the server and copied onto the hard disk of the server (**UploadCopyFile**).

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

Operation	Sequence of codes run	Notes
Send to server (submit)	<p>Associated action: None Run the click code of button/link</p> <p>Code run: 1. Browser click code of control. 2. Browser code of page submit (if it exists). 3. Sending the values found in the page controls to update the page context on the server. 4. Server code of control.</p>	Page context automatically updated on the server (dynamic pages only).
	<p>Associated action: Display the page</p> <p>Code run: 1. Browser click code of control. 2. Browser code of page submit (if it exists). 3. Sending the values found in the page controls to update the page context on the server. 4. Server code of control. 5. Displaying the page.</p>	Page context automatically updated on the server (dynamic pages only). Automatic page display.
Reinitializing pages (reset)	<p>Associated action: None</p> <p>Code run: 1. Browser click code of control. 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).</p>	
None	<p>Associated action: None</p> <p>Code run: Browser code of control only.</p>	
	<p>Associated action: Run the click code of button/link</p> <p>Code run: 1. Browser click code of control. 2. Server code of control.</p>	
	<p>Associated action: Display the page</p> <p>Code run: 1. Browser click code of control. 2. Server code of control. 3. Displaying the page.</p>	Automatic page display



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

Note: WEBDEV also proposes an evolved Upload control that requires a Flash player on the computer of Web user. This control is used to manage a progress bar, to select several files, ...

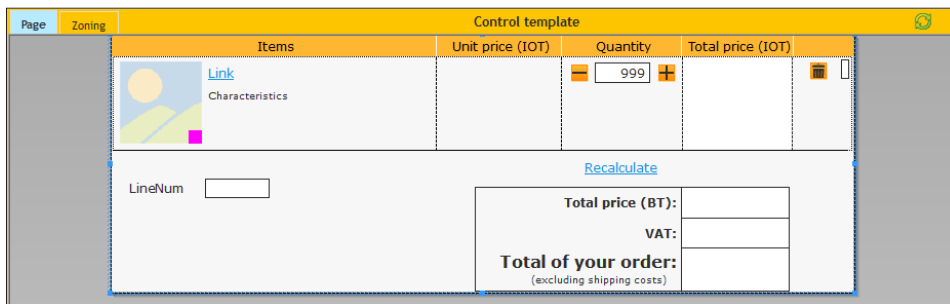
Control templates

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

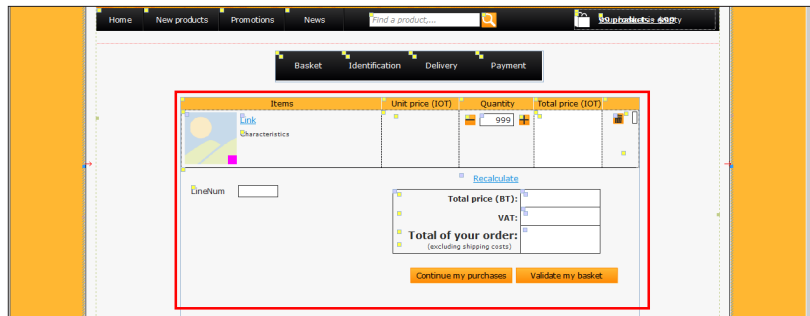
The modifications performed in a control template are automatically applied to all the pages that use this template.

A control template is used to:

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



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



Using the template in a page: the elements belonging to the template are bordered by a blue line and identified by a yellow square.

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

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

The characteristics of elements can be dissociated from the template. For example, dissociating the position of a template control to position the control somewhere else while keeping the other evolutions performed on the control (code, style, ...). We talk of **control inheritance**. In this case, the elements are identified by a blue square.

The reports

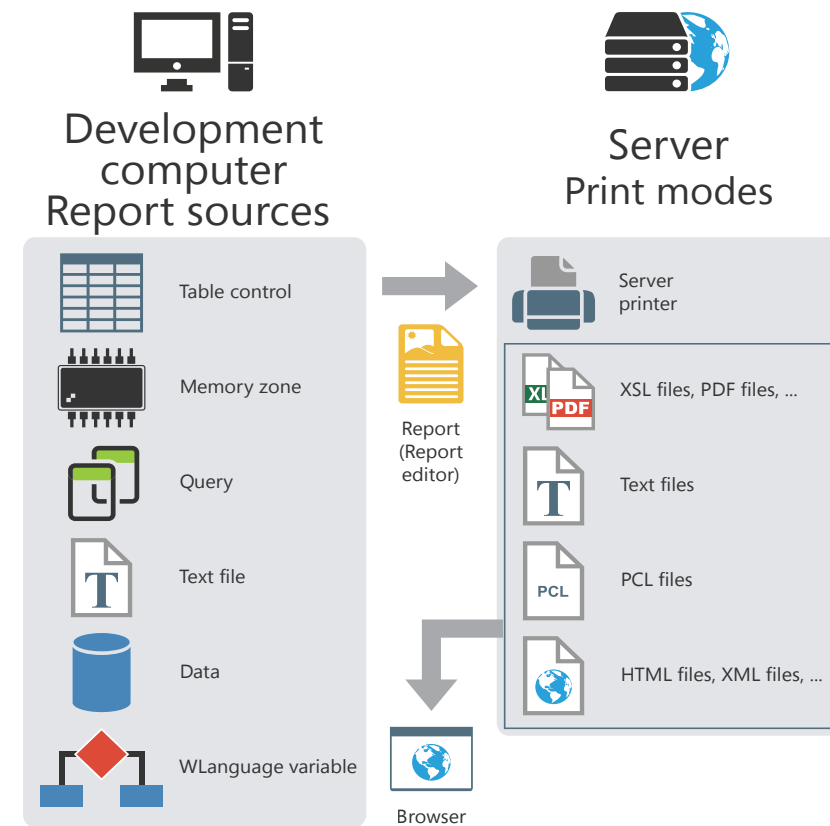
A report provides a custom view of data: data entered in the database, data found in a text file, data coming from a Table control, ...

The principle for creating a report is as follows:

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

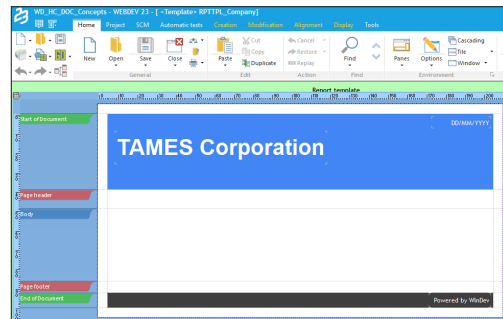
The created report can be:

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

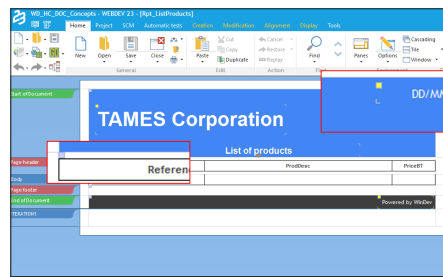
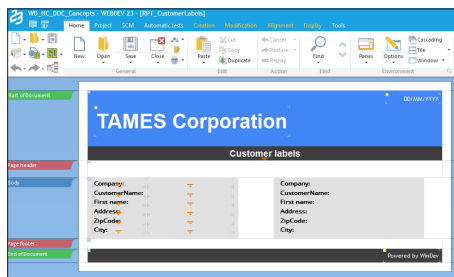


Report templates

Most of the time, the prints are using a standardized appearance and layout: date in the top right corner in a specific format, page footer with print time and file name, logo in the top left corner, ...
The report templates are used to easily standardize the layout of your reports.



Defining a report template in the report editor.



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

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

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

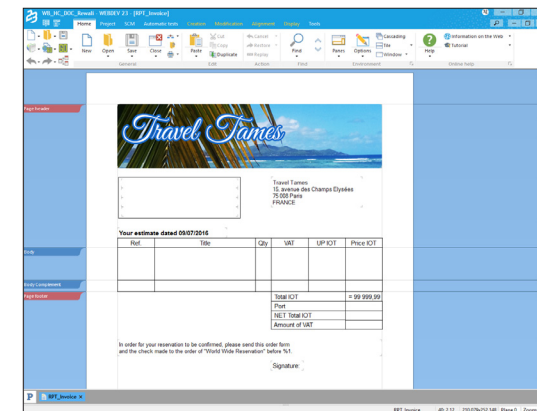
The characteristics of elements can be dissociated from the template. For example, dissociating the position of a template control to position the control somewhere else while keeping the other evolutions performed on the control (code, style, ...). We talk of **inheritance**. In this case, the elements are identified by a blue square.

Different print modes

The report editor allows you to create reports in order to print the documents generated by your site (invoices, quotes, ...).

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

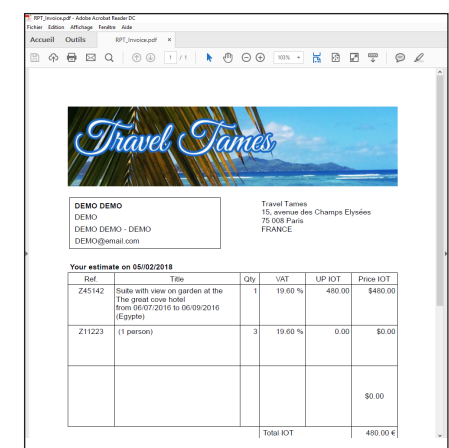
- **Print in HTML format:** The report is displayed in a new site page. The "Print" button found in the browser allows the Web user to print the report. No specific software is required.
- **Print in PDF format:** This print mode requires the presence of Acrobat Reader (or similar software) on the computer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



Report in the report editor of WEBDEV



Printing in HTML



Printing in PDF

Displaying and sequencing pages

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

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

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

Which page should be displayed?

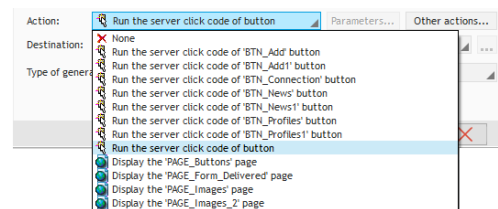
The page to display can be defined:

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

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

To define the page to display:

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



Selecting the page to display by programming

The page to display must be selected by programming when:

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

Where should the page be displayed?

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

The target can be defined:

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

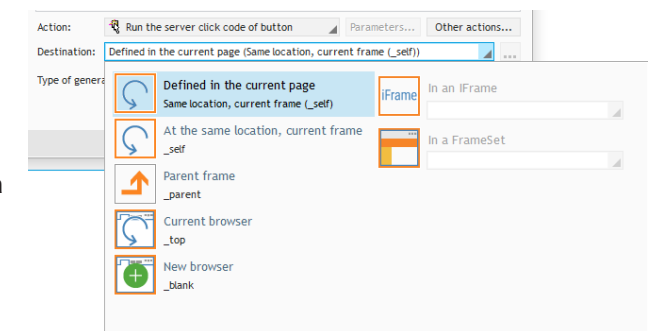
Several WLanguage functions allow you to open a new page (or frameset). The main functions are:

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

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

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

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



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

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

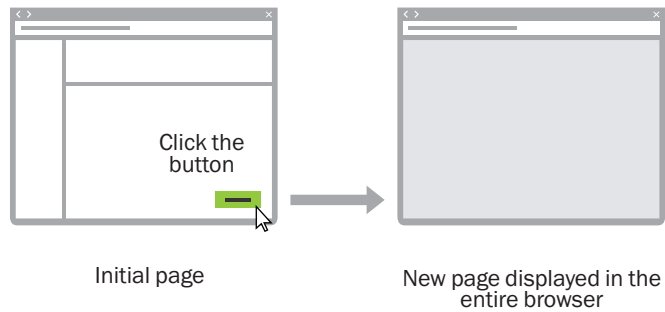
Choosing a target

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

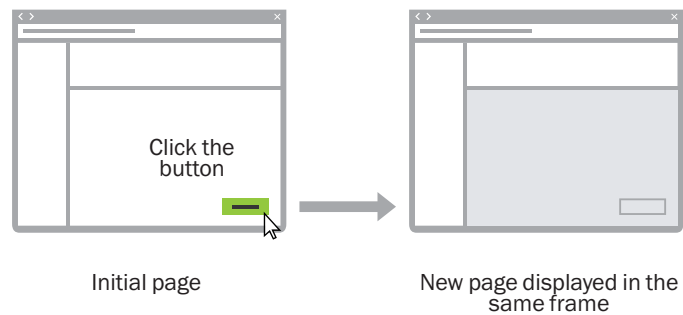
- a preset target: four preset targets are proposed:
 - Current browser (`_top`)
 - Current frame (`_self`)
 - Parent frame (`_parent`)
 - New browser (`_blank`). In this case, you will have the ability to choose the characteristics of new browser window (with or without menu bar, status bar, ...).
-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

Note: in the diagrams below, the grayed area represents the area where the page is displayed during the click on button.

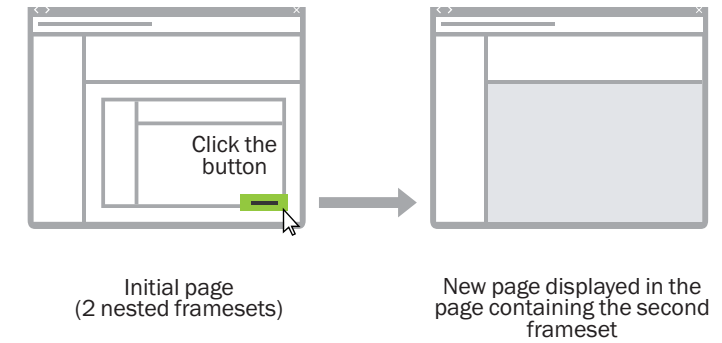
Current browser (`_top`)



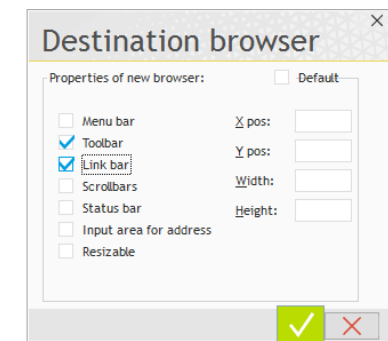
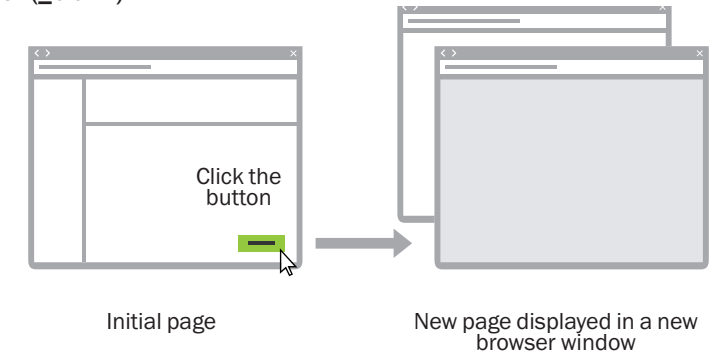
Current frame (`_self`)



Parent frame (`_parent`)



New browser (`_blank`)

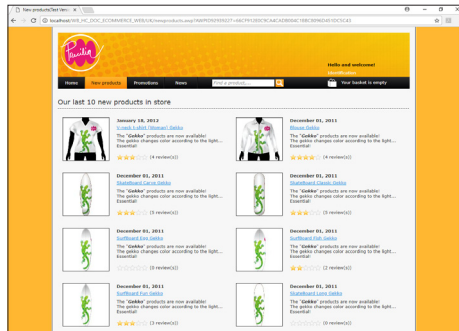


Window for defining the characteristics of new browser

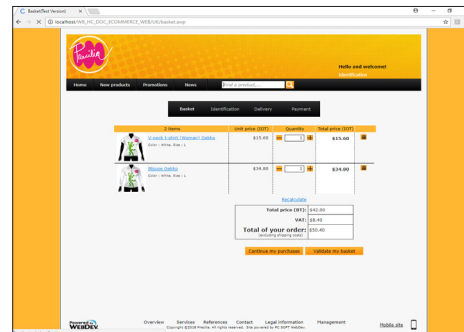
Performing a process on several successive pages

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

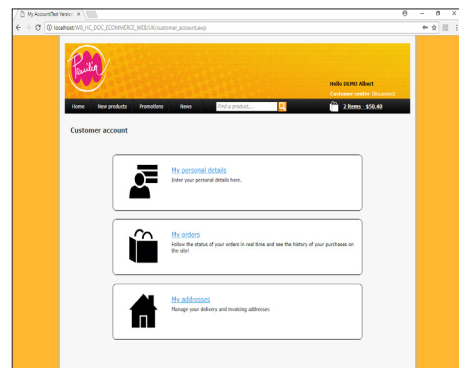
- page 1: Selecting the products to order.
- page 2: Showing the basket.
- page 3: Validating the order and typing the customer details.



Page 1: Selecting products



Page 2: Showing the basket



Page 3: Typing the customer details

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

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

Style sheets: to simplify the layout

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

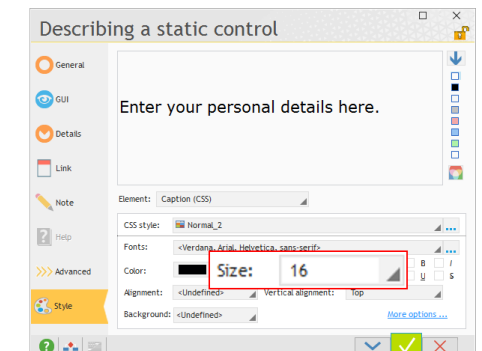
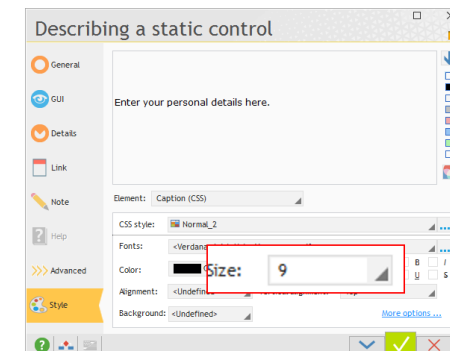
The style sheets are used to define:

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

And much more.

When a style characteristic is modified, all the controls associated with this style are updated. **This allows you to modify the style book of a site by modifying the styles used.**

Example for using style sheets: increasing the size from 9 to 16:



Note: The "Custom CSS" tab is used to type CSS code directly. Then, this code will be added into the style sheet.

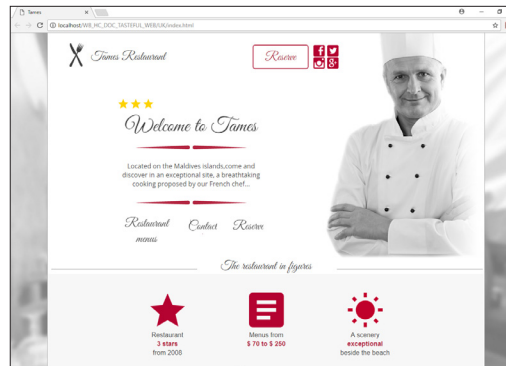
Site centered or anchored in width?

The page layout is one of the most important elements when building a site.

Two types of page layout can be used:

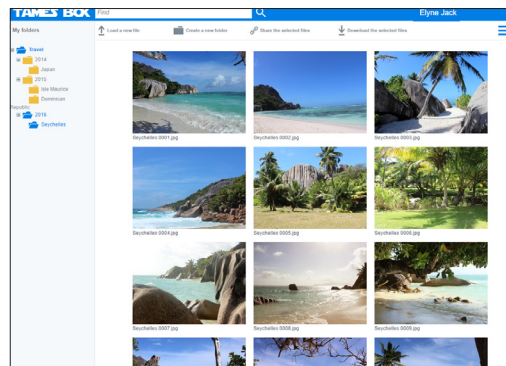
- **centered site:**

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



- **site anchored in width:**

The page content occupies the entire available space in the browser.



Notes:

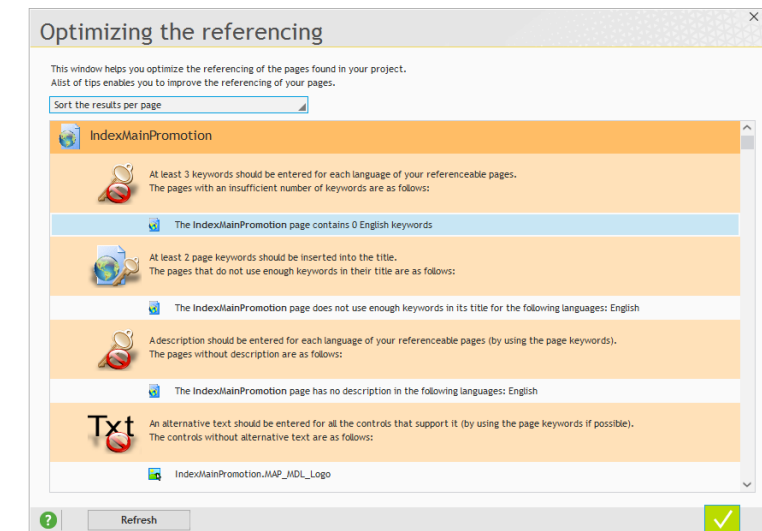
- The type of layout is defined by the options for anchoring the page ("Anchor" from the popup menu of page).
- The type of layout (centered or anchored in width) configured in a page template will be automatically used by the pages associated with this template.

Referencing a site

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

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

- a referencing wizard. This wizard lists all the optimizations that can be performed to improve the referencing of your site. To start the wizard, on the "Project" pane, in the "Web" group, expand "Referencing" and select "Optimize the referencing".



- the ability to enter a description and keywords on each page of your site.

You have the ability to include a home page in your site when developing the project, by creating a new page.

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

Referencing a site in practice

1 Principles of referencing

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

1.1 Optimizing the referencing of a WEBDEV site

To optimize the referencing of your pages in the search engines, we recommend that you use AWP pages.

To reference a dynamic site, use:

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

Note: In AWP, to pass parameters between your pages, we advise you to pass parameters via the

URL rather than using AWP contexts. Indeed, the URL content can be referenced while the AWP contexts cannot.

1.2 The available tools

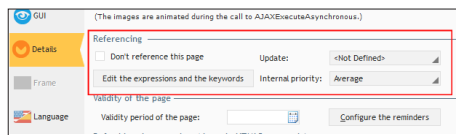
To help you, WEBDEV proposes:

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

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

To reference a static or AWP page of your site:

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



3. You can:

- Choose not to reference the current page.
- Reference the current page by associating it with expressions and keywords ("Edit the expressions and the keywords").

3 Reference a page and type its keywords

To reference a static or AWP page of your site:

1. Display the description window of page.
2. In the "Details" tab, a specific area is used to manage the page referencing. In this area, click the "Edit the expressions and the keywords" button.
3. **In the "Description" tab of referencing window**, type the page description. In most cases, this description appears when displaying the result of a search performed by a search engine. Some tips:
 - Use short sentences, limit the number of words (up to 200 characters)
 - This description must entice the Web users to display the page.

Note: This description can be modified dynamically by **..Description**.
4. **In the "Keyword" tab of referencing window**, type the keywords associated with the page. These

keywords (or expressions) will be the keywords for which the page will be referenced: the Web user types one of these keywords, the search engine will propose the corresponding page. The input of keywords is performed as follows: enter the first keyword then press the [ENTER] key to type the next one.

Some tips:

- The first three keywords are the most important ones. Use common keywords and pertinent keywords.
- The keywords must not be repeated.
- Use variations (singular/plural, noun/verb, ...).

Note: These keywords can be modified dynamically by **..Keywords**.
- 5. Validate the referencing window and the description window of page.

4 Using the referencing wizard

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

Some 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 wizard for referencing a page:

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

To start the wizard for referencing all site pages:

1. On the "Project" pane, in the "Web" group, expand "Referencing" and select "Optimize the referencing".
2. The referencing wizard starts. Double-click a suggestion to perform the corresponding optimization. The "Refresh" button is used to update the list of suggestions. The results can be sorted by page or by type of advice.

5 Registering your site in the search engines

The site referencing is performed on the search engines directly.

Note: The site referencing may not be free.

Two types of search engines are available:

- the "Automatic" search engines:

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

How to get referenced?

In most cases, a "Reference your site" link is available on this type of search engine. All you have to do is specify the site address and the Webmaster email. The site will be automatically monitored (according to keywords and to the site content) and referenced.

- the "Directory" search engines:

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

How to get referenced?

In most cases, a "Reference your site" link is available on this type of search engine.

The referencing procedure is as follows:

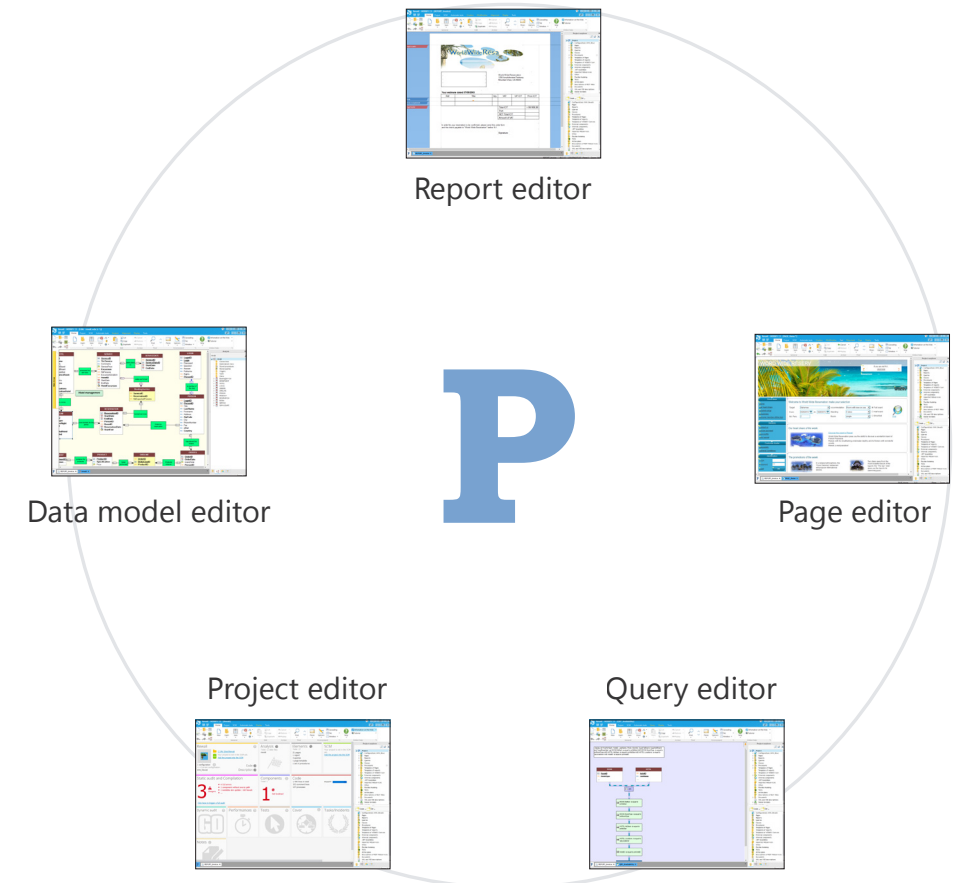
1. Choosing the site category (leisure, culture, ...).
2. Answer questions about the site to reference: Internet address of site, WebMaster email, site description, ...
3. Once your questionnaire has been sent, your site will be evaluated by a person from the company that manages the directory. This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

The WEBDEV editors

To handle a project, WEBDEV proposes several integrated editors:

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

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



User-friendly editors in practice

1 Introduction

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

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

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

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

2 The different editors

2.1 The project editor

The project editor has two main purposes:

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

Some characteristics of project editor:

- WYSIWYG editor ("What You See Is What You Get"). All project elements can be viewed directly.
- Project management via a comprehensive dashboard.
- Creating and viewing sequences between several project elements (pages, reports, ...) via the project graph.
- Interaction between the different panes and the project elements.

2.2 Data model editor

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

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

The data model editor allows two types of database descriptions:

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

Let's see some characteristics of data model editor:

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

2.3 UML editor

The UML language is a graphical language allowing you to:

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

The UML editor is mainly used to:

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

See "The UML model", page 122 for more details.

Let's see some characteristics of UML editor:

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

2.4 Query editor

The query editor is used to automatically create queries on the data files. The programming is simplified: pages, tables, combo boxes, reports, ... can be based on queries.

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

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

Let's see some characteristics of query editor:

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

2.5 Page editor

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

Several templates, skins and types of pages are proposed. They help you make your pages more ergonomic and make your sites more user friendly. Let's see some characteristics of the page editor:

- WYSIWYG editor ("What You See Is What You Get"). You can view directly your page as it will appear to the user.
- Creation and description of pages and controls.
- Drag and Drop is used to copy, paste or move controls from a page to another one.
- Page templates and preset control styles.
- Presence of several icon catalogs used to associate images with controls.
- The real-time interface checker is used to simplify the positioning of controls.
- Ability to type the captions of controls in the work area of editor.
- Management of context-sensitive help in the pages.
- Zoom in the page editor.
- Interaction between the different panes and the editor elements.

2.6 Report editor

The report editor is used to easily create printed reports.

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

Let's see some characteristics of report editor:

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

2.7 Code editor

The code editor allows you to type all processes in WLanguage (the programming language included in WEBDEV). It allows you to type the source code:

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

Let's see some characteristics of code editor:

- Automatic formatting of information typed.
- Automatic completion.
- Glossary of functions.
- Immediate detection of typos and help for correction.
- Incremental compilation.
- Previewing the different processes of a window, control or report found in the project.
- Inserting processes specific to the use of mouse

or keyboard.

- Managing the history of code modifications with ability to rollback.
- Opening several code windows (to perform code comparisons for example).
- Zoom in the code editor.
- Interaction between the different panes and the editor elements.

2.8 Image editor

The image editor is used to edit the images and icons of project (and elements).

Let's see some characteristics of image editor:

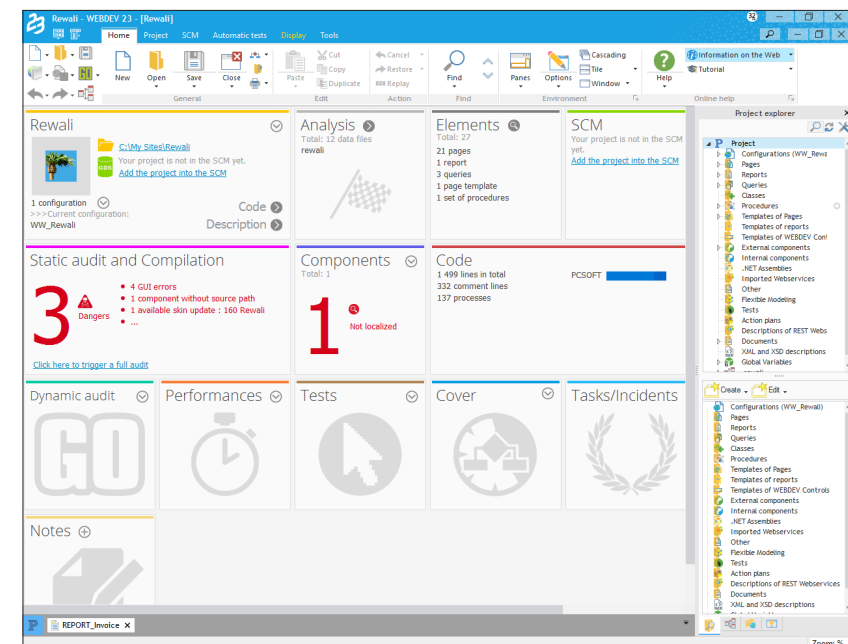
- Centering.
- Resizing.
- Symmetry, rotation.
- Managing texts.
- Managing layers.
- Access to images of image catalog.
- Preserving transparency.

Project dashboard

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

The project dashboard includes several indicators about the project content:

- statistics about the project,
- incidents,
- tasks,
- status of automatic tests,
- result of audits,
- list of elements checked out from SCM (Source Code Manager),
- result of action planes (continuous integration), ...



The elements found in this dashboard are presented in Widget format. These Widgets can be configured, moved, enabled, disabled, ... You have the ability to add new indicators.

The dashboard configuration is saved for each user. The dashboard configuration is the same for all projects that belong to the same user.

WEBDEV, WINDEV, WINDEV Mobile: 100% compatible

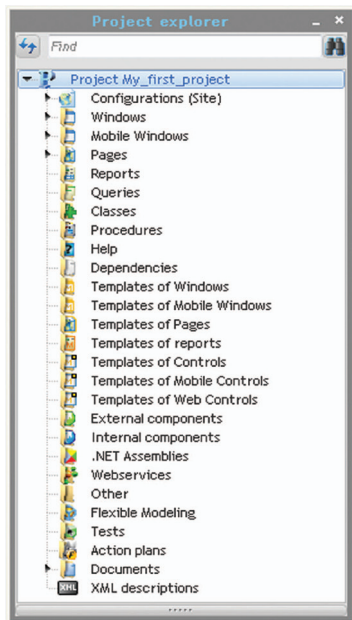
The created projects are often multi-target projects.

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

All the elements, except for the GUI (pages and windows), are 100% compatible and sharable between the WINDEV, WEBDEV and WINDEV Mobile projects.

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

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



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

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

You now have the ability to view the elements of each target from each environment. A project in WEBDEV displays the thumbnails of 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

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

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

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

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



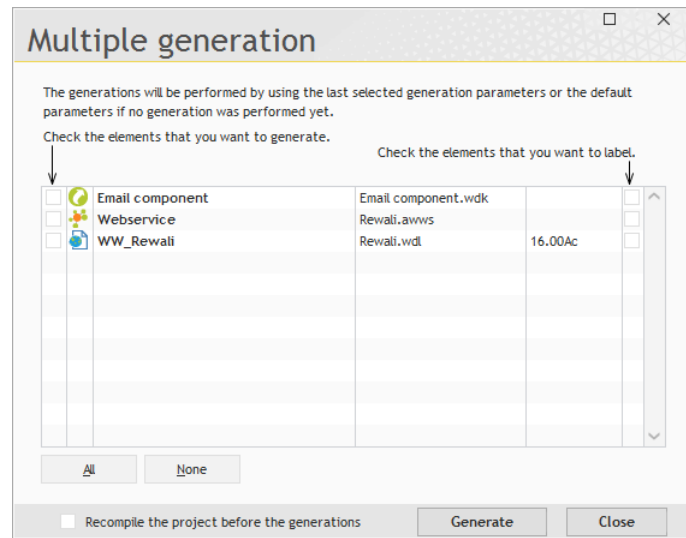
The multiple generation allows you to generate all project configurations (or some of them) in a single operation.

Multiple generation

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

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

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

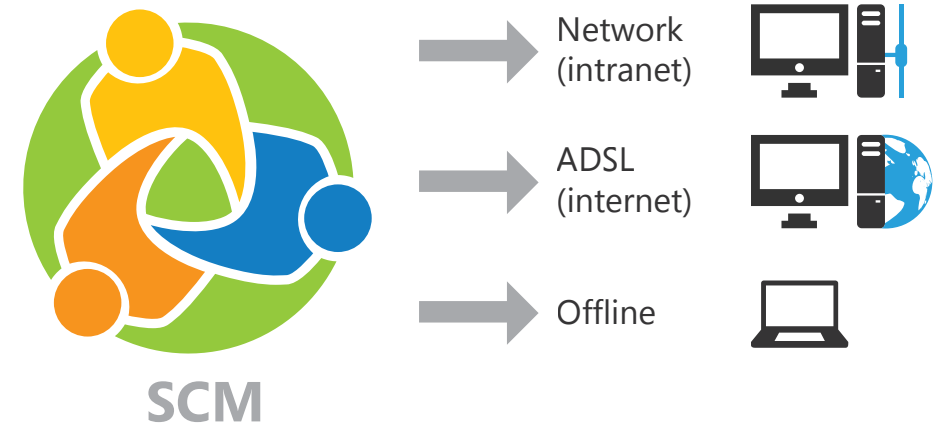


To start a multiple generation, on the "Project" pane, in the "Generation" group, expand "Generate the configuration" and click "Multiple generation".

Source Code Manager (SCM)

Overview

To simplify teamwork, WEBDEV proposes a Source Code Manager. This Source Code Manager allows several developers to work together on the same project at the same time and to share elements between several projects.

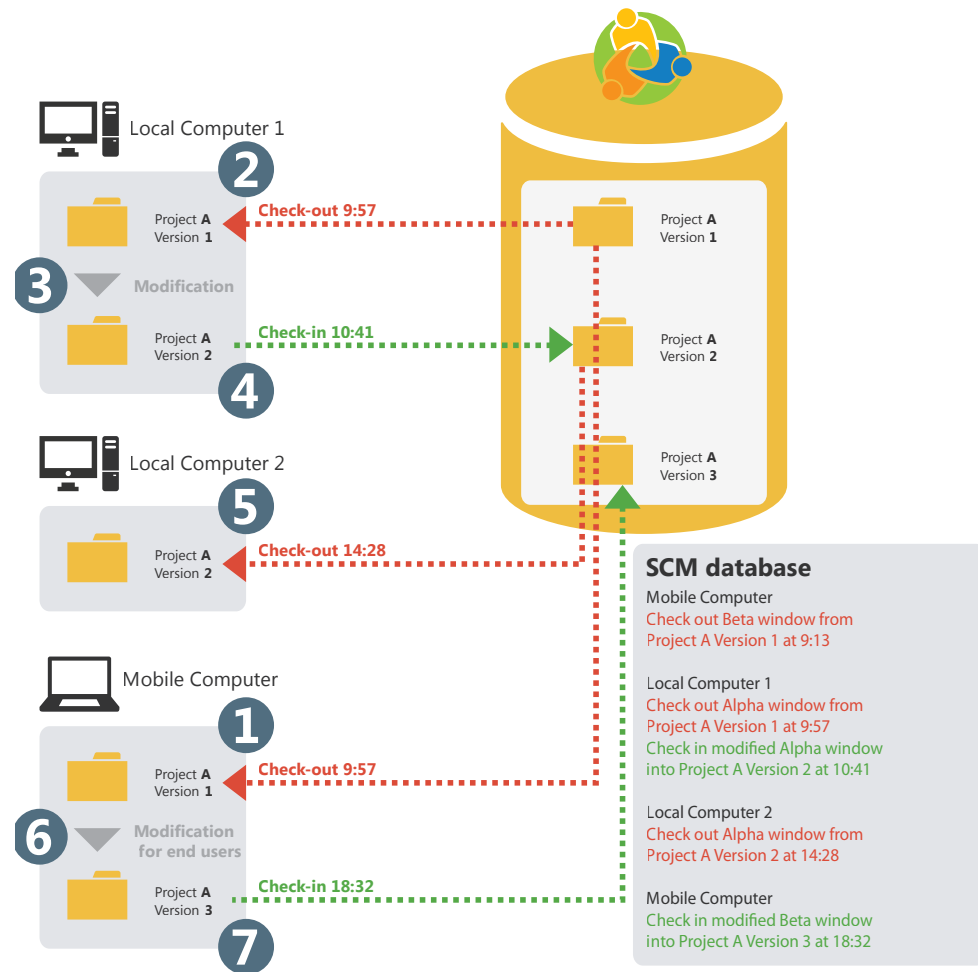


The elements found in the SCM can be shared:

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

Operating mode of Source Code Manager

The following example presents the Source Code Manager:



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

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

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

Sharing a project in practice

The development of a large IS system requires the participation of several developers. These developers must work on a single WEBDEV project while sharing the different resources (queries, classes, ...).

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

- the source code manager.
- the developer groupware. It is kept for backward compatibility with the earlier versions. See the online help for more details (keyword: "Developer groupware").

1 The Source Code Manager

1.1 Overview

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

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

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

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

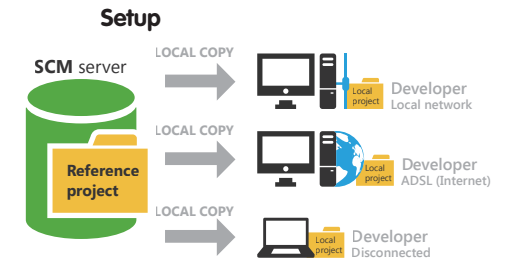
The SCM is using a database: procedures, classes, pages, reports, components, analyses, ...

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

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

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

1.2 Principle



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

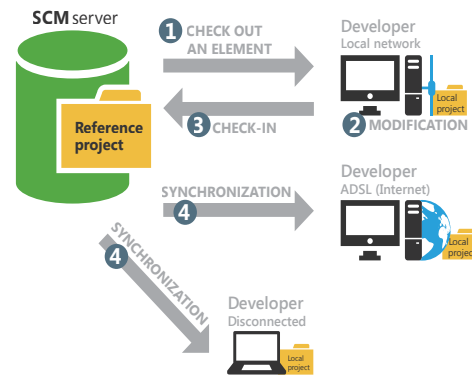
Each developer who is using the Source Code Manager retrieves a local copy of project.

Use

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

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

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



Tips

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

- Use a dedicated server with a comfortable size disk (200 GB at least)
- Use the Source Code Manager (SCM) in Client/Server mode.
- The hard disks may encounter physical problems: 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.
- Perform backups of SCM database on a regular basis (at least once a week)
- Place the server in a "secure" area and use a firewall.

2 Creating a database for SCM

2.1 Overview

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

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

The SCM database can be created:

- in a shared directory on network: all the SCM users must have full rights on this directory. The SCM database will be in HFSQL Classic format. To do so, specify the network directory where the SCM database will be created.
- on a HFSQL Client/Server server: the SCM database 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 administrator of HFSQL database.

2.2 When should a SCM database be created?

The SCM database must be created once only.

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

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

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

2.3 Backups

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

3 Configuring the project in order 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 into SCM

To add a local project into SCM:

1. Open your project in WEBDEV.
2. On the "SCM" pane, in the "Project" group, click "Add the project".
3. In the wizard, select the SCM database to use. Specify whether you want to use:

- a source database found on a network share (SCM database in HFSQL Classic format). Specify the directory of the SCM database (network computer or shared directory).
- an SCM database found on a HFSQL Client/Server server. The SCM database 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 administrator of HFSQL database.
 - an SCM Drive database. Specify the information for identifying to your SCM Drive: email, password and team.
 - an SCM database found in PCS Cloud. Specify the name of cloud platform used and the project to open.
- Note:** If no SCM database was created yet, click the "Create a database" button.
4. Validate the wizard. The project is added to 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, ...).

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

4 Working with SCM

4.1 Project options affecting the SCM

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

- **Propose to get the latest version of elements when opening the project:** When opening a project found in the SCM database, this option proposes to retrieve the latest version of project elements. By default, the latest version of elements is automatically retrieved.
- **Propose to check the elements back in when closing the project:** When the project is closed, this option is used

to display the list of elements that are currently checked out in order for some of them (or all of them) to be checked back in.

By default, the checked-out elements are not checked back in when the project is closed.

- **Check out/Check in the project automatically:** This option is used to automatically manage the "project file". If this option is checked, the project file is checked out only if the action performed requires it. Once the action was performed on the project, the project file is automatically checked back in. This option is used to disable the "Master/Guest" management on the project. This option can also be enabled on the "SCM" pane, in the "Project" group, by expanding "Master/Guest" and by selecting "Manage the project check-out automatically".

4.2 Checking out an element

The different check-out modes

The SCM proposes two modes for checking out project elements:

- **the standard mode:** if you display an SCM element that is not checked out, a dialog box indicates that this element must be checked out before it can be modified. The element can be checked out immediately (check-out button found in the dialog box).
- **the automatic mode:** if you try to modify an SCM element that is not checked out, the SCM automatically proposes to check out this element. Once the check-out is validated, the element can be modified.

Note: this mode is not recommended when using SCM with a slow Internet connection.

To change the check-out mode:

1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV".
2. In the "General" tab, the option "Checking out the elements during the first modification" is used to switch all the next opened projects to automatic mode.

Opening a project element to modify its characteristics

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

1. Check out the element from the Source Code Manager.
2. Select the check-out mode of element. The check-out mode can be:
 - **exclusive:** no one else will be able to check out this element until it is checked back in. The element can be checked out for test only.
 - **for test:** the element can be modified but the modifications will not be checked back in.
 - **multiple:** the element can be checked out by 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, all you have to do is select "Check in" from the popup menu of the element (in the project graph or in the "Project explorer" pane).

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

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

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

4.4 Management modes of project

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 required to:

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

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

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

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

Caution: Modifying the project options:

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

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

Automatic mode

With the automatic mode, the project file is checked out only if the action performed requires it (regardless of the user). Once the action was performed on the project, the project file is automatically checked back in.

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

5 Working in offline mode with SCM

The Source Code Manager allows you to easily work in offline mode (or mobile mode).

This mode allows a developer who is using a laptop computer to continue to work on a project found in the SCM database while being disconnected from the SCM database.

The principle is straightforward:

- before the disconnection, on the "SCM" pane, in the "Other actions" group, expand "Remote work" and select "Disconnect for a mobile use". Before the disconnection, we advise you to check

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

- during the reconnection, on the "SCM" pane, in the "Other actions" group, expand "Remote work" and select "Reconnect and synchronize". Then, all you have to do is check back in the modified elements.

See the online help for more details.

6 SCM administrator

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

It allows you to:

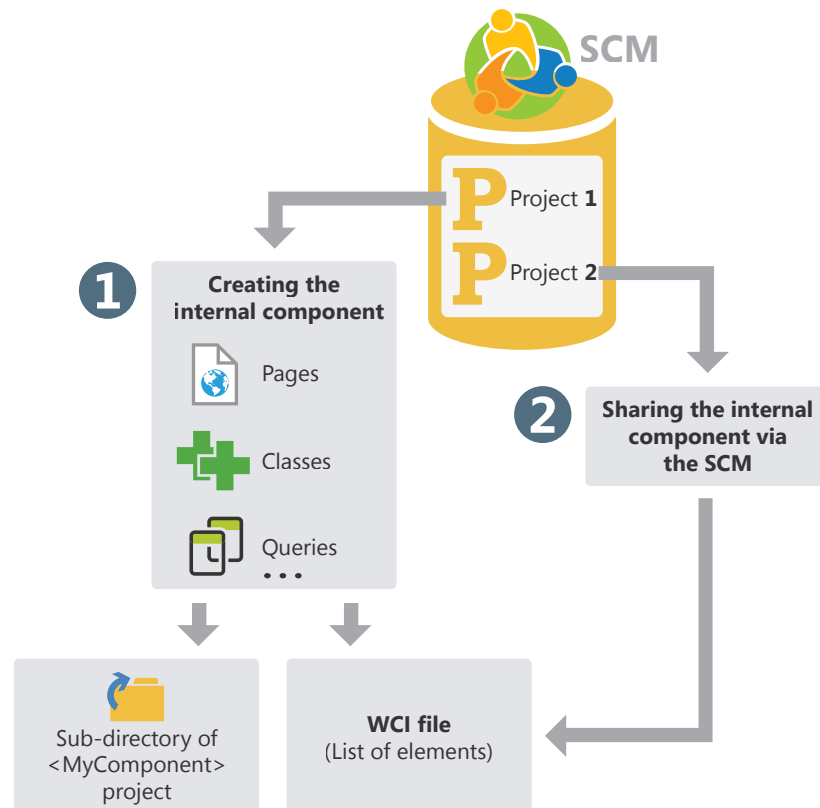
- manage the SCM databases (creation, connection to a SCM database)
- manage the branches
- manage the files and directories found in a project of SCM database (Add, delete, rename, ... files and directories)

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

Internal component

An internal component is a group of project elements. This grouping is used to:

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



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

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

Internal component in practice

1 Overview

An internal component is a group of project elements. This grouping is used to:

- **Organize a project:** the internal components can be used to group the project elements, by feature for example.
- **Share elements between different projects:** The elements found in an internal component can be shared between several projects via SCM. See "Sharing the internal components (via SCM)", page 102 for more details.

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

internal component can be debugged from the project that is using it.

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

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

2 Creating an internal component

2.1 The different steps

To create an internal component:

1. On the "Project" pane, in the "Project configuration" group, expand "New configuration" and select "Internal component". The wizard for creating an internal component starts.

2. Specify the characteristics of the internal component:

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

- its caption.

- its description.

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

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

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

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

- **Use the project analysis or no analysis (full share):** The internal component accesses the data files of project. The internal component and the project are using the same runtime contexts. This mode corresponds to the default mode if the internal component is using no analysis.
- **Use the project analysis with different runtime context (advanced mode):** The internal component accesses the data files of project. The internal component and the project are using different runtime 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 are using different runtime contexts. This mode corresponds to the default mode if the component is using an analysis. In this case, you must specify the analysis used by the internal component. You also have the ability to create a new analysis. This analysis will be associated with the internal component.
 - 6. Validate the creation of internal component. You will have the ability to:
 - Modify the characteristics of internal component via the description window of internal component.
 - Handle the internal component and its elements.
- Tip:** An internal component includes no declaration code of global variables. A set of procedures can be used to initialize the internal component.

2.2 Internal component and analysis: case of total autonomy

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

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

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

3 Sharing the internal components (via SCM)

The internal components can be shared between projects via SCM.

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

- create the internal component from a project

found in SCM. The internal component will be automatically found in SCM.

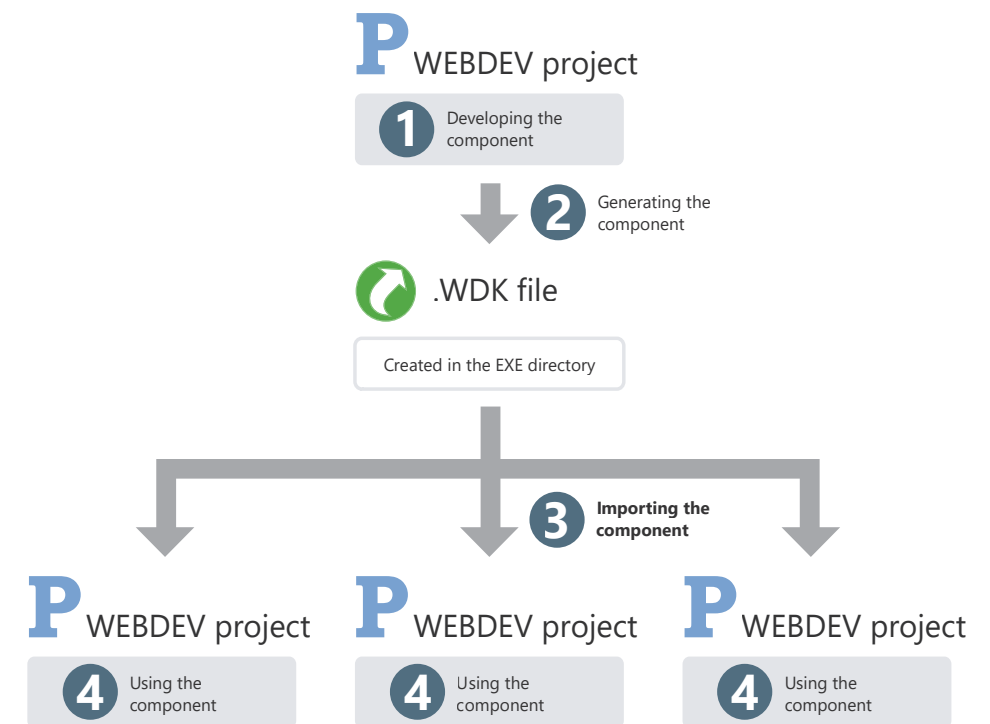
- import a project containing one or more internal components into SCM. The internal components will be automatically included in SCM.

External component

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

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

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



External component in practice

1 Overview

1.1 Definition

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

- Sending faxes
- Sending emails, ...

Notes:

- To optimize the organization of your projects, you have the ability to use internal components. See "Internal component", page 100 for more details.
- In the rest of this chapter, "component" means "external component".

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

When creating the component, the author specifies:

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

Note: Examples of components are supplied with WEBDEV. These components are accessible from the home window.

1.2 What is an external component made of?

A component is defined and generated from an existing WEBDEV project. Then, it can be included in other WEBDEV projects.

All component elements are independent of the project in which the component is included. The component can have its own analysis, pages, reports, data files, etc. When creating the component generation, all you have to do is specify whether these elements can be handled (or not) in the WEBDEV project that includes this component.

The component includes three files:

<ComponentName>.WDK	Component file. Contains all the elements of the component. This file is required to include the component in a project. This file must also be supplied in the setup procedure of the site that is using the component.
<ComponentName>.WDI	Description of component (for developing sites only). This file is required to include the component in a project. This file must not be supplied in the setup procedure of the site that is using the component.
<ComponentName>.WDO (optional file)	Optional file File in text format containing the list of additional elements supplied with the component (data files, .INI files, ...). See "Internal page", page 46.

1.3 What is included in an external component?

An external component contains:

- the different elements to distribute (pages, reports, ...). These elements can be accessible (or not) when the component is imported into a WEBDEV project.
- a short description of component.

- a help allowing this component to be re-used. This help is generated from the code comments.

Note: By default, if a component is using an analysis, files, ... the HFSQL functions will handle these elements in an independent context. These parameters can be modified in the advanced options of component. See the online help for more details.

2 Creating and generating an external component

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

The creation of a component is performed in several steps:

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

Then, the component can be distributed and re-used.

3 Distributing an external component

3.1 Overview

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

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

3.2 Direct use of the external component

The component is created and used on the same computer.

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

Caution: If the component is using specific files (data files, text files, ...), a <ComponentName>.WDO file must be created once the component is generated. This file contains the list of external files (data files, text files, ...) used by the component. These files referenced in <ComponentName>.WDO will be automatically copied into the EXE directory of the project that is using the WEBDEV component.

3.3 Simple distribution of external component

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

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

To distribute a component, you must supply:

- The files automatically generated by WEBDEV (<ComponentName>.WDK file and <ComponentName>.WDI file).
- If necessary, the specific files handled by the component (data files, initialization files, ...) as well as <ComponentName>.WDO. The <ComponentName>.WDO file contains the list of files that must be supplied with the component. See "Characteristics of current connections", page 224 for more details.

3.4 Distribution via a setup procedure

The distribution of components via a setup procedure consists in supplying a setup program to the users of WEBDEV component. This program installs all the files required for using the component in the directory specified by the user.

This setup mode is used to automatically manage:

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

4 Using an external component in a site

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

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

See "Distributing an external component", page

- the setup of specific tools (MDAC, ODBC driver for HFSQL, ...).
- the automatic update of data files used by the component (if necessary).
- the uninstall program of component.

To propose a setup procedure for a component:

1. Create the setup procedure of component: on the "Project" pane, expand "Deploy the site" and select "Deploy via physical media".

2. In the wizard planes, specify:

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

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

Reminder: When creating the setup program of an application, the necessary files are created in the INSTALL directory of project.

105 for more details.

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

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

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

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

5.1 Overview

The method for deploying a site that is using one or more components is the same as the method for deploying a standard site: on the "Project" pane, in the "Generation" group, expand "Deploy the site".

5 Deploying a site containing an external component

5.2 Updating the components and the deployed sites

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

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

See the online help for more details about the compatibility between the site and its component.

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

1. The recompilation of host project

Recompile the project that is using the component

6 Modifying an external component

6.1 Overview

A component that was created and generated can be modified at any time. This modification can correspond to:

- adding elements into the component.
- deleting elements from the component.
- modifying rights on the component elements.
- modifying one of the component elements.

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

6.2 The different types of compatibility

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

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

- New features have been added into the component and they must be taken into account.
- The parameters of some procedures have been modified.
- Incompatibility between the new version and the earlier versions of component.
- ...

2. Direct distribution of the .WDK file

Supply an update of the component (.WDK file) to the users without recompiling the project.

In most cases, this possibility applies when:

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

use the component in deployment may encounter runtime problems if they are not synchronized with this component.

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

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

7 Advanced characteristics of external component

7.1 Automatic documentation

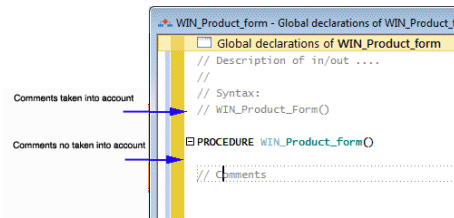
A component must be supplied with a documentation in order to be re-used. WEBDEV simplifies the creation of this documentation by proposing:

- a **general component overview**. This general overview is typed when generating the component. When using the component in a WEBDEV site, this overview will be visible when the component found in the "Wizards, Examples and Components" pane is hovered by the mouse cursor. See the online help for more details.
- an **automatic generation of technical documentation** from the comments inserted into the code of component elements.

Which code comments are taken into account?

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

- The comments found at the beginning of WLanguage procedures



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

When is the documentation generated?

The component documentation is generated during the first generation of component ("Generate the configuration" option on the "Project" pane, in the "Generation" group).

When generating the documentation:

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

Note: Whenever re-generating the component, you have the ability to regenerate the documentation associated with the component ("Regenerate" button in the "History of component" window).

Caution: The modifications performed in the generation wizard will be deleted if you click the "Regenerate" button.

How to access the component documentation?

The general component overview and the technical component documentation are available:

- when including a component in a project (on the "Project" pane, in the "Project" group, expand "Import" and select "Import an external component").
- when double-clicking the component icon found in the "Project explorer" pane or when selecting "Description" from the popup menu of component.

To get the documentation specific to a component element, double-click this element ("Project explorer" pane or project graph) or press the [F2] key from the code of this element.

7.2 Visibility of component element

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

- If the element is accessible, the component user will see this element in the list of project elements. The component user will be able to handle these elements by programming (like any other project element).
 - Note:** However, the code of this element is not visible.
- If the element is not accessible, the user will not even know that this element exists.

Caution: Depending on the declaration mode of project elements (class, set of procedures, ...), the accessible elements may change.

7.3 The WDO file

When generating the external component, two files are automatically created in the EXE directory of 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: <ul style="list-style-type: none"> - 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 two files must be distributed along with the component.

If the component is using additional elements (data files, ...), the following elements must be added into the EXE directory of project:

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

7.4 What is this WDO file?

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

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

This ".WDO" file can contain:

- the **full file name**.
C:\Components\PickerComponent\InitialStatus.INI
- the **file name**. This file will be sought in the current directory of component. InitialStatus.INI
- a **file name that is using a relative path**. The possible syntaxes are:
 - Directory\FileName.xxx to specify a sub-directory of current directory
 - .\FileName.xxx to specify the current directory
 - ..\FileName.xxx 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 path where the files are installed on the development computer of component.

When including the component, the tree structure specified in the WDO file will be stored and reproduced in the EXE directory of project. See "Using an external component in a site", page 106 for more details.

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

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

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

Distributing a component with WDO

To distribute a component that is using a WDO file:

- If no setup procedure is used for the component, you must supply:
 - the WDK file,
 - the WDI file,
 - the WDO file,
 - all the necessary files referenced in the WDO file.
- If you are using a setup procedure for the component, the WDO file will be automatically created when creating the setup procedure of component.

In the wizard, you must:

 1. Ask to modify the files to install ("Modify the list of files to install" option in the "Additional modules").
 2. Select the additional files to install.

The WDO file will be automatically created and installed with the component.

Generation modes

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

**Sites**

The sites are the most common generation mode. WEBDEV allows you to generate:

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

**Libraries**

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

**Gadgets**

A gadget is a specific type of Web site intended to be included as a mini-application on the Windows desktop (Vista and later). WEBDEV provides the necessary elements for creating and distributing gadgets.

**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.

Note: 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 even knowing the PHP language. The site can be entirely developed in WLanguage, like any standard WEBDEV site. However, you have the ability to enter your own PHP code (server code).

1.1 What is a PHP site?

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

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

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

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

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

Note: PHP version 4.3.2 (or later) is required. A MySQL database is required. The PHP sites generated by WEBDEV can also be used by PHP servers version 5 or 7.

1.2 Why develop a PHP site with WEBDEV?


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

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

2 Features of PHP project

2.1 Creating a PHP project

The method for creating a PHP project is the same as the method for creating any WEBDEV project:

1. Click  among the quick access buttons of WEBDEV. The window for creating a new element is displayed: click "Project".
2. The wizard for project creation starts.
3. In one of the first screens, the wizard allows you to choose the generation mode of your project. To do so, select "PHP" and follow the wizard.

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

- The PHP RAD that generates the PHP code from a project for several WEBDEV features.

- The compiler indicates the functions or controls that cannot be used in PHP. A purple message is displayed in the "Compilation errors" pane.
- Deploying the PHP site is proposed via WDeploy.

Note: We advise you create a new WEBDEV project to develop a PHP site rather than transform a WEBDEV site into PHP.

2.2 PHP RAD

Like for any project associated with an analysis, the RAD can be used to generate the pages of your site. The RAD takes into account all the specific features of PHP generation (controls, processes, functions that can be used in a PHP site generated from WEBDEV).

2.3 Running the test of a PHP site in the editor

During the test of a PHP site, the site is compiled in PHP and it is started in your browser. Reminder: A PHP engine is required on the development computer.

Compiling a PHP site allows you to:

- find out the WLanguage programming errors
- see the controls, processes, WLanguage functions

not available in PHP generation. See the online help for more details.

2.4 Deploying a PHP site

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

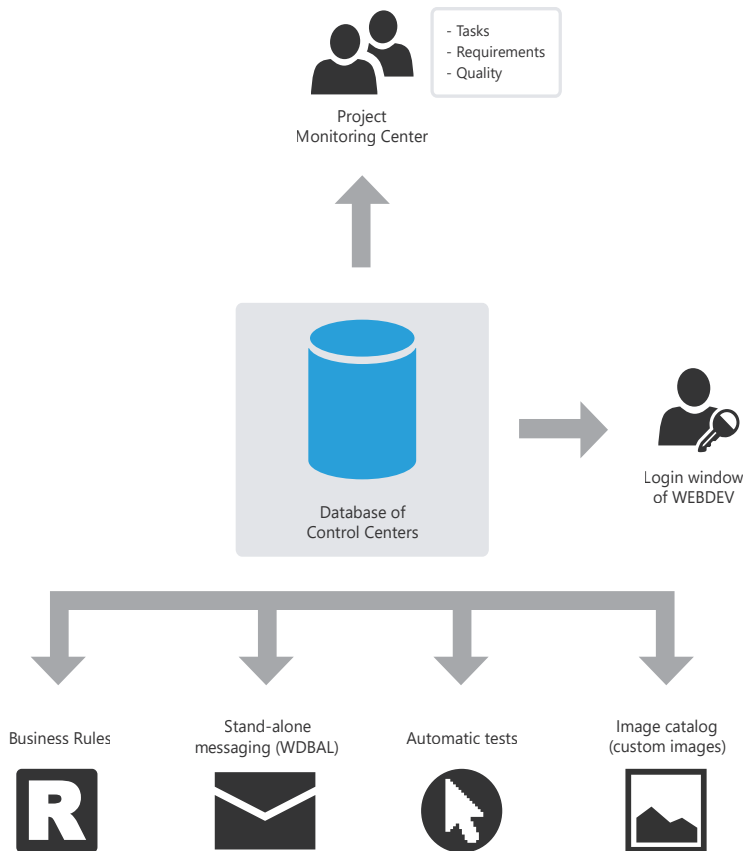
See the online help for more details.

Project Monitoring Center

To manage the life cycle of your development projects, WEBDEV proposes to use the Project Monitoring Center. The Project Monitoring Center is used to:

- Manage the project requirements,
- Manage the project monitoring (schedule of tasks to perform),
- Manage the bugs and evolutions signaled by the users on a project.

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



Note: When installing WEBDEV, the setup program proposes:

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

Managing requirements

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

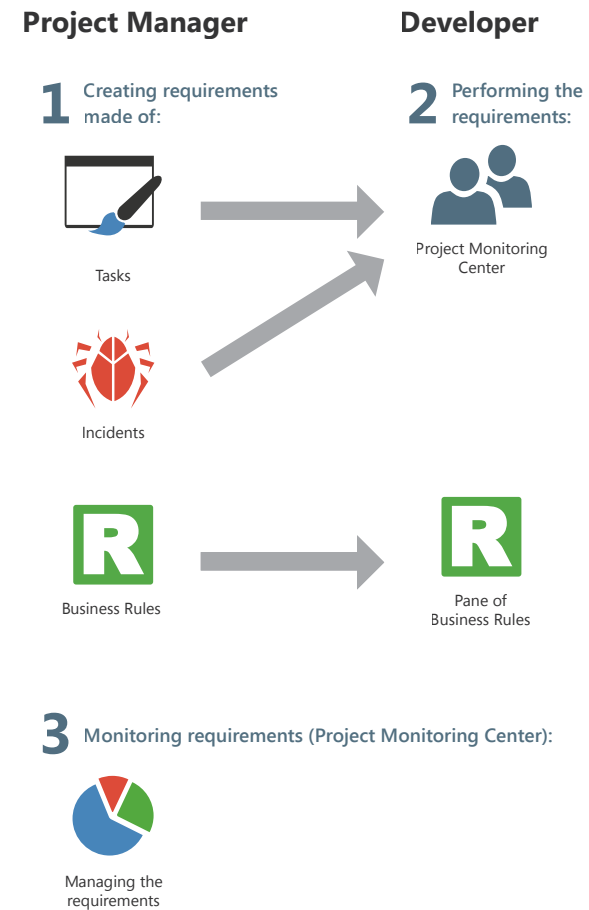
The Project Monitoring Center allows the project manager to manage the requirements of a development project.

To do so, you must:

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

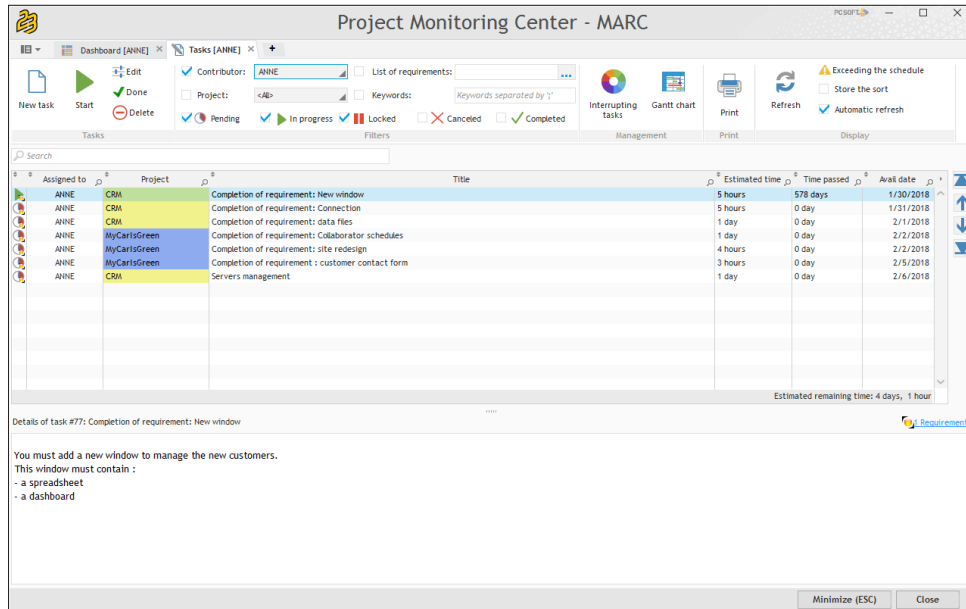
Each developer performs the different tasks assigned to him.

The project manager can follow the progress status of 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 requirements and they can correspond to several projects.



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

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

A task can be linked to a project element (page, report, ...). Whenever the relevant element is opened, the time spent on this element is counted and stored in the Project Monitoring Center. Conversely, the element corresponding to the task that you want to perform can be automatically opened from the task list. Each developer can also see his own task list in the "Project Monitoring Center" pane found in the editor.

Managing business rules

WEBDEV allows you to manage the business rules. A business rule is used to define a specific operating mode or to specify a particular process. For example: calculating a specific VAT rate, the rules for changing the status of a customer, the formula for calculating shipping costs, a sales commission, a discount rate, a decay coefficient, ...



A business rule can be simple or complex.

- The business rules can come from:
- the specifications (corresponding to requirements).
 - the suggestions made by the application users.

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

PART 4

**Advanced
concepts**



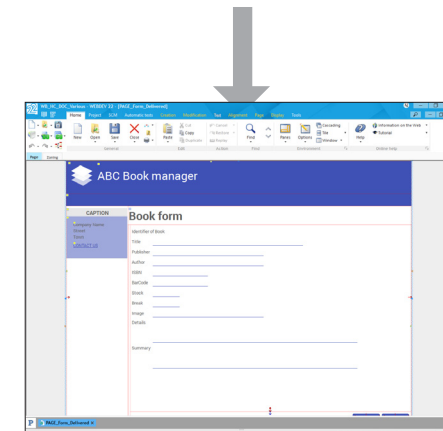
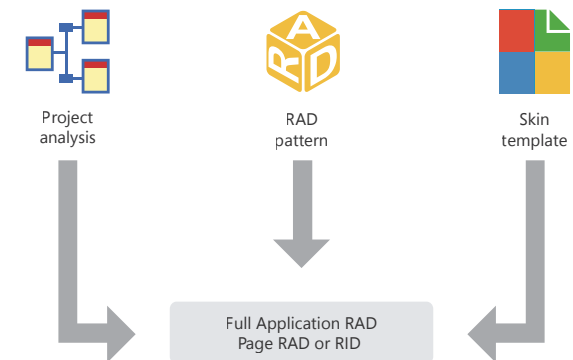
RAD RID

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

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

In RAD generation, the generated pages contain the entire code required for them to operate. The test of these pages can be run immediately with the data found on the development computer.

In RID generation, the generated pages only contain the controls linked to the analysis items. The code required for these pages to operate must be implemented by the developer. Your custom code can be typed directly.

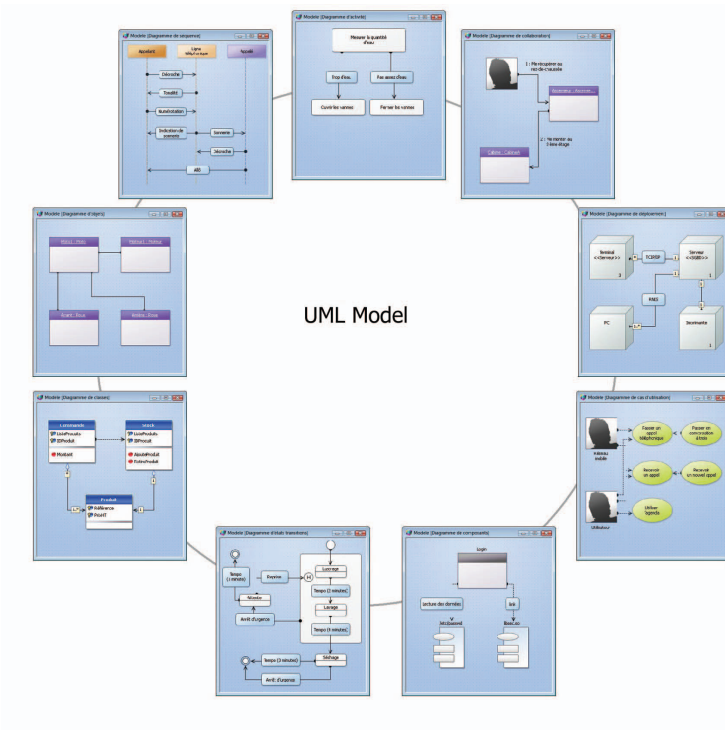


Example of page generated by RAD or RID

The UML model

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

- **Class diagram:** describes the overall structure of a system.
- **Use case diagram:** represents the 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.



The UML model in practice

1 Overview

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

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

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

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

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

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

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

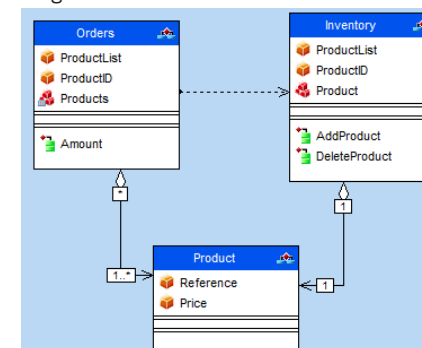
2 The different UML diagrams

2.1 Class diagram

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

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

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



A class diagram contains the following elements:

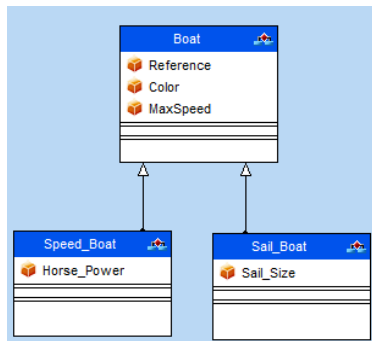
- **Class:** represents the application structures. Each class is divided into three compartments:
 - *the class name* indicates what the class is and not what it does.
 - *the class attributes* give the class characteristics.
 - *the class operations* represent the possible actions on the class.

For example, the Stock class contains the ProductList attribute. This class also groups the AddProduct and RemoveProduct operations. These operations can be applied to the class instances.

Note: The UML language defines three visibility levels for the attributes and for the operations:

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

- **Relationship:** describes the behavior of classes between themselves. Three types of relationships are available:
 - **Association:** Structural relationship between classes. For example, the *Order* class is linked to the *Product* and *Customer* classes. A Customer can place several Orders. An order includes several products. An order must necessarily contain at least one product.
 - **Dependency:** Use relationship establishing that the instances of a class are linked to the instances of another element. For example, the *Order* class is using the *Stock* class: before adding a product to an order, you must check whether this product is found in stocks.
 - **Generalization:** Relationship between a general class (parent) and a specific class (child) that derives from it. For example, the *Sail Boat* class and *Speed Boat* class are derived from the *Boat* class.



- **Packages:** used to divide and organize the diagram representation (like the directories organize the files). Each package can contain classes and relationships.

Via the generation of a class diagram, you have the ability to create the structure of the WEBDEV classes used in your application.

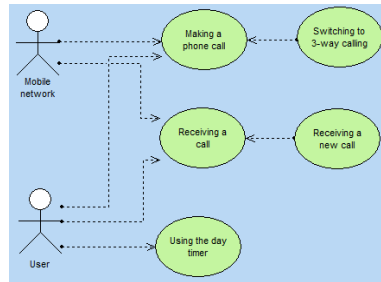
2.2 Use case diagram

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

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

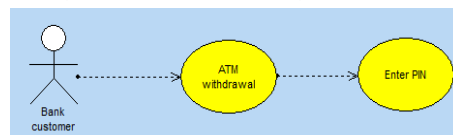
For example, the behavior of a cell phone can be

described via a use case diagram.

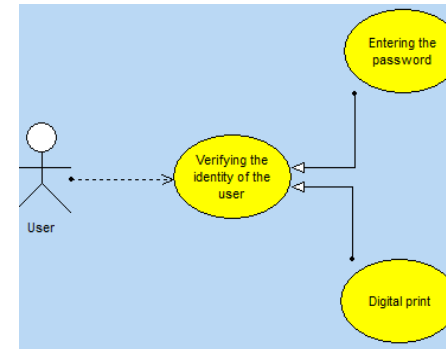


A use case diagram contains the following elements:

- **Actor:** represents the role of application users. For example, a person who works in a bank will be the . If this person has an account in this bank, he will also play the role of *Customer*.
- **Use case:** describes a sequence of actions run by the application. For example, Place an order, type an invoice, Create a new customer formA 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 actors in relation to use cases. Three types of relationships are available:
 - **Association:** Structural relationship between two linked elements.
 - **Dependency:** Relationship establishing that an element is using another one. For example, a bank *customer* may *get cash* from an ATM. In this case, the *action* depends on the *In* order to get cash, the *Customer* must type his *PIN number*. In this case, the *Get cash* action depends on the *Password input*.



- **Generalization:** relationship used to organize the elements according to a hierarchy. For example:
 - two types of *Customer* actor are available: *Individual customer* or *Company customer*.
 - the identity check can be performed according to two methods: typing a password or checking a fingerprint.



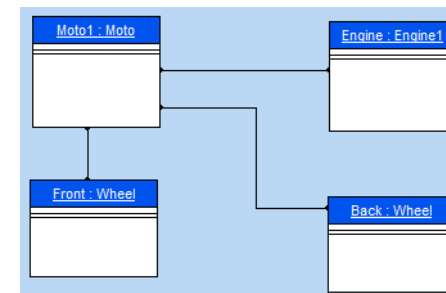
- **Package:** divides and organizes the diagram representation (like the directories organize the files). Each package can contain actors and use cases.

2.3 Object diagram

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

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

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



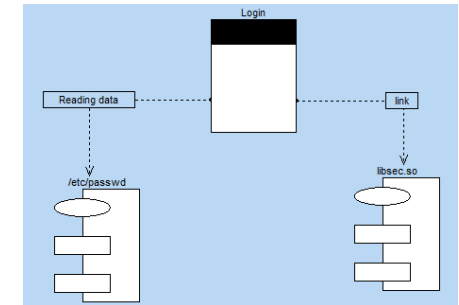
An object diagram includes the following elements:

- **object:** represents a class instanceNote: If a class diagram is opened, you have the ability create an object from a class found in this diagram (Drag and Drop from the "UML analysis" pane).
- **composite object:** visually represents an object made of other objects. For example: a window that contains scrollbars, buttons, ...
- **link:** presents the relationships between the different objects.

2.4 Component diagram

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

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



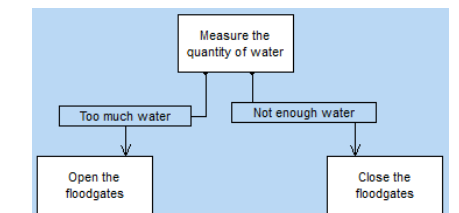
A component diagram includes the following elements:

- **module:** represents the different physical elements that constitute a computer application. For example: a file, a libraryA module can be represented:
 - by a **specification** showing the module interface. This specification can be **generic** for the customizable classes.
 - by its **body**, that presents the module implementation.
- **task:** represents a component that includes its own control flow (thread).
- **main programs** of application.
- **sub-programs:** groups the procedures and functions that don't belong to classes.

2.5 Activity diagram

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

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



An activity diagram includes the following elements:

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

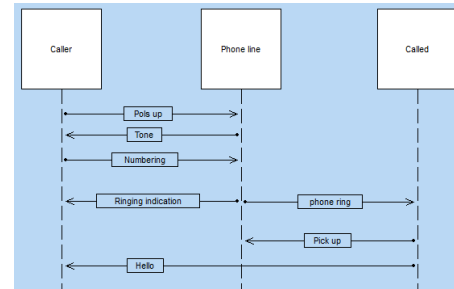
2.6 Sequence diagram

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

A sequence diagram includes the following elements:

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

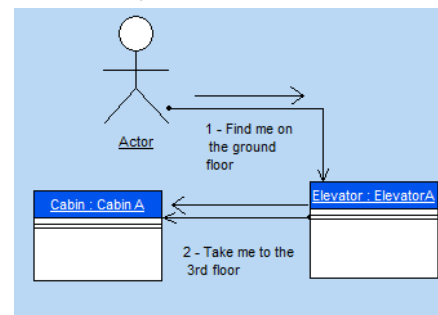
For example, the following diagram represents the beginning of a phone call:



2.7 Collaboration diagram

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

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



A collaboration diagram includes the following elements:

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

2.8 State-transition diagram

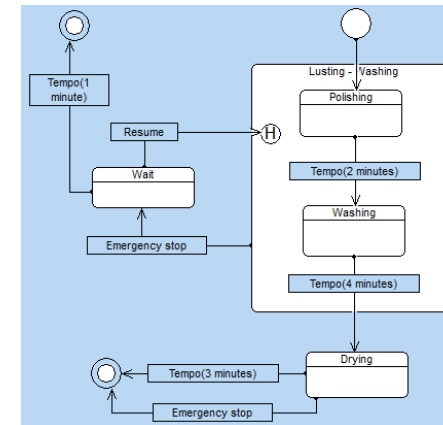
A state-transition diagram presents a sequence of states that an object goes through during its lifecycle. It is used to describe the changes of states for an object or for a component.

A state is defined by its duration and by its stability. A transition represents the instantaneous change from one state to another one.

A transition is triggered:

- by an event.
- automatically when no triggering event is specified.

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



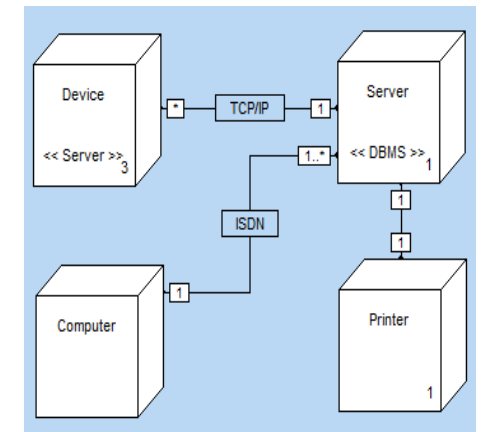
A state-transition diagram includes the following elements:

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

2.9 Deployment diagram

A deployment diagram presents the physical layout of hardware devices (nodes) used in a system as well as the association between the executable programs and these devices.

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



A deployment diagram includes the following elements:

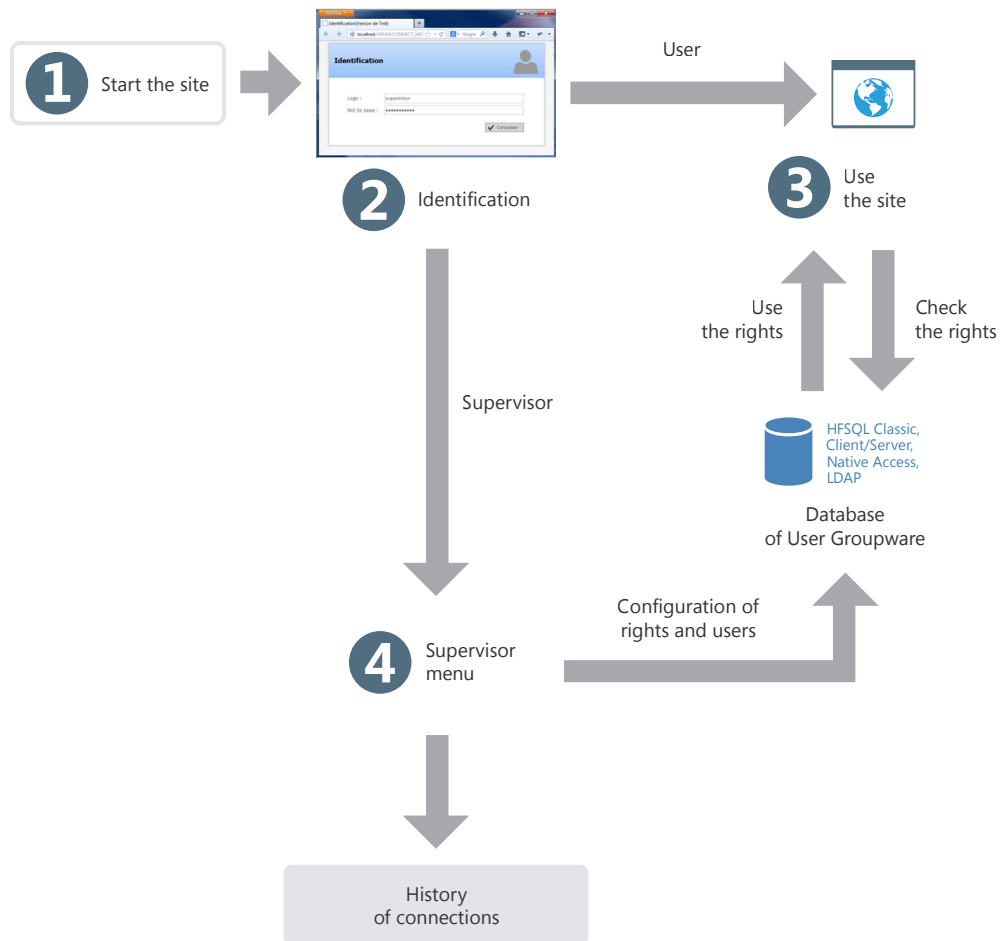
- **node class:** represents a class of hardware resources. For example: a server, a PC, a printer, ...
- **node instance:** represents a hardware resource. For example: the server #3, the printer #7, ...
- **connection:** describes the communication support between two nodes. For example: RNIS or TCP/IP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

User groupware

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

Two types of users are taken into account:

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



The user groupware in practice

1 Overview

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

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

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

- Viewing the price list,
- Modifying the price list,
- Entering orders,
- Enter customers.

The accesses differ according to the user. Some examples:

- the administrative assistants can see the price

list and create orders.

- the sales people can see the price list, place orders and create new customers.
- the sales directors have access to all the 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 mouse clicks, a standard application can be changed into an application that manages several access levels. This feature can be easily implemented.

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

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

2 How does the user groupware operate?

A site using the user groupware has two use levels:

- the user level.
- the supervisor level.

2.1 The user level

The user connects to the application via a login page and he accesses the authorized features.

Note: The user groupware can allow you to connect by using an LDAP directory.

2.2 The supervisor level

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

The advanced menu allows the supervisor to:

- start the site.
- configure the groupware.

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

See "Configuring the site that is using the user groupware", page 132 for more details.

Note: The information regarding the users and their rights is stored in data files in HFSQL format.

3 Implementing the user groupware

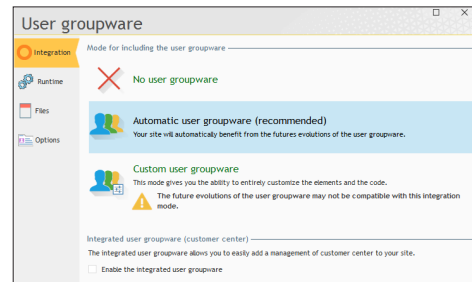
3.1 Adding the user groupware into a site

To implement the user groupware in a WEBDEV site, a single option is required: on the "Project" pane, in the "Project" group, click "User groupware". The window for configuring the user groupware is used to define:

- The management mode of user groupware.
- The mode for starting the user groupware.
- The location of data files of user groupware.

Mode for integrating the User Groupware

Several modes can be used to integrate the user groupware:

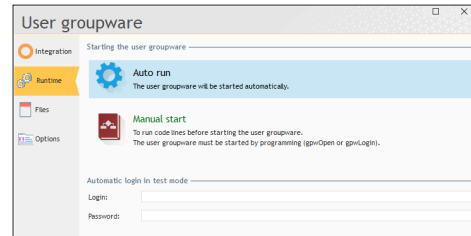


- **Automatic user groupware:** the entire programming of user groupware is automatically included in your site. No customization is allowed. The skin template of project is automatically applied to all pages used by the user groupware. This integration mode is used to automatically follow the possible evolutions of user groupware.
- **Custom user groupware:** the entire programming of user groupware is automatically included in your application via an internal component. **All the pages required to manage the user groupware (login and user management) are automatically added to your project.** This allows you to customize all the pages used by the user groupware.

The different pages found in the User Groupware are presented in details in the online help.

Starting the User Groupware

Two modes can be used to start the user groupware:



- **Auto run:** The login window of user groupware is started before any other application page. The initialization code of project is run once the login page is opened and validated.
- **Manual start:** The login page will be opened only if `gpwOpen` is used. This option is used to run the initialization code of project before opening the login page. This allows you to display a page asking for the runtime language of site for example.

Automatic login in test mode

You can specify the login and password to use in "automated test" mode. This information will be used if an automatic test is run on the application.

Data files of User Groupware

The user groupware is using several specific data files. These data files are mainly used to manage the users, the groups, the rights and the elements of the site.

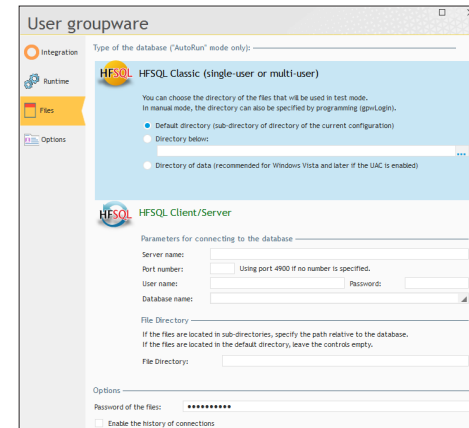
The user groupware can be used:

- with HFSQL Classic data files. In this case, the "Files" tab is used to specify (if necessary) the location of the data files.
- with HFSQL Client/Server data files. In this case, the "Files" tab is used to define the parameters for accessing the HFSQL Client/Server database as well as the file directory.

The password of data files

By default, the data files found in the user groupware have a specific password. This password is: "PCSGPW2001".

To change this password, type the new password in the "Files" tab of the description window of user groupware. The characters typed are displayed as stars.



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

See the online help for more details.

Identification

The user groupware proposes several modes for identifying users:

- Identification managed by the user groupware of WINDEV.
- Identification from an LDAP directory.

Identification managed by the user groupware

The login and password are saved in the data files of user groupware.

Identification from an LDAP directory

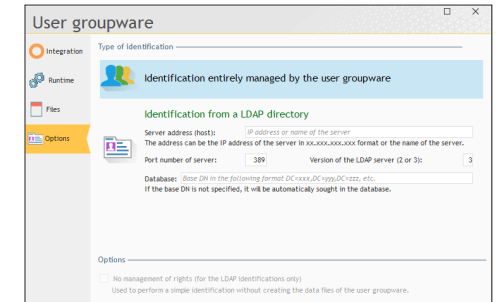
If the company of your customer is using an LDAP directory, you have the ability to branch the user groupware onto this directory. To do so, specify the characteristics of LDAP directory (server, port, database, ...) in the "Identification" tab, .). Therefore, the account and the password found in the LDAP directory will be automatically asked to the user.

Two modes are available:

- **Without management of rights:** No groupware file will be created. In automatic run mode, the login page will ask the user to identify himself. If the user is saved in the LDAP directory, the site will start ; otherwise, it will be closed. A supervisor cannot configure rights on the pages.
- **With management of rights:** Only the users found in the LDAP directory will be able to connect. The rights can be configured on the windows.

See the online help for more details.

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



3.2 Running the site test

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

A single user exists by default: the supervisor. To connect as supervisor, all you have to do is type the following information in the login page:

- Name: SUPERVISOR
- Password: SUPERVISOR

You can now run the test of your site or configure the user groupware.

Notes:

- In order for the first page of your site not to be the login page, check "Manual run" in the "Execution" tab of the options of User Groupware. All you have to do is call `gpwOpen` to display the login page.
- The pages for managing the user groupware are run before the initialization code of the project.

- The first page of your site defined in the project will be started after the login page (when the user is not the supervisor).
- **To avoid starting the user groupware when running the application tests**, select "No user groupware" in the "Integration" tab of the options of user groupware.
If the user groupware is re-enabled later, the data files previously used by the user groupware will not be erased.

3.3 Installing a site that is using the user groupware

The method for installing a site that is using the user groupware is identical to the method for installing any other site: on the "Project" pane, in the "Generation" group, click "Deploy the site".

4 Configuring the site that is using the user groupware

When the site is deployed, the user groupware is configured by the supervisor of the application.

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

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, you must:

1. Start the site and connect as supervisor:
 - Name: SUPERVISOR
 - Password: SUPERVISOR
2. Select "Configure the groupware".
3. The page for managing the user groupware is displayed. This page is used to:
 - manage the users and groups.
 - manage the rights.
 - display the statistics.
 - migrate the data coming from a user groupware version 18 and earlier.

A screen specific to the user groupware is used to configure the location of data files specific to the user groupware.

Notes:

- If the data files of user groupware 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 files have not been configured, only the Supervisor user will exist during the first application start.

Tip: if you do not configure the different use levels of user groupware for the end users, we recommend that you provide them with a document listing all the controls found in your pages to get an optimized configuration.

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 last name of user (mandatory)
- the first name of user
- the user login. This login corresponds to the identifier of user when he connects to the application.
- the user password. This password is optional and it can be typed by the user during his first connection (check the corresponding option).

The user can be defined as being a site supervisor. You also have the ability to modify or delete a user. When deleting a user, you have the ability to delete the entire user or to delete his rights only.

The users can be organized in groups.

A user can be associated with several groups.

Notes:

- The supervisor password should be changed during the first use.
- If you are using an LDAP directory, you have the ability to import the users found in the directory in order to manage the rights of these users.

4.2 Managing the rights

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

For each association between a user and a page, a specific status can be defined for all the page elements.

The elements managed in the pages, internal pages and page templates are as follows:

- the controls, the groups of controls,
- the menu options.

5 Tips for a site that is managing the user groupware

5.1 Using groups of controls

To simplify the configuration of groupware management according to users, we recommend that you use groups of controls.

In your pages, you have the ability to create groups of controls according to the controls that will be displayed for a specific type of user.

The ability to associate a control with several groups of controls increases the number of possible combinations.

These groups of controls can be created in your site only for managing the groupware, no specific programming is required.

5.2 Visibility of controls

When developing your application, you can define the visibility options for the elements found in your pages:

- when describing the element (7-tab window),
- by programming (`..State` or `..Visible` property).

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 displayed but it is grayed. No input can be performed.
- **Invisible:** the element is not displayed.

4.3 Displaying the statistics

For each site, you have the ability to get use statistics according to the user, for a specific period.

When configuring the user groupware, the supervisor can define different visibility characteristics. The characteristics defined by the supervisor have priority.

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

5.3 Defining rights

To get a definition of rights corresponding to the features of your site, we advise you to:

- configure the rights for your application before creating the setup procedure. Then, all you have to do is add the groupware files when creating the site setup.
- supply a programming documentation that provides the names of controls, groups of controls and options that must be configured according to the level of use defined for the site.

Multilingual sites

WEBDEV allows you to easily create multilingual sites from the same project.



Multilingual sites

1 Overview

A multilingual site can be distributed in several languages. The different languages of the site will be taken into account during the different development steps.

The main steps for developing a multilingual site are as follows:

1. Choosing the languages supported by the project and the analysis.
2. Entering the different project elements (pages, code, ...) in the different project languages.
3. Defining the project language by programming.
4. Managing specific character sets in the data files.
5. Creating the library and the setup program.

Notes:

- If the operating system of current computer supports several languages (Hebrew, Arabic, Greek, ...), the corresponding character set will be automatically used when the translations are typed in these languages.
- If your site is a multilingual site, this feature must also be managed in the pages of user groupware and in the pages for automatic management of HFSQL errors. See the online help for more details.
- The management of Unicode is available in the HFSQL data files and in the controls of pages.

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" pane, in the "Project" group, click "Description". The description window of the project is displayed.
2. In the "Languages" tab, you have the ability to add or delete the languages supported by the project.

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

When the project languages are modified, the modifications will be automatically taken into account by:

- for each new element or object created in the WEBDEV editor,
- all each element or object opened in the WEBDEV editor.

The main language corresponds to the language used by default at run time.

Linguistic options

The "Languages" tab of project description is also used to define the options specific to a language: number, currency, date, ...

The parameters used by default are the ones defined in the linguistic options of Windows (accessible from the control panel of Windows).

When creating an edit control or a table column that displays numeric data (number, currency, date, time, duration, ...), the input mask used will be the mask defined in the language options of the project. This option is available in the pages.

At run time, when an edit control or a table column has a mask "Defined by the project", the input mask and/or the display mask will automatically adapt according to the options selected in the project for the language displayed at run time.

Note: The linguistic options are also used to define the writing direction and the character set used ("Various" option).

2.2 Languages supported by the analysis

If your project is using an analysis, the choice of languages supported by the analysis is performed in the data model editor. Indeed, the same analysis can be shared among different projects that do not propose the same languages. Therefore, the

number of languages defined for the analysis can be greater than the one defined for the project.

To configure the languages supported by the analysis:

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

The different languages configured in the analysis will be proposed:

- when configuring the shared information of items. The description of the controls linked to items (options, captions, ...) can be typed in the different languages supported by the analysis. When generating a "full application RAD" or a

page RAD, this information is automatically taken into account by all the languages common to the analysis and to the project.

- for the information printed in the program documentation (notes of the data file or item).

2.3 Languages supported by the different project elements

By default, the different project elements (pages, reports, code, classes, ...) support the same languages as the project where they have been created.

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 several 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, several 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 by programming.

3.1 Selected translation mode

WEBDEV proposes several translation modes:

- Automatic translation of interfaces via a specific translation tool, found on the development computer. This translation is performed in the WEBDEV editor.
- Check out all the messages to translate via WDMMSG and check them back in once they are translated.

Translating interfaces

For each object, several multilingual areas are displayed in the description window. The multilingual areas allow you to enter the information in the different languages supported by the project.

To translate this information from WEBDEV:

1. Display the WEBDEV options for configuring the translation options of software used: on the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV".

2. In the "Translation" tab, choose:

- the default translation tool (as well as the transmission mode for the text to translate and for the translated text).
 - 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. To perform the translation, all you have to do is select the text to translate and click the "Translate" button.

Checking out and checking in the information to translate

WDMMSG is used to check out all project messages (captions of controls, ...) and check them back in once they are 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 by using these specific character sets.

WEBDEV allows you to automatically manage the use of specific character sets in the editor.

Indeed, as soon as the edit cursor of the mouse is located in an input area of a language that uses

a specific character set, the corresponding input language (character set used by the keyboard) is automatically displayed.

Therefore, if a caption is entered in the Russian section of the caption description, the keyboard will automatically switch to the Russian character set.

Reminder: To use specific character sets, the files corresponding to the requested character sets must be installed in the regional options of Windows (control panel).

3.3 Translating specific pages

Managing HFSQL errors

By default, the pages for managing the HFSQL errors are supplied in English and in French. To translate them into another language, you must:

1. Include the default error pages in your project. These pages are supplied as example in \Programs\Datas\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 supplied in English and in French.

To translate a site that uses the user groupware, you must choose the "Custom" mode when the user groupware is included in your site. Reminder: To configure the user groupware, on the "Project" pane, in the "Project" group, click "User groupware". The different pages of user groupware will be

included in your project.

Then, all you have to do is translate these pages.

Note: Uncheck "Automatic run" if necessary: this allows you to display a page for language selection before starting the login page.

3.4 Translating messages found in the WLanguage code

In your code, several WLanguage functions allow you to communicate with users via character strings. These messages must also be translated into the different languages of your site.

To translate a character string entered in the code editor:

1. Select the character string to translate.
2. Select "Translate" from the popup menu of the selection. The window for translating messages is displayed.
3. Type the translation and validate. The code line is displayed in the code editor:



The flag indicates that translations exist for this character string. Click this flag to display the translation window.

The number (2 in our example) indicates the number of translations typed for this character string.

Note: To transform all the messages found in your code into multilingual messages, on the "Code" pane, in the "Languages" group, expand "Translate the strings" and select "Convert the simple strings into multilingual messages".

4 Choosing the language by programming

A multilingual site is an application that can be distributed in several languages. The user will be able to choose the runtime language of application. You can for example:

- ask for the runtime language during the first

application start

- implement an option (menu option or button for example) allowing the user to change the current application language.

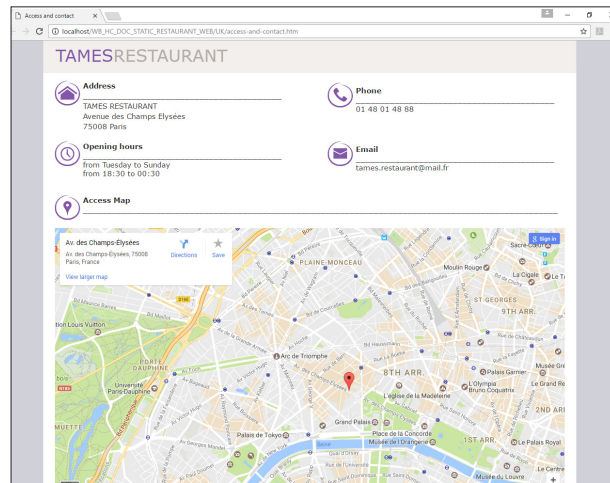
10 tips for better interfaces ...

1 Enhance the style of home page.

The home page is a major element of your site. In some sites, it is just a connection form but in most Internet sites, this is truly the main page for attracting visitors (and future customers). Its content and its presentation must be studied with great care. Don't hesitate to read books about user-friendliness of sites to help you.

2 Get in touch

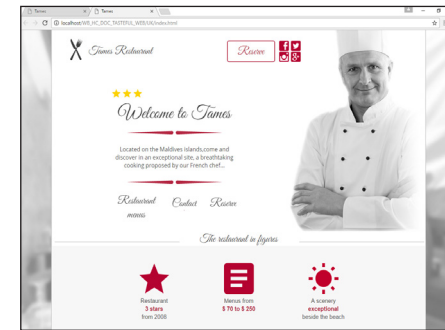
When a Web user visits a site, he must be able to identify the person or the company in charge of the site via a choice on the page: all you have to do is specify a name, a postal address and an Internet address, ... allowing the Web user to contact you.



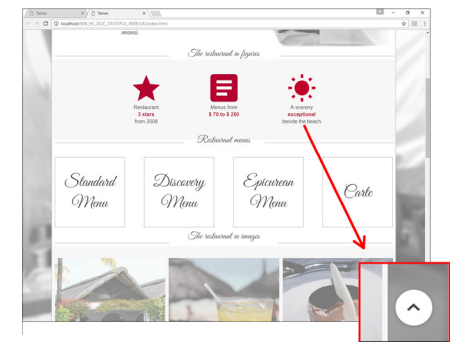
This page can display a map used to specify the store location, the opening times, ...

3 Your pages are very long? Give the Web user the ability to go back to the top of page.

If your pages are very long (more than 2 screens), use buttons or links to go back to the top of page (to avoid using the scrollbar).



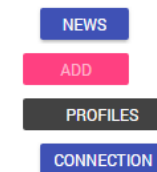
Top of page



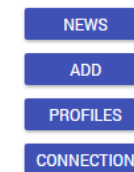
End of page with a button used to go back to the top of page

4 Standardize the style and alignment of buttons.

Check whether your buttons are using the same style and check their alignment. The page editor includes a snap-on and alignment mechanism to simplify this task.



Buttons are not aligned, using different styles



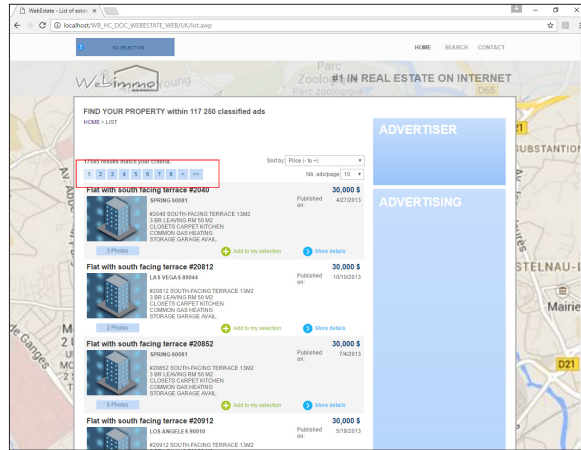
Buttons are aligned and standardized

5 Avoid using framesets.

The framesets, even though they seem to be more efficient at first, have specific problems: they include several pages. Referencing them beside search engines is more complicated. The navigation (especially the use of browser "Back" key) is hard to understand for the users. Today, it is not recommended to use framesets when developing sites.

6 You display lists containing more than 20 elements? Use several pages to list the elements.

The pager control will allow you to go from a page to another one without having to write a single code line and the overall interface of your site will be lighter. The site will be more pleasing and faster to view.



7 Use images displayed at 100% or Homothetic centered.

To keep the proportions of your images, we recommend that you choose:

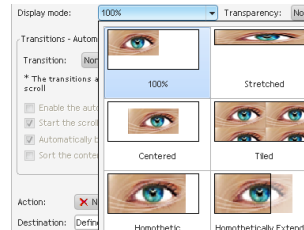
- images displayed at 100% (for the static and dynamic images or if the image size is identical to the control size).
- homothetic centered images (for the generated images or for the images of different sizes coming from a database): the image size will be adapted by homothety to the area defined for the image. The proportions will be respected.



Image at 100 % if the control dimension corresponds to the image dimension, otherwise homothetic image



Stretched image (not recommended)



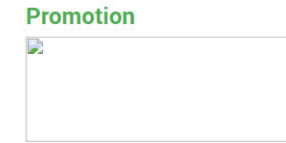
Selecting the display mode in the editor (image description, "General" tab)

8 You are using images? Check whether they are displayed properly in your pages.



%Title
%Price
%Instead of

Image in the editor



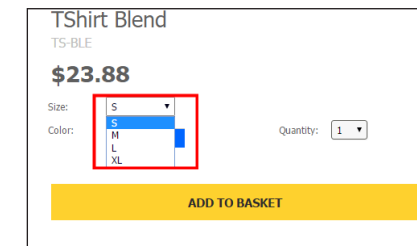
Smart&Modern garden lounge
1249,00 €
1699,00€

Image in the browser. Problem while displaying the images

If a display problem occurs, check whether:

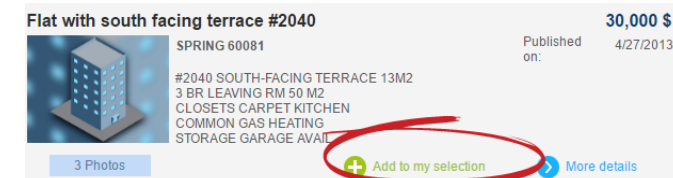
- the image exists,
- the image is saved in a format recognized by the browser (GIF or JPG for example),
- the image is found in the sub-directory of <ProjectName>_WEB project.

9 The Web user must type a quantity? Use a combo box to simplify the input of quantities.



10 You are proposing a product catalog with ability to order?

Specify all the details about the product (name, reference, price, ...), including the button or the link used to order the product, to add it to the selection, to add it to the basket, ...



Overlaying the controls

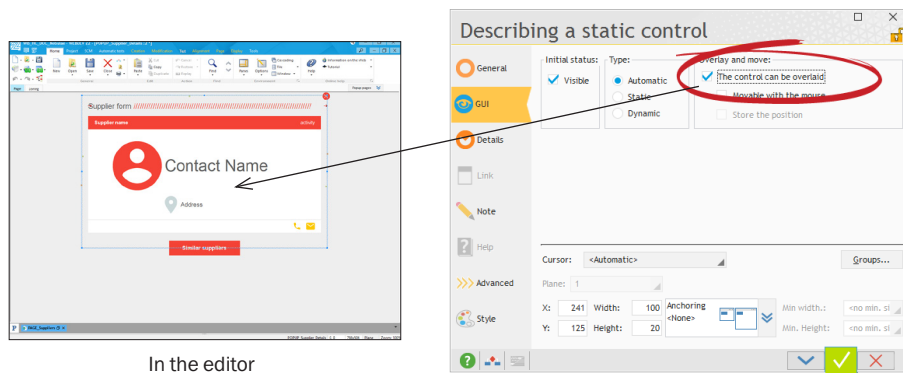
To get special effects (transparency or depth), your controls can be overlaid.

Each control includes an option "The control can be overlaid" ("GUI" tab in the description window of control).

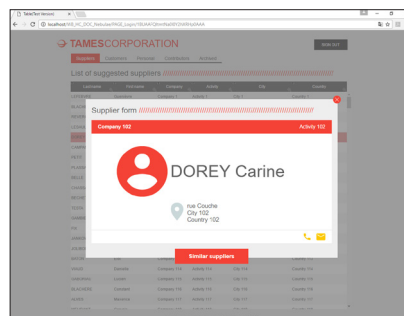
Check this option and the control is no longer linked to the positioning tables found in the page. The control can be moved and placed "above" each control found in your page.

Caution: Any control including the option "The control can be overlaid" will be displayed above the other controls. If several controls have this option enabled, the order in which they are stacked can be defined via the "Up", "Down", "Background" and "Foreground" options of "Control" menu.

For example, in order for a static to be displayed on an image, the static must be made stackable and it must be positioned on the image.



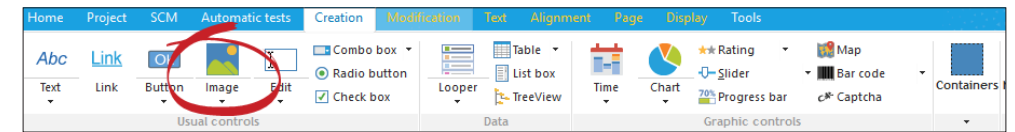
In the editor



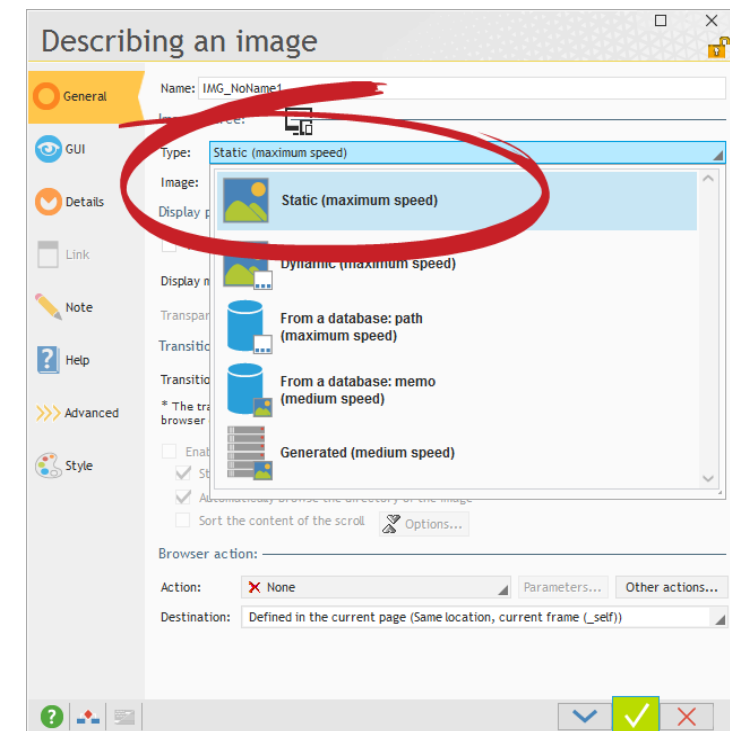
In the browser

Type of image: dynamic, static, generated, ... ?

The images are often used to illustrate the site pages.



Different types of images can be used in WEBDEV. Each type corresponds to a specific use of image.



Summary of characteristics of these images and how they should be used.



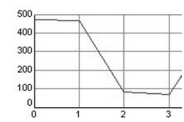
Tip: always prefer a homethetic display for the images coming from a database.

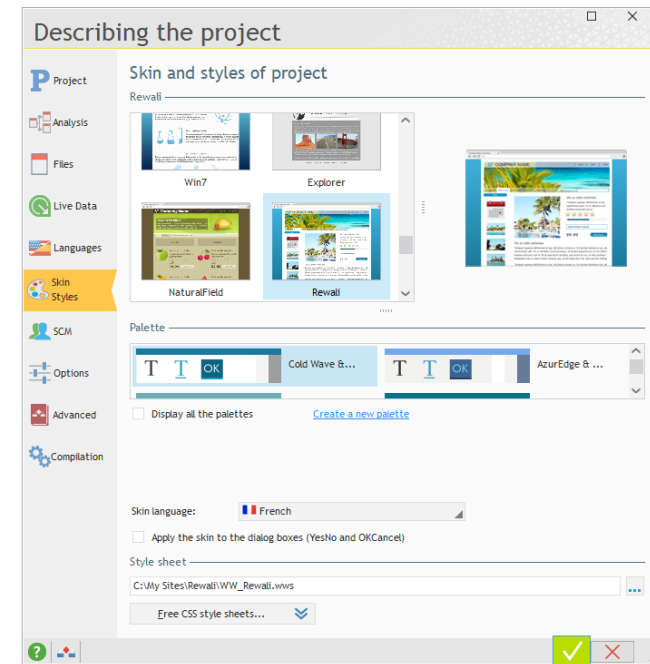
Customizing the aspect of a site: skin elements

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 description window of project (on the "Project" pane, in the "Project" group, click "Description"). The skins are configured in the "Skin, Styles" tab. Several skins are supplied with WEBDEV.

Type of image	Characteristics
<p>Static</p>  <p>Image not modified when using the site (company logo, ...) Image modified in browser code only (image selected by an Upload control).</p>	<p>The image associated with a static image must be selected in the WEBDEV editor. This image can be changed by programming in browser code only. The image formats recognized by WEBDEV are allowed. When saving the page, the image is automatically converted to GIF, JPEG or PNG format and it is saved in the <ProjectName>_WEB directory (except if the image is already in GIF or JPEG format). When deploying the WEBDEV site:</p> <ul style="list-style-type: none"> the file associated with the image must be found on the server. if the image is changed in browser code, the new image must be accessible and in a format recognized by the browser.
<p>Dynamic</p>  <p>Image modified when using the site (image of a product in a form, photo in a phone book, ...) This is the most common type of image.</p>	<p>A dynamic image can be changed by programming in server code or in browser code. The image associated with the control by programming must be:</p> <ul style="list-style-type: none"> found in the <ProjectName>_WEB directory. in a format recognized by the browser (GIF, JPEG or other) <p>Caution: no binary memo can be used.</p>
<p>Generated</p>  <p>Image drawn when using the site: - charts (grXXX functions) - drawings (dXXX functions) A background image can be used.</p>	<p>A generated image is built while using the dynamic WEBDEV site. This type of image should not be used to display existing "image" files: the performance of your site might be reduced. When the browser asks to display the image (when displaying or refreshing the page for example), a temporary image of current drawing is built on the server and saved in JPG format. This image is displayed in the browser.</p>
<p>From a database</p> <p>Images of products displayed in a loop</p>	<p>These images are saved in binary memos (memo) or in text controls (path). We recommend that you use a centered homothetic image: all the displayed images will be proportionally resized in order to be displayed in the image control.</p>



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 must use a specific skin (the "Promotion" pages of a site for example), you have the ability to replace the project skin by redefining it for a page template. If a particular page must have a different skin, this skin can be chosen for the page itself.

The color palettes

The color palettes define a set of colors that are proposed in all color pickers of 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 must use a specific palette (the "Management" pages of a site for example), you have the ability to replace the project palette by redefining it for a page template. Finally, if a specific page must have a different palette, this palette can be defined for the page itself.

The preset page templates

The preset page templates provide structures of pages corresponding to a large range of sites and they simplify the re-usability as well as the graphic harmony between the projects.

Several templates are supplied with WEBDEV.

The same preset template can be added several times into a project so that it can be used with different skins or different color palettes for example.

Once it 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

Several types of button/link are available according to the operation that must be performed on the page controls. The type of button/link is defined in the control description ("General" tab).

Operation on the controls:

- Send the value of controls to the server (submit)
- Reinitialize the page controls (reset)
- None

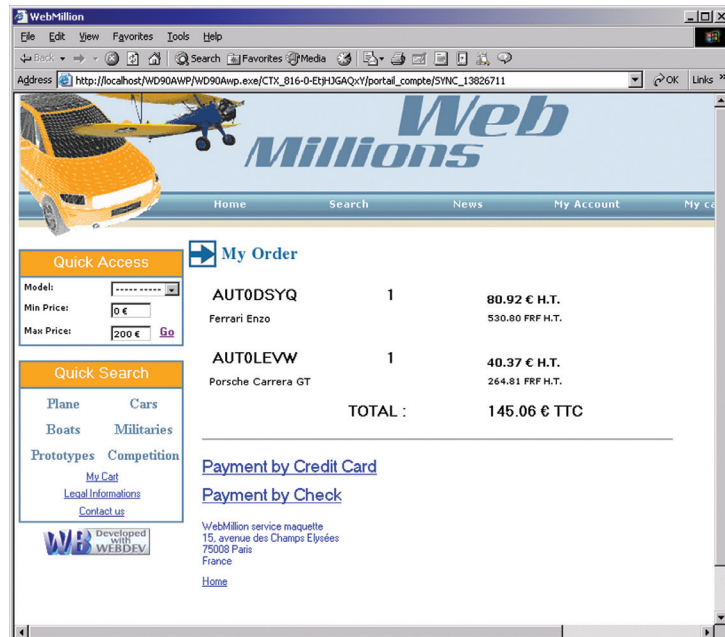
Let's see some examples for "standard" use of buttons/links in a page:

- **Send data from the current page to the server for processing**
Example: Registration form: the link is used to check the validity of information typed and to display a confirmation page.
To perform this type of process, use a "Send the value of controls to the server" link. The page data is sent to the server in order to be processed.
- **Perform a process on the server, without retrieving data from the current page**
Example: Menu page made of buttons: each button is used to start a site page.
To perform this type of process, use a "None" button.
- **Perform a process in browser code**
Example: Perform a simple calculation locally, an input check, ...
To perform this type of process, use a "None" button.
- **Display a page external to the WEBDEV site**
Example: Start a search engine site via a button.
To perform this type of process, use a "None" button.
- **Reinitialize the page controls**
Example: Link used to clear the controls found in the current page.
To perform this type of process, use a link such as "Re-initialize the page controls".


Preventing from going back to a page

The browser "Back" button allows the Web user to go back to a page that was already visited. However, it may not be a good idea to go back to a page.



For example, in a business site, if the Web user goes back to the page for order validation, he may validate his order twice and therefore place 2 orders.



To prevent from going back to a page via the browser "Back" button:

1. Display the page in the page editor of WEBDEV.
2. On the "Page" pane, click the group icon of "Edit" group (). The description window of page is displayed.
3. Display the "Details" tab.
4. Check "Prevent from using the browser "Back" button to go back to this page".
5. Validate.

Security benefit: managing the "Back" button

A browser allows you to navigate between the different site pages via the "Next" and "Previous" buttons ( and ).

A page context is created on the server whenever a new dynamic page is displayed in the browser.

When the same dynamic page is used to display different data (page with browsing table, loopers or "Form with browse" page), the page context found on the server changes according to the data displayed.

When the Web user clicks the "Back" button, the browser does not inform the site or the server. The browser goes back to the previous page without the corresponding context.

No specific management is required when a "Back" operation is performed between different pages.

A specific management is required when the same dynamic page displays different data.

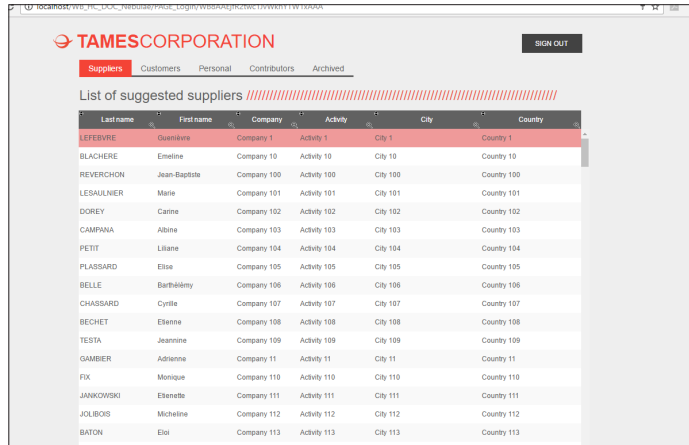
Why manage the "Back" button? A simple example ...

To understand why the "Back" button must be managed, let's take a look at a simple example: a dynamic page that is used to display the list of suppliers.

Step 1: Displaying the list of suppliers

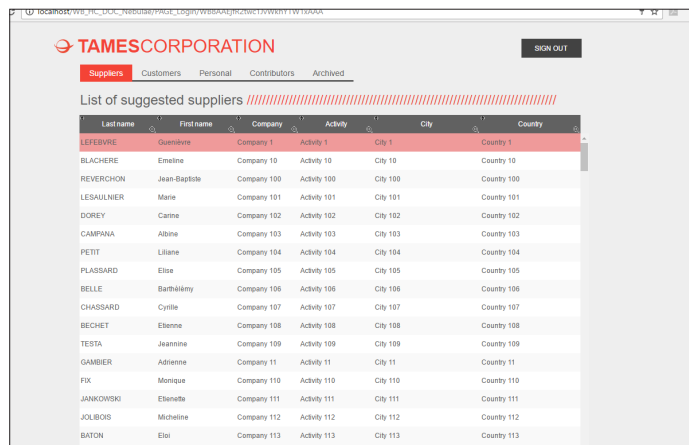
Last name	First name	Company	Activity	City	Country
LEFEBVRE	Guillaume	Company 1	Activity 1	City 1	Country 1
BLANCHERE	Estelle	Company 10	Activity 10	City 10	Country 10
REVERCHON	Jean-Baptiste	Company 100	Activity 100	City 100	Country 100
LESAULNIER	Marie	Company 101	Activity 101	City 101	Country 101
DOREY	Celine	Company 102	Activity 102	City 102	Country 102
CAMPANA	Albine	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	Barthélemy	Company 106	Activity 106	City 106	Country 106
CHASSARD	Cyrille	Company 107	Activity 107	City 107	Country 107
BECHET	Esienne	Company 108	Activity 108	City 108	Country 108
TESTA	Jeannine	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	Eliette	Company 111	Activity 111	City 111	Country 111
JOLBOS	Micheline	Company 112	Activity 112	City 112	Country 112
BATON	Eloi	Company 113	Activity 113	City 113	Country 113

The context contains the list of suppliers displayed in this table.

Step 2: Clicking the "Customers" link.


Last name	First name	Company	Activity	City	Country
LEFEVRE	Guillaume	Company 1	Activity 1	City 1	Country 1
BLACHERE	Emeline	Company 10	Activity 10	City 10	Country 10
REVERCHON	Jean-Baptiste	Company 100	Activity 100	City 100	Country 100
LESAILNIER	Marie	Company 101	Activity 101	City 101	Country 101
DOREY	Carine	Company 102	Activity 102	City 102	Country 102
CAMPANA	Albine	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	Barthélémy	Company 106	Activity 106	City 106	Country 106
CHASSARD	Cyrille	Company 107	Activity 107	City 107	Country 107
BECHET	Eloane	Company 108	Activity 108	City 108	Country 108
TESTA	Jeanne	Company 109	Activity 109	City 109	Country 109
GAMBER	Adrienne	Company 11	Activity 11	City 11	Country 11
FIX	Monique	Company 110	Activity 110	City 110	Country 110
JANKOWSKI	Eloane	Company 111	Activity 111	City 111	Country 111
JOLBOIS	Micheline	Company 112	Activity 112	City 112	Country 112
BATON	Eloi	Company 113	Activity 113	City 113	Country 113

The customers are displayed. The page context is updated on the server: the context contains the list of elements displayed in this table (list of customers).

Step 3: Clicking the browser "Back" button.


Last name	First name	Company	Activity	City	Country
LEFEVRE	Guillaume	Company 1	Activity 1	City 1	Country 1
BLACHERE	Emeline	Company 10	Activity 10	City 10	Country 10
REVERCHON	Jean-Baptiste	Company 100	Activity 100	City 100	Country 100
LESAILNIER	Marie	Company 101	Activity 101	City 101	Country 101
DOREY	Carine	Company 102	Activity 102	City 102	Country 102
CAMPANA	Albine	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	Barthélémy	Company 106	Activity 106	City 106	Country 106
CHASSARD	Cyrille	Company 107	Activity 107	City 107	Country 107
BECHET	Eloane	Company 108	Activity 108	City 108	Country 108
TESTA	Jeanne	Company 109	Activity 109	City 109	Country 109
GAMBER	Adrienne	Company 11	Activity 11	City 11	Country 11
FIX	Monique	Company 110	Activity 110	City 110	Country 110
JANKOWSKI	Eloane	Company 111	Activity 111	City 111	Country 111
JOLBOIS	Micheline	Company 112	Activity 112	City 112	Country 112
BATON	Eloi	Company 113	Activity 113	City 113	Country 113

The dynamic 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 Web user. The page context on the server is still the one corresponding to the customers. If the Web user selects an element (a supplier) in the table, 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 the reason why the Back button must be managed.

How to manage the "Back" button?

To manage the "Back" button, you can use:

- the **synchronization code of dynamic page**. 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 **"Disable the mechanism for page synchronization for this control"** on your buttons and/or links ("Advanced" tab of button description).

You can choose and combine one or another of these options according to the type of action that can be performed in the dynamic page.

Detailed example of Back management

A dynamic page is used to browse the different products of a site. The buttons found in the page are as follows:

- **Back to main menu**
This action requires no specific management of "Back" button. This button does not handle the data files: a desynchronization between the context and the page displayed has no effect.
- **Delete the current product**
The action "Delete the current product" handles a data file. The deleted record must necessarily correspond to the record viewed by the Web user. In this case, to manage the "Back" button, we advise you to 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 Web user). Re-read the record corresponding to the identifier stored in the hidden control. If the record is not found: an error page is displayed and the code of "Delete" button is not run. 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, 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 desynchronization problem between the pages displayed on the browser and the corresponding contexts found on the server, WEBDEV proposes two modes for managing the browser "Back" button:

- **Solution 1:** Prevent from going back to this page with the browser "Back" button. If the browser "Back" button is used to display the previous page, this action will have no effect.
- **Solution 2:** Manage 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 are available for managing 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 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 table rows, the "Next" link is used to display the next 6 rows.

When opening the page, the table displays the 6 first file records (from 'A' to 'F'). Let's see the sequence of actions performed by the user:

1. Clicking the "Next" link

Result: the server is positioned on the next 6 records of ITEM and their contents are returned to the browser. The browser displays the next table page with the 6 new contents ('G' to 'L').

2. Clicking the browser "Back" key

Result: the browser displays the page preceding the first action. The table displayed contains the letters 'A' to 'F'. The server was not contacted, therefore it is still positioned on the records 'G' to 'L'.

3. Clicking "Next"

Result: the server is positioned on the next 6 records of ITEM (from 'M' to 'R'). The browser is synchronized with the server and the same elements are displayed: 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 performed on the browser must trigger a server action: then, 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 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 pressing the browser "Back" key.

Notes:

- Clicking the browser "Back" button can make the page blink.
- This mechanism can fail if the [STOP] button of browser is clicked before the forward() statement is run by the browser.

2.2 Implementation

To disable the browser "Back" button for a specific page:

1. Display the description window of page: on the "Page" pane, click the group icon  of "Edit" group.
2. In the "Details" tab, check "Prevent from using the browser "Back" button to go back to this page".
3. Validate.

To disable the browser "Back" button for a frameset:

1. Display the "Details" tab in the description window of frameset ("Description" from the popup menu of frame).
2. Check "Prevent from using the browser "Back" button to go back to this frameset".
3. Validate.

To disable the browser "Back" button for all project pages:

1. Display the project description: on the "Project" pane, in the "Project" group, click "Description".
2. Click the "Options" tab.
3. Check "Prevent from using the browser "Back" button to go back to this page".
4. Validate. This option will be automatically taken into account for all new site pages.

3 Managing the 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 are available for managing the synchronization:

1. Default management of synchronization.
2. Management of synchronization by programming, in the synchronization code of the page.

3.2 Default synchronization

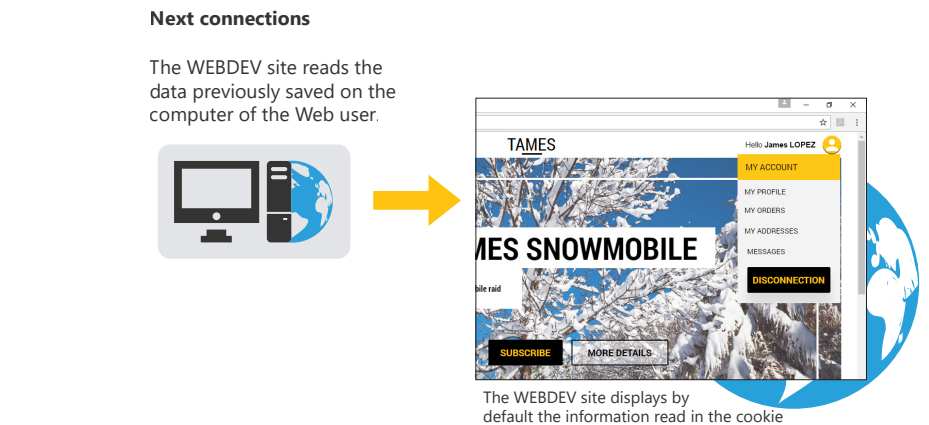
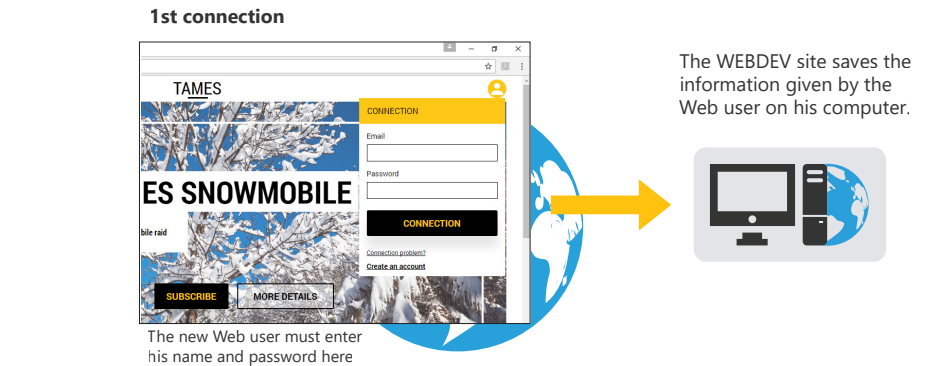
The synchronization mechanism is triggered only if the option "Use the mechanism for synchronizing pages" is selected for the page.

If a desynchronization occurs, a warning message informs the Web user that the requested action has not been performed. The page corresponding to the context found on the server is redisplayed. The site can continue to operate.

Cookies: information stored on the computer of Web user


A cookie is a file saved by the WEBDEV site on the computer of Web user. This file can be read at any time from the server or by the browser. The cookies are used to store persistent information on the computer of Web user.

For example, various information is requested when a Web user connects to a site for the first time: user name, login, ... and it is stored on his computer. During the next connections, the WEBDEV site will read this information in the cookie created during the first visit and it will be able to greet the Web user personally on the first page.



To create cookies in a WEBDEV site, use **CookieWrite**.
To read the cookies in a WEBDEV site, use **CookieRead**.

To implement the management of synchronization in a page:


1. Display the description window of page: on the "Page" pane, click the group icon  of "Edit" group.
2. In the "Details" tab, check "Use the mechanism for synchronizing pages".
3. Validate. This page will be automatically included in the history of browser pages: to go back to this page, use the "Back" key of the browser.

To implement the management of synchronization in all the project pages:

1. Display the project description: on the "Project" pane, in the "Project" group, click "Description".
2. In the the "Options" tab, check "Use the mechanism for synchronizing pages".
3. Validate. The project pages will be automatically

4 Synchronization by programming

To manage the synchronization by programming:

1. Display the description window of page: on the "Page" pane, click the group icon  of "Edit" group.
2. In the "Details" tab, check "Use the mechanism for synchronizing pages".
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 management of synchronization, check "Disable the mechanism for page synchronization for this control" in the "Advanced" tab of control description.

4. the the code required for custom management of synchronization in the synchronization code of page. Use **ChangeAction** in the synchronization code of page. This function is used to define the action that will be performed in case of page desynchronization.

included in the history of browser pages: to go back to these pages, use the browser "Back" key.

Notes:

- This management mode requires no specific WLanguage code.
- The synchronization mechanism can be disabled for the page controls that do not require a management of synchronization ("Close" button for example): all you have to do is check "Disable the mechanism for page synchronization for this control" in the "Advanced" tab of control description.
- The warning message can be customized (see next paragraph).

Notes:

- **ChangeAction** is initialized with "No action" if a WLanguage function used to display or refresh a page is used in the synchronization code of page.
- To customize the desynchronization message, all you have to do is type in the synchronization code of page:
 1. the custom message.
 2. the code for refreshing the current page on the server (with **PageRefresh** for example).
- To perform a synchronization from the information found on the computer of the Web user, we advise you to:
 1. Use a hidden control containing the identifier of displayed and selected record.
 2. In the synchronization code, find the current record on the browser. This search is performed from its identifier found in the hidden control.
 3. Refresh the page.

Protecting the access to the site: passwords

Managing the access to a site by password is required in two cases:

1. Protect the personal details
2. Restrict the access to a site

Protect the personal details

To access his personal details, the Web user can specify a password. In a business site, a password defined by the Web user allows him to access the details of his previous orders.

In this case, the management of site access requires:

- an identification: During the first connection, the Web user identifies himself and types his password.
During the connections, the user identifies himself and he accesses the features reserved to him (and to his history, ...).
- the ability for a Web user to retrieve a forgotten password.

WEBDEV is supplied with all resources required to manage the personal passwords.

Restrict the access to a site

By default, an Internet site can be accessed by any Web user. But a site is not always intended for the general public: some features can be reserved to some types of Web users (site administrator, ...).

In this case, the site manager assigns a password to each Web user. This password cannot be modified. This password gives access to specific features.

For example, in a business site:

- a specific password will be assigned to salespeople: they will be able to access the sales statistics, ...
- a specific password will be assigned to the members of marketing team: they will be able to on-line new products.

WEBDEV allows you to easily manage these different types of access to a site.

TLS/SSL: Encrypting information on the Web

By default, the data exchanged between the computer of Web user and the Web server is not encrypted. The authenticity of server identity, the confidentiality and the integrity of exchanged data are not guaranteed.

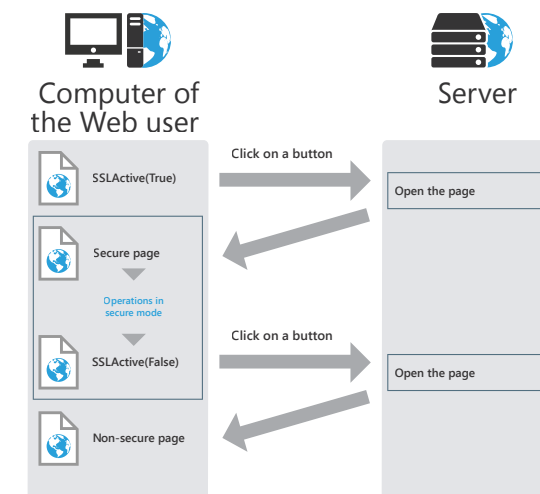
To insure confidentiality, the most used method is the TLS/SSL protocol (Transport Layer Security, formerly Secure Socket Layer). This protocol guarantees the identity of accessed server and it encrypts the data exchanged between the server and the browser.

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). See the online help of WEBDEV for more details.

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. Indeed, the secure mode is slower. However, an increasing number of sites entirely operate in TLS/SSL mode via the more powerful computers, insuring a better confidentiality for the user operations.



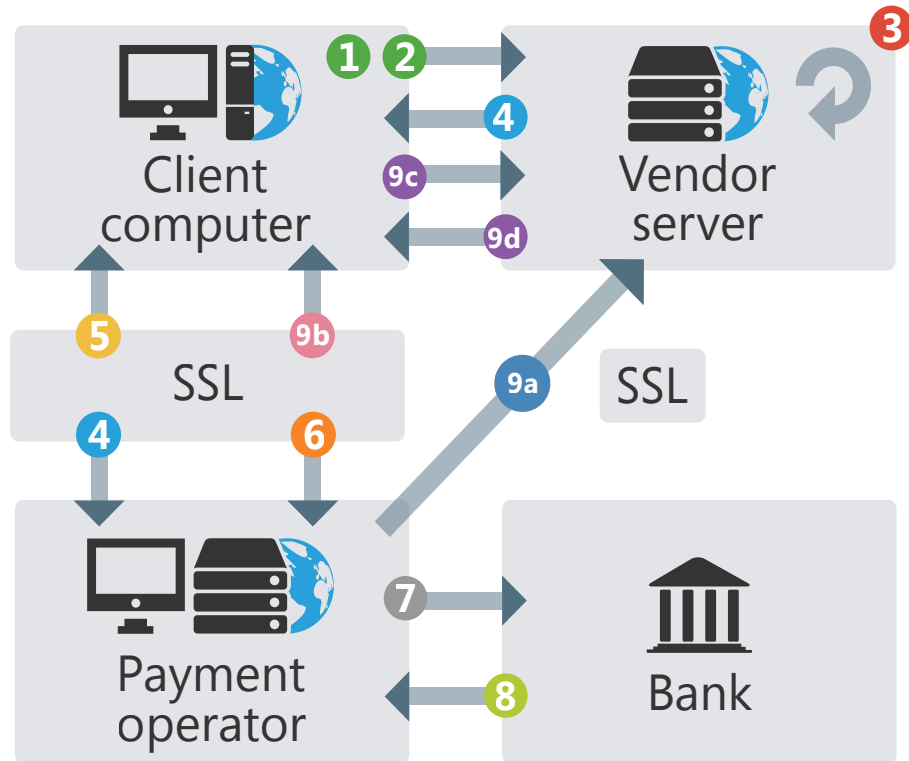
The secured mode is implemented when opening a new page via a button (or a link). To do so, use **SSLActive** in the browser code of button (or link). As soon as the secure page is opened, all the actions will be performed in secure mode (which means encrypted).

To go back to non-secure mode, all you have to do is use **SSLActive(False)** in the browser code of a button (or link) used to open a new page.

Secure payment with provider

Several providers allow you to implement systems for secure payment. The provider validates all payments by credit card beside 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 Web site, the emails allow:

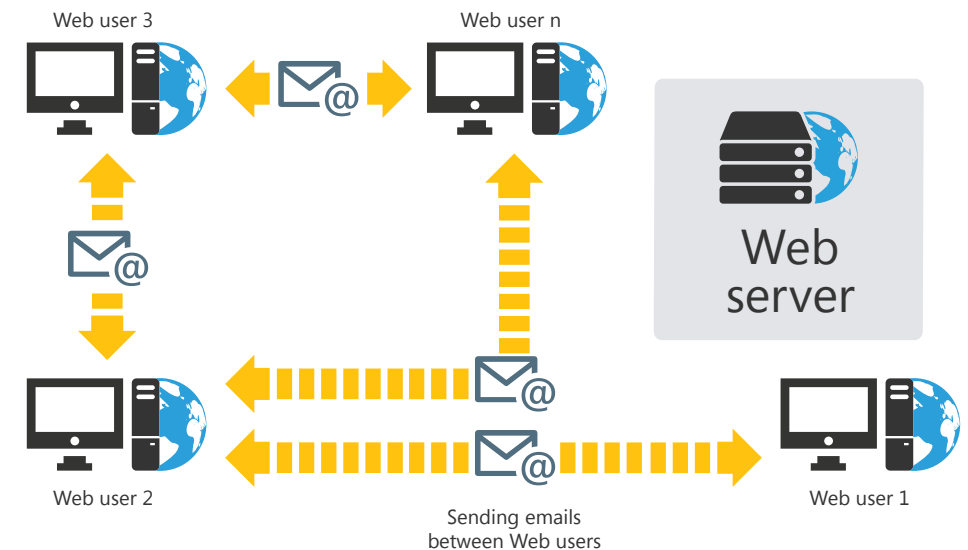
- the Web users to communicate between themselves: sending emails to the site manager, mailshot, ...
- to validate an order: in a business site, an email is sent to the Web users in order to validate their order.
- to transfer data from a computer to another one: the new orders entered in a business site are transferred by email to the company headquarters, ...

Two methods can be used to send emails from a site.

1 Sending the email from the computer of Web user

Use this method when the Web user must send an email to a specific address: author of site, technical support, sales department, ...

The *EmailOpenMail* browser function is used to open the email software of Web user. Some parameters can be filled by default: email address of recipient, ...



This method allows you to use the messaging software of Web user: no specific process must be performed in the WEBDEV site.

2 Sending the email from the server (dynamic pages only)

In this case, the server manages the emails.

The emails can be sent:

- in the dynamic WEBDEV site directly (to transfer data by email or to validate an order by email for example) ;
- by the email spooler (supplied with WEBDEV).



Use this method for a custom management of emails:

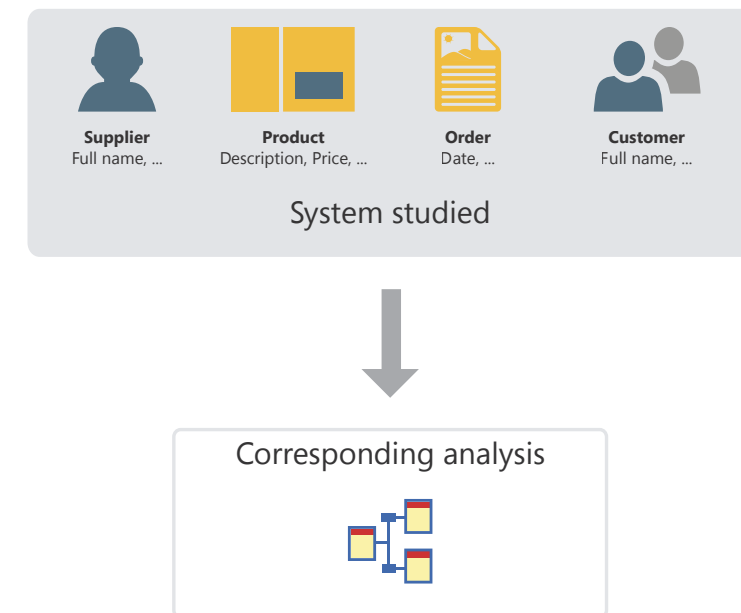
- checking the content of emails sent by the Web users ("mailing-list" site),
- checking the validity of recipient address,
- encrypting the information sent by email,
- validating a process, ...



Analysis: Database structure

When a WEBDEV, WINDEV or WINDEV Mobile project is using data files, this project must be associated with an analysis. An analysis allows you to describe the structures of data (files, items, ...) used in your project.

The data model editor is used to easily create an analysis.



The analysis of a WINDEV project corresponds to the LDM (Logical Data Model). The entire structure and data organization are described: the data is grouped by file. Each file contains several data called item.

In the analysis, the description of a data file can be linked to a type of file (HFSQL, Oracle, ...).

The analysis in practice

1 Overview

When a project is using data files, this project must be associated with an analysis. An analysis allows you to describe the structures of data (files, items, ...) used in your project.

Two methods can be used to describe an analysis:

- **1st method:** Direct creation of analysis (which

means Logical Data Model (LDM)).

- **2nd method:** Creating the Conceptual Data Model (CDM) then creating the analysis from CDM. See the online help for more details.


This chapter presents the first method.

2 Creating an LDM (Logical Data Model)

In WEBDEV, the terms "LDM" and "Analysis" are interchangeably used to define the structure of database associated with a project.

2.1 Creating an analysis (or LDM)

To create a LDM:

1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Data" then "Analysis". The wizard for analysis creation starts.

2. Specify:

- **the analysis name and directory.** The analysis corresponds to a ".WDA" file. By default, this file will be created in the directory of project analysis (<Project Name>.ANA directory). This directory must be accessible in read/write.
 - **the analysis caption** that briefly describes the purpose of analysis.
 - **whether the analysis must be associated with the current project.**
 - the type of database used by the project.
3. The creation of first data file is automatically proposed.
4. Create all the elements (data files, items and links) of your analysis.

2.2 Adding a data file into an analysis

The analysis is used to define the structure of data files used by the project. A data file found in the analysis can be:

- A new data file.
- A preset data file, supplied with WEBDEV.
- 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" pane, in the "Creation" group, click "New file". The wizard for creating data files starts.

2. Select "Create a new description of a data file".

3. Specify:

- **the name of data file.** This name is the logical name of data file. It will be used to handle the data file.
- **the caption of data file** that briefly summarizes the subject of data file.
- **the representation of a record** in the data file. This representation improves the legibility of questions asked when describing links. This option must be preceded by an indefinite article (A or AN).

- whether the data file includes an **"Automatic identifier" item**. The value of this item is unique for each record and it is automatically calculated by WEBDEV.

- **the type of database associated with the file.** Depending on the selected type, this data file will be handled by the HFSQL engine, by an OLE DB driver or by one of the native accesses (also called native connectors) of WEBDEV (SQL Server, Oracle, ...).

4. The created data file becomes the current data file. The description window of file items is automatically opened. It allows you to describe the file items.

To import a preset description of a data file:

1. On the "Analysis" pane, in the "Creation" group, click "New file". The wizard for creating data files starts.

2. Select the option "Select a description among the preset data files".

3. Choose the preset data file to create (you have the ability to select several ones). This data file will be imported into the current analysis. This data file can be modified later.

4. Select the items to store. These items can be modified later.

5. Specify whether the links must be automatically sought. If this option is checked, the items with the same name will be linked.

6. The imported data file is automatically inserted into the current analysis.

Note: To modify a data file or the items found in a data file:

1. Select the data file.
2. Select "Description of data files" or "Description of items" from the popup menu.

To import the existing description of a data file, two solutions are available:

Solution 1: From the data model editor

1. On the "Analysis" pane, in the "Creation" group, click "New file". The wizard for creating data files starts.

2. Select "Use the data files from an existing database" and select the type of database.

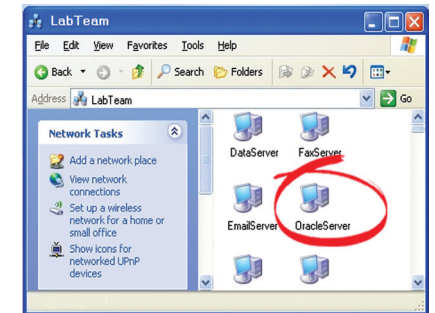
3. Specify the source database containing the descriptions to import and the type of this database. Depending on the selected type, specify the requested information.

4. Select the tables or the files whose description must be imported and validate.

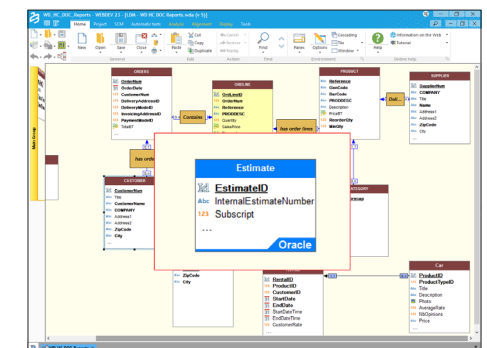
5. The imported data file is automatically inserted into the current analysis.

Solution 2: From the Windows explorer

Drop the file description from the Windows explorer to the data model editor. For example:



Here: an Oracle database on the server. All you have to do is drag the database name ...



... to transfer its description into 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 file items is opened.
2. Click the first empty row in the table of items.
3. Specify the name, the caption and the type of item in the table.
4. In the right section of the screen, specify the details about the new item (type, size, default value, sort direction, ...).
5. In the bottom section of the screen, specify the details about the shared information.
6. Validate the item description.

Note: you also have the ability to create an item from the meta-types proposed by WEBDEV. To do so, click the "+M" button on the right of table. The list of available meta-types is displayed.

2.4 Creating a link

Different types of links can be created between data files. See "Characteristics of links defined in an analysis", page 166 for more details.

To create a link:

1. On the "Analysis" pane, in the "Creation" group, click "New link". The mouse cursor turns into a pen.
2. Select the two data files to link. The window for link description is automatically opened.
3. To define the cardinalities:
 - select the cardinalities among the proposed ones (0,1 ; 1,1 ; 0,N ; 1,N).
 - answer the questions asked. The cardinalities will be automatically updated.
4. To describe the advanced cardinalities, check "Display the advanced cardinalities" and answer the questions asked.
5. Type the link caption by briefly describing the link purpose.
6. Specify the keys to link.
7. Define the integrity rules. These rules are used to ensure the data integrity when one of the relation keys is modified or deleted.
8. Validate. The link is automatically created.

3 Characteristics of links defined in an analysis

3.1 Owner file and member file

When a link is defined between two data files, an **owner** file and a **member** file are found:

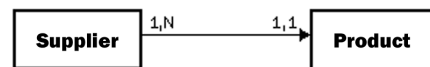
- the owner file is the key owner.
- the member file is a member of analysis files containing a copy of key.

To manage the link between two data files, the key of owner file is copied into the member file.

For example, the key of Supplier file is copied into each record of Product file. Several records of Product file can have the same key in Supplier file:

- the owner file is Supplier file,
- the member file is Product file.

This type of link is represented as follows:



3.2 The cardinalities

The cardinalities are used to count the links between data files.

The cardinality is defined according to the answers given to the two following questions:

1. For each file record, what is the minimum number of records in the other data file to which this record is linked?

The answer to this question provides the first part of cardinality (**minimum cardinality**):

- if the answer is "none", the cardinality is 0,X.
- if the answer is "a single one", the cardinality is 1,X.

2. For each record, in the data file, what is the maximum number of records in the other data file to which this record is linked to?

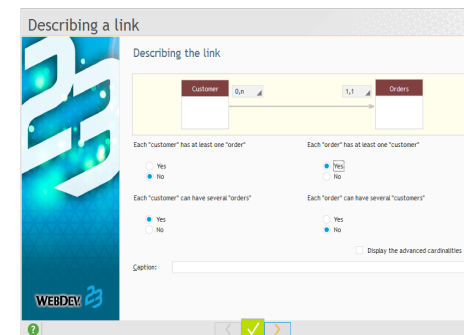
This answer provides the second part of the cardinality (**maximum cardinality**):

- if the answer is "a single one", the cardinality is X,1.
- if the answer is "several", the cardinality is X,N.

The answer to these two questions defines the cardinality that can be: 0,1 ; 0,N ; 1,1 ; 1,N

The description of cardinalities is fundamental: it is the basis for respecting the referential integrity of database.

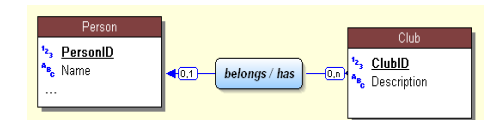
To be clearer, these two questions can be broken down into four questions. For example, to describe a link between Customer file and Orders file:



- each "customer" has at least one "order": Yes/No?
- each "customer" can have several "orders": Yes/No?
- each "order" has at least one "customer": Yes/No?
- each "order" can have several "customers": Yes/No?

Example of cardinalities:

This example presents two different cardinalities:



- **Cardinality 0,1:** A person can be a member of a single sport club. This person does not have to belong to a sport club.
- **Cardinality 0,N:** A club can have no member or several members.

3.3 The advanced cardinalities

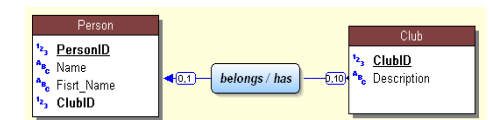
The cardinalities can be defined more precisely. We talk of **advanced cardinalities**.

These cardinalities are used to exactly specify the minimum cardinality and the maximum cardinality.

To define the advanced cardinalities, check "Display the advanced cardinalities" in the link description.

Example of advanced cardinalities:

This example presents two different cardinalities:



- **Cardinality 0,1:** A person can be a member of a single sport club. This person does not have to belong to a sport club.
- **Cardinality 0,10:** A club can have no member or up to 10 members.

3.4 The referential integrity

The referential integrity of a database corresponds to the respect of constraints implied by the links between data files.

The referential integrity consists in checking that:

- if a record is deleted from the owner file, the corresponding records are also deleted from the member files,
- if a record is added into a member file, a corresponding record exists in the owner file,
- if a record is modified in the owner file, the unique key is not modified, ...

The check of referential integrity depends on the nature of the link between the data files.

See the online help for more details.

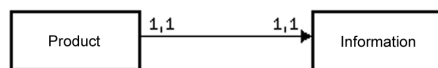
3.5 The different types of links

Several types of links can exist between the data files:

- parallel
- optional
- complement
- shared
- complex

Parallel link

For a parallel link, **each record** found in a data file (Product) is linked to **a record** found in another data file (Information), and **conversely**.



To manage a parallel link, the identifier of Product file is copied into the Information file. This identifier is also a unique key in the Information file.

The owner file is Product and the member file is Information.

Note: This type of link is rare because the two files can be grouped into a single file.

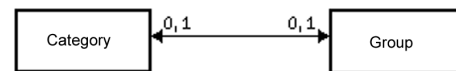
The parallelism of records in the file is respected if the following operations are simultaneously run on the two files:

- creating a record
- deleting a record
- reindexing with compression

Optional link

For an optional link:

- **Each record** of a data file (Category) **is associated** with no record or with a single record in another file (Group).
- **Each record** of the other data file (Group) **is associated** with no record or with a single record in the first data file (Category).

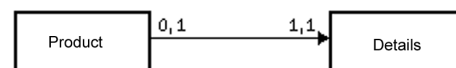


An optional link is performed by copying the identifier of each data file into the other data file.

Complement link

For a complement link:

- **Each record** of a data file (Product) **has no record or a single associated record** in another data file (Detail).
- **Each record** of the other data file (Details) **is necessarily associated with an associated record** in the first data file (Product).



The complement links are quite common. They are used when a record can have additional optional information.

To manage a complement link, the key of Product file is copied into Details file. To ensure the maximum cardinality of 1, it remains a unique key. This key's uniqueness forbids inserting more than one record in the Detail file for a record in the Product file.

The owner file is Product, the member file is Details.

Shared link

For a shared link, the same record in a data file (Supplier) can be shared by several records in another data file (Product).

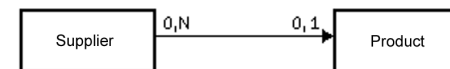
To manage a shared link, the key of Supplier file is copied into the Product file. It becomes a multiple key to increase the speed of integrity check.

The owner file is Supplier, the member file is Product.

Depending on the cardinality, we can distinguish between four types of shared links:

- Shared link with a 0,n - 0,1 cardinality
- Shared link with a 0,n - 1,1 cardinality
- Shared link with a 1,n - 0,1 cardinality
- Shared link with a 1,N - 1,1 cardinality

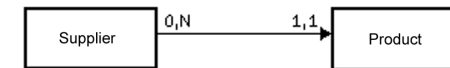
Shared link with a 0,N - 0,1 cardinality



With this type of link:

- an owner may have no member (a supplier does not necessarily have a product).
- a member may have no owner (a product does not necessarily have a supplier).

Shared link with a 0,N - 1,1 cardinality



With this type of link:

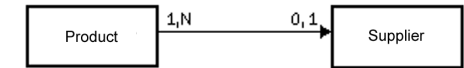
- an owner may have no member (a supplier may have no product).
- each member has a single owner (each product has a single supplier).

Note: This type of link is quite common. WEBDEV allows you to automatically create the pages used to manage the data files linked by a 0,N - 1,1 link.

Shared link with a 1,N - 0,1 cardinality

With this type of link:

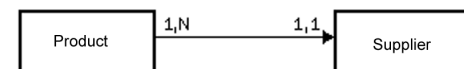
- each owner has at least one member (a supplier has at least one product).
- a member may have no owner (a product may have no supplier).



Shared link with a 1,N - 1,1 cardinality

With this type of link:

- each owner has at least one member (each supplier has at least one product).
- each member has a single owner (each product has a single supplier).



Complex link

For a "complex" link, you must manage a link file, called **relation file**.

The relation file will be automatically created. It will contain a unique key containing the two keys of the linked files.

The relation file can also contain information specific to the link.

A complex link includes two shared links.

Example of complex link

An order (Order file) can contain one or more products. A product (Product file) can be used in several orders.

In this case, a link file is required (OrderLine file).

The OrderLine file contains:

- a unique key containing the keys of Product and Order.
- the number of products ordered.

4 Operations that can be performed on an analysis

WEBDEV allows you to perform the following operations on an analysis:

- Duplicate/Copy 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 the display of LDM in the editor.
- Move the display of 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.

5 Generating the analysis (LDM)

The analysis generation is performed before the programming step and after the description of data files. This generation is used to:

- validate the modifications performed on the analysis (LDM).
- create the modules required for programming.
- automatically update the data files if necessary.

As long as the analysis (the LDM) has not been generated, the description of analysis (which means the data files) cannot be used in the project.

6 Managing the analysis versions

WEBDEV allows you to manage the different analysis versions:

1. On the "Analysis" pane, in the "Analysis" group, expand "Generation" and select "Manage the versions".
2. In the wizard, specify whether you want to work on the current analysis or on a specific analysis.

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.

See the online help for more details.

To start generating the analysis, on the "Analysis" pane, in the "Analysis" group, click "Generation".

The generation is performed in three steps:

- checking the modifications made and generating the data files of analysis description.
- automatic modification of accessible data files (files found in the "EXE" directory of project).
- Synchronizing the project.

Note: when an analysis is damaged, you have the ability to select the damaged analysis to restore one of the earlier versions for example.

All the analysis versions are viewed in a graph. The yellow rectangle indicates the version number of analysis ; the caption on the right indicates the generation date.

The "small rectangles" displayed between two versions have the following meaning:

- one rectangle: few modifications have been made between the two versions.
- several rectangles: several modifications have been made between the two versions.

To see the details of modifications performed between two versions: Double-click the line containing the "small rectangles" or click the [Modifications] button.

The options available in the version manager are:

- Restoring an analysis.
- Canceling the last generation.
- Resetting the version number to 1.

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 into the specified directory. An independent analysis corresponding to the selected version is created. This option is selected by default.
 - overwrite the current analysis.
3. Validate. The restore operation is performed.

Caution: The data files corresponding to an analysis whose version number is greater than the restored version cannot be opened anymore. In this case, you must also restore the files corresponding to the version of the restored analysis or delete the existing files in order to re-create them.

6.2 Canceling the last generation

Canceling the last generation of analysis is used to restore the analysis to its status before the last generation. The modifications performed since then are not applied.

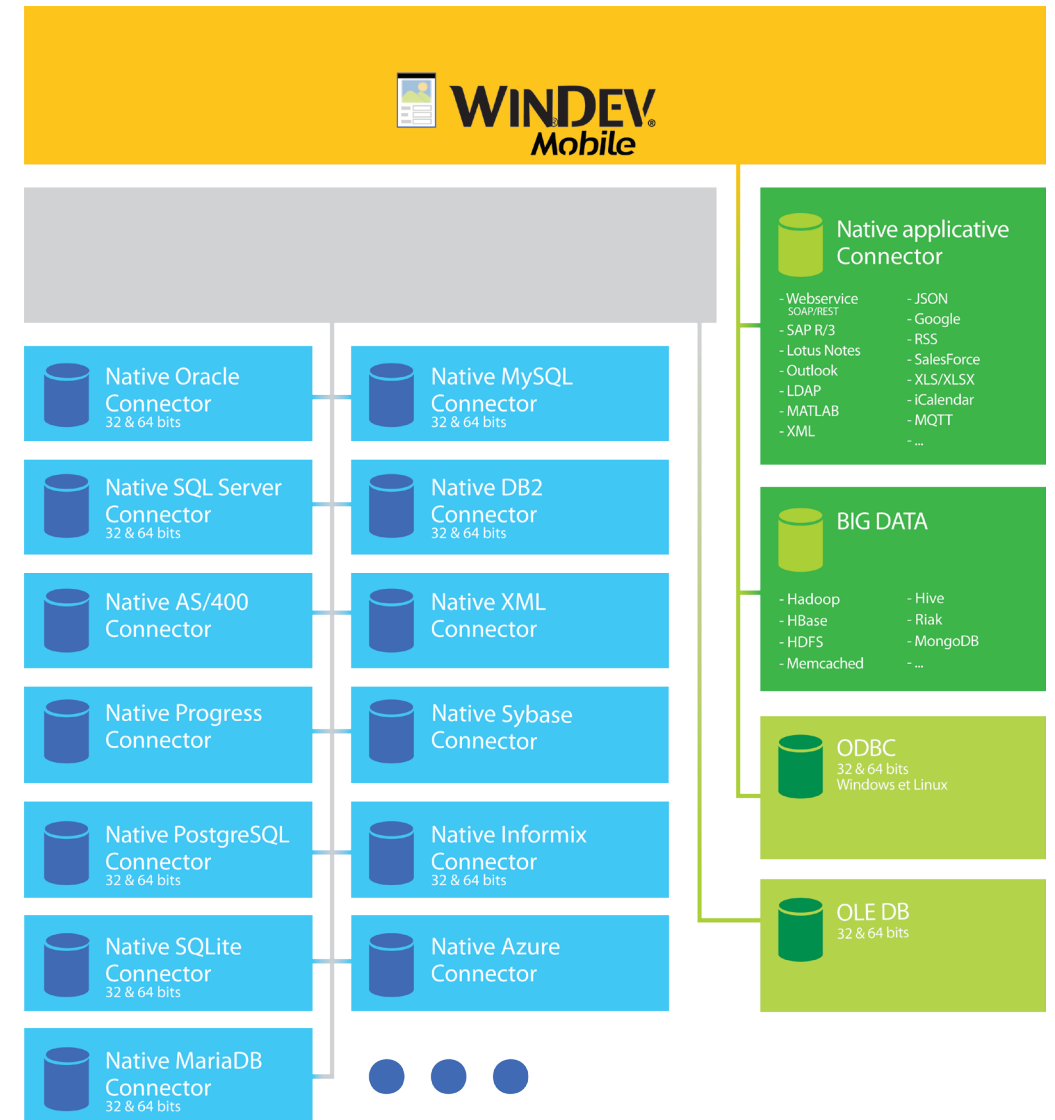
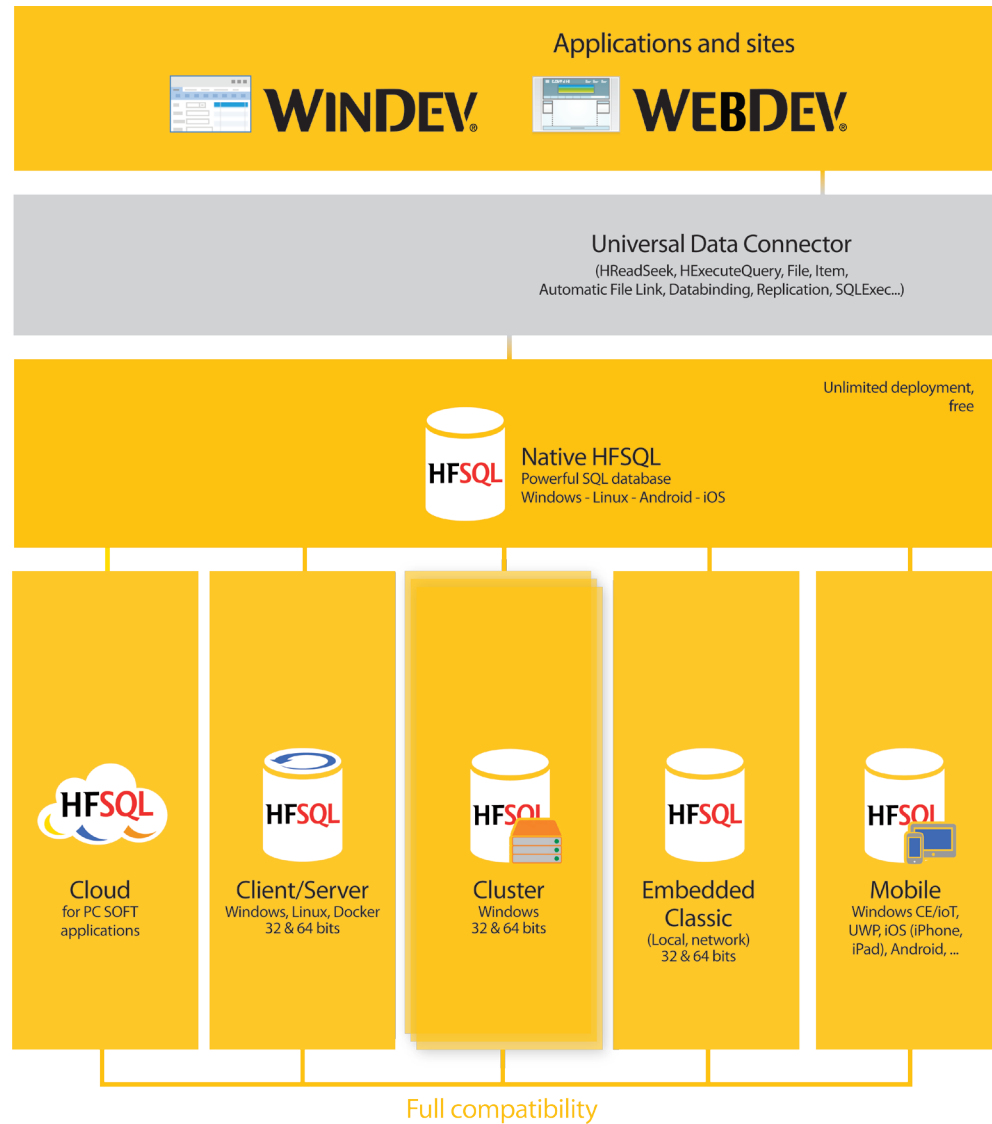
6.3 Resetting the version number to 1

The generation number of analysis can be reset to "1": on the "Analysis" pane, in the "Analysis" group, expand "Generation" and select "Reset the analysis version to 1".

In this case, the version number of analysis is reset to one. No specific action is performed on the data files.

The different types of accessible files

WEBDEV, WINDEV and WINDEV Mobile propose a simple access to most of the databases available on the market.

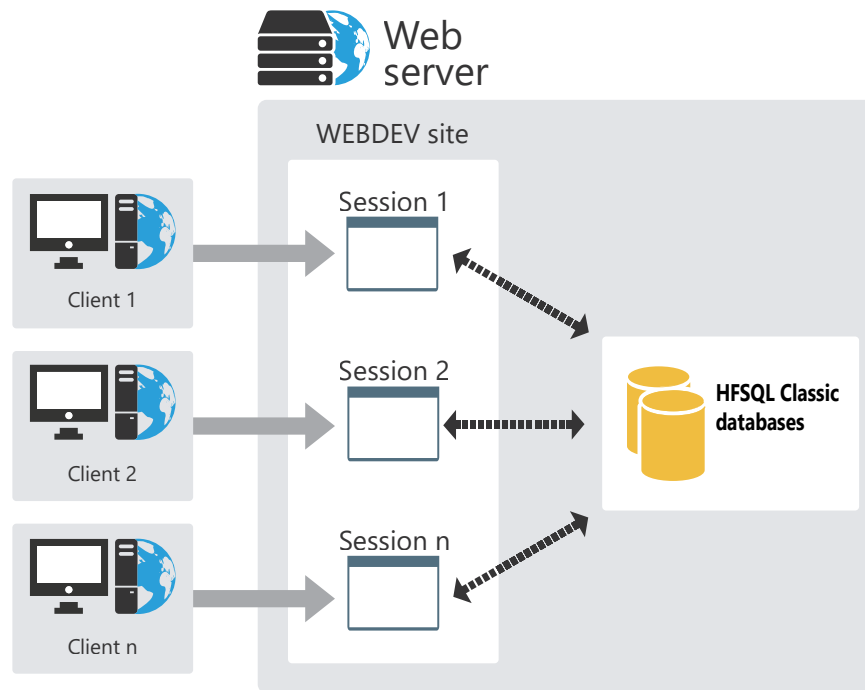


HFSQL Classic

A WEBDEV HFSQL application can operate in Classic mode (called HFSQL Classic) or in Client/Server mode.

The characteristics of the Classic mode are as follows:

- A site that is using HFSQL Classic is run on different browsers. A session is started on the server for each site run.
- The data files are found in a directory that can be accessed by the sessions of the WEBDEV sites (on the Web server or on another computer). Each session physically accesses the data files.
- The processes (query, read/add in a data file, ...) are performed by each session.

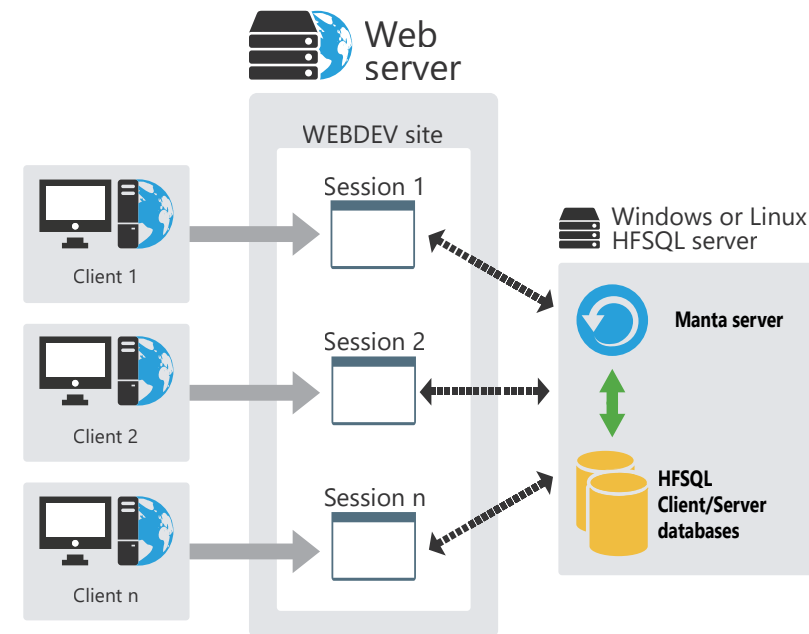


HFSQL Client/Server

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:

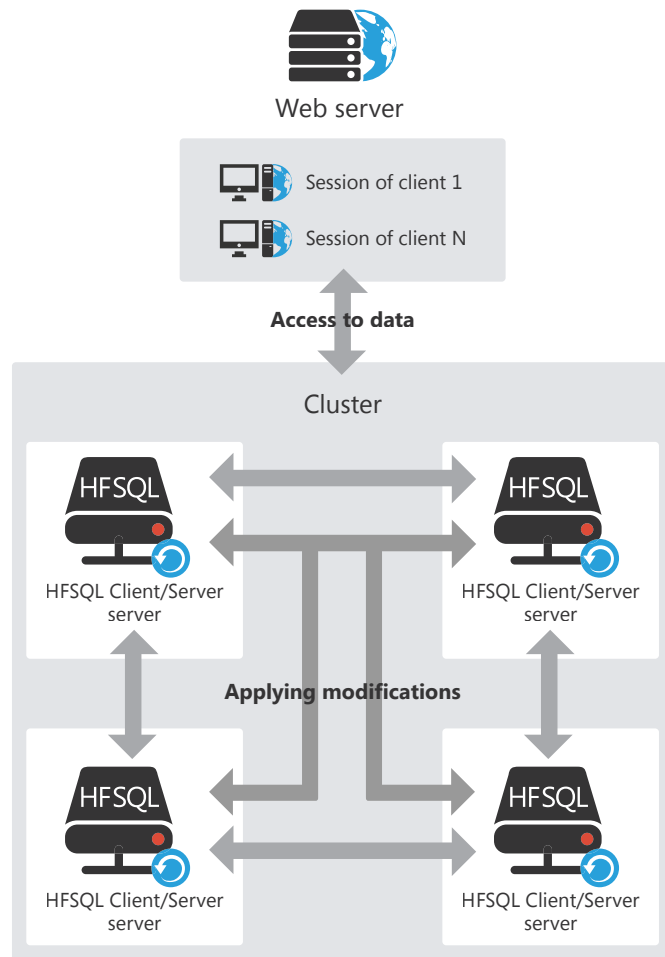
- A HFSQL Client/Server site is run on different user computers. A session is started on the Web server for each site run.
- The data files are found on a server (HFSQL server). Only the server physically accesses the data files.
- All the processes (query, read/write operation in a data file, ...) are performed on the server.



HFSQL Client/Server clusters

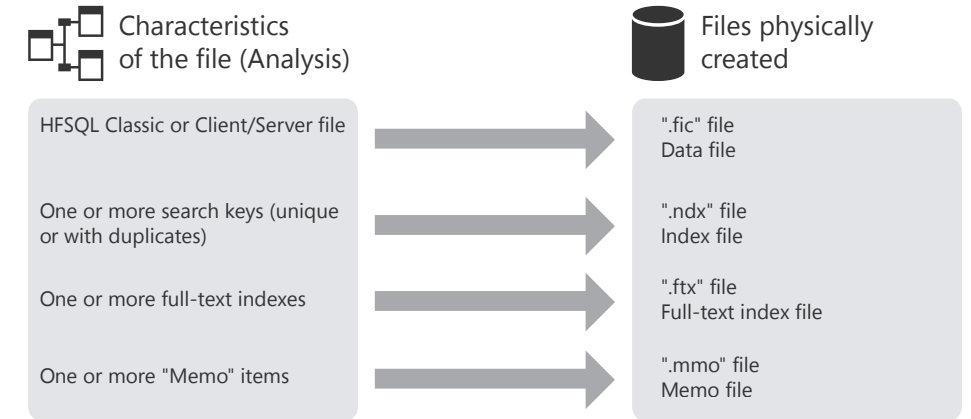
HFSQL Cluster is an extension of the database model of HFSQL Client/Server. In a database cluster, all HFSQL servers contain a copy of databases and they are synchronized in real time.

- The read load can be balanced between the different servers.
- The physical configuration can evolve without any interruption for the client computers.
- If one of the servers crashes, the client is automatically redirected to an operating server.



HFSQL: the physically created files

The data model editor is used to describe the structure of data files. Depending on the information typed in the data model editor, different files are physically created.



Note: This diagram presents the main created files only. Other specific files can be created if the data file is using the logs, the transactions or the replication.

Associating controls with data

A page can display information coming from:

- a database: the controls are directly linked to the items found in the files or queries available in the database.
- variables found in the application code (variables global to the page or to the project or parameters passed to the page).

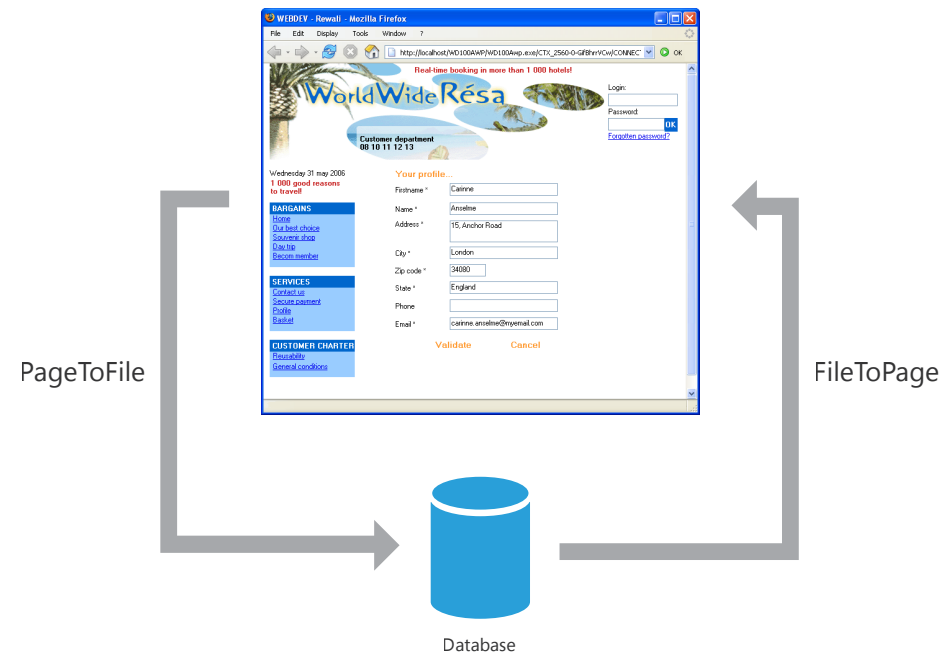
To display this information in a page, the controls of this page must be linked to:

- the different database items.
- the available WLanguage variables.

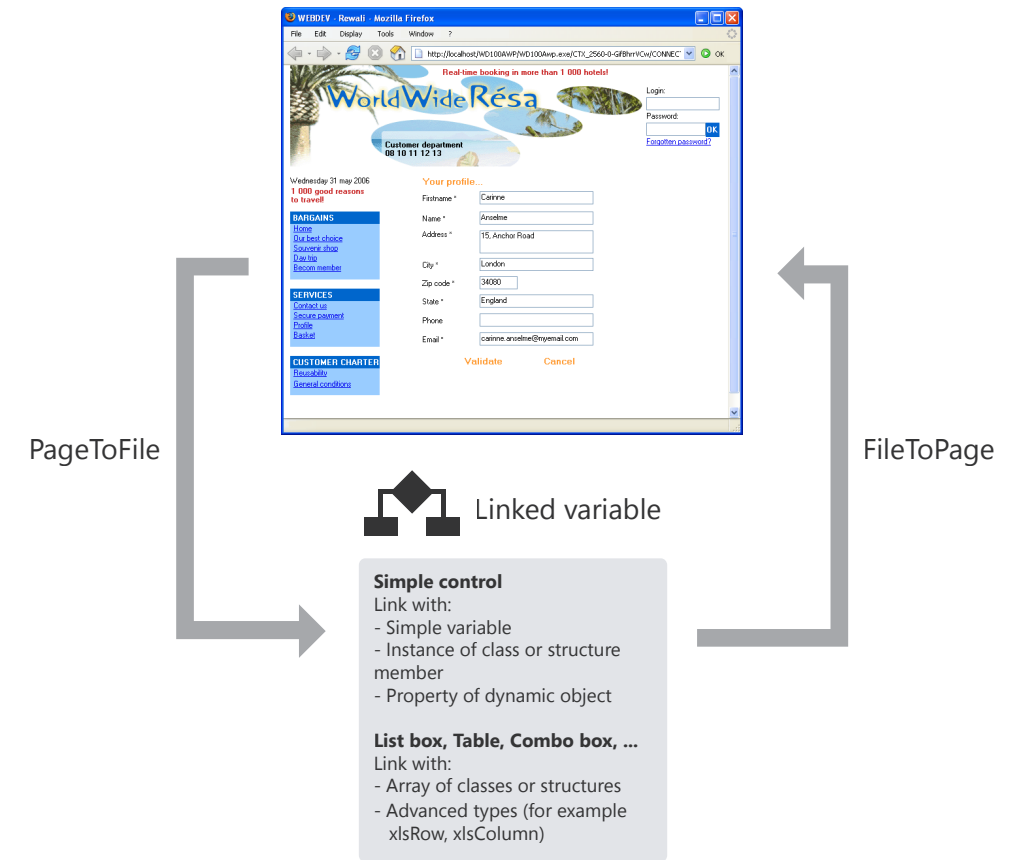
The method for displaying and retrieving information is straightforward:

- The link between a control and an item or variable is defined in the page editor during the control description ("Link" tab).
- **PageToFile** is used to update the record or variable with the data found in the page.
- **FileToPage** is used to update the data displayed in the page with the information saved in the data file or with the information saved in the variable.

Link between control and item



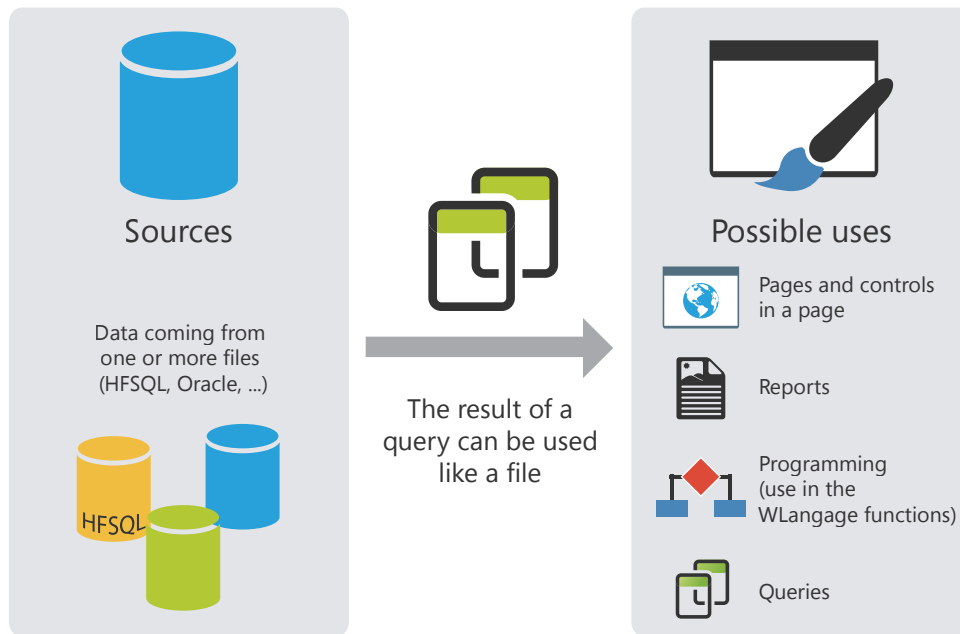
Link between control and variable



The queries

A query is used to interrogate a database in order to view, insert, modify or delete data. The query structure defines the data used.

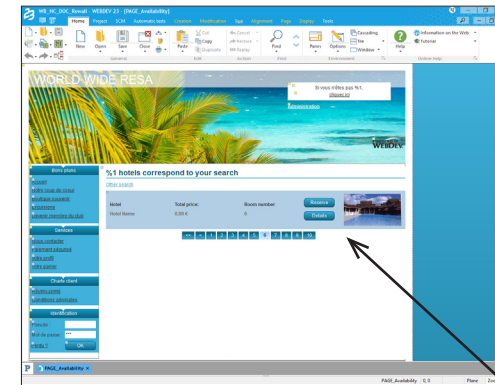
A query can interrogate one or more data files. The query editor allows you to easily create queries without programming.



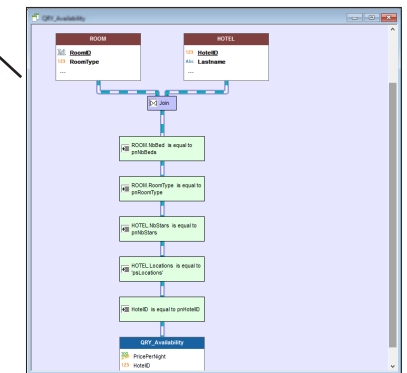
Note: In programming, a query can be handled like a data file. Especially, it can be associated with a display control (a table for example) that will present the data returned by the query.

The embedded queries

The controls found in a page can be linked to a data file or to an existing query, ... These controls can also be linked to a query created when building the control.



Looper linked to an embedded query



Embedded query: MyPage_1\$Query

In this case, the query is included in the page. It is found in the WWH file corresponding to the page. If the WWH file is copied (into another project for example), the embedded queries used by this page will also be copied.

The Table/Looper control

The Table/Looper controls can be used to display a set of information (the content of a data file for example). The content of these controls can come from three different sources:

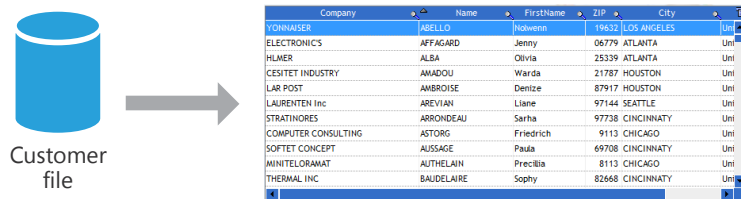
- Browsing Table/Looper controls with direct access
- Memory Table/Looper controls
- Browsing Table/Looper controls loaded in memory

Note: These three fill modes will be presented in details for the Table control. The same concepts apply to the Looper control.

Browsing Table control with direct access

A browsing Table control with direct access is used to directly display the data coming from a data file or from a query. The file structure was described in the data model editor and the data was typed in the site.

Browsing the data file allows you to display data in the Table control. The data file is read for each row displayed: the record read is displayed in a row of Table control.



The number of records displayed in the control can be limited by a filter (**HFilter** used in the initialization code of control).

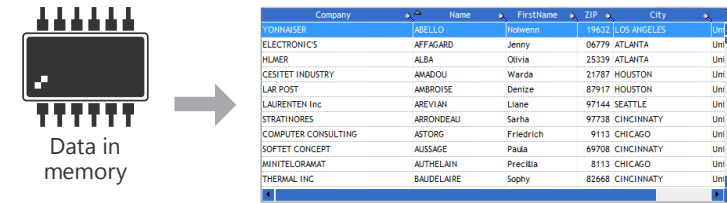
Several WLanguage functions can be used to handle the browsing Table controls. These functions start with "Table".

To add a record into the Table control, add the record into the corresponding HFSQL data file (**HAdd**) and redisplay the Table control with **TableDisplay**.

To delete a record from the Table control, delete the record from the corresponding HFSQL data file (**HDelete**) and redisplay the Table control with **TableDisplay**.

Memory Table control

A memory Table control is used to directly display the data loaded in memory. The data is added into the Table control by programming (by **TableAddLine** for example).



The data being found in memory, the Table control allows you to perform all the operations on the data (sort on any column, search performed in the columns, ...).

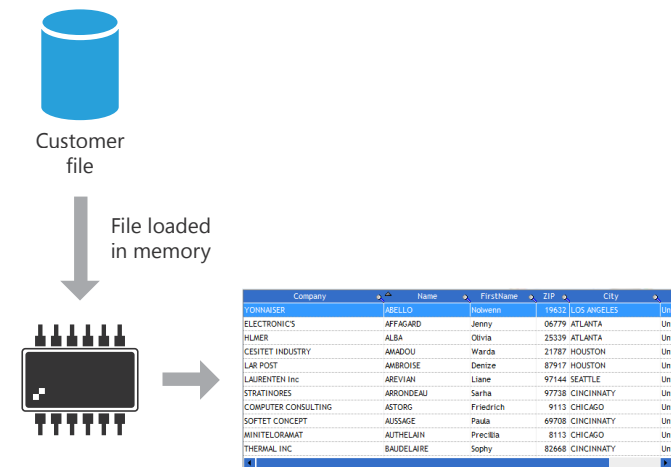
Browsing Table control loaded in memory

The browsing Table controls loaded in memory combine the benefits of browsing Table controls with direct access with the benefits of memory Table controls.

The Table control is linked to the data file but the content of data file is entirely loaded in memory. The sort and the search are available for all columns.

The data not linked to the data file is kept when handling the scrollbar (Check Box column for example).

The records found in the data file being loaded in memory, this type of control is recommended for the data files containing less than 100 000 records (to avoid memory overflow).



Note: The different fill modes (memory, browsing, browsing loaded in memory) are available for the List Box controls, Combo Box controls, Table controls, Looper controls, ...

Retrieving data from a site

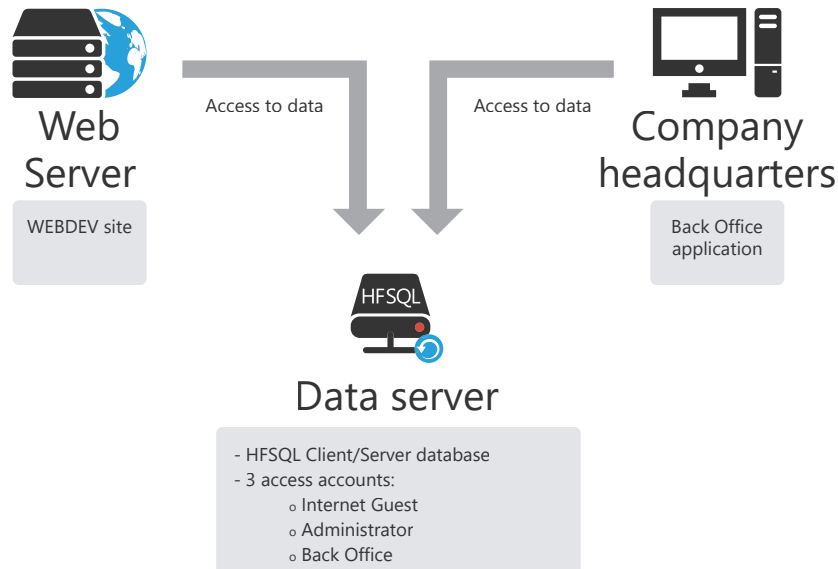
A commercial site allows the Web users to place orders, to make reservations, ... Several methods can be used to retrieve this information.

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. All you have to do is use a HFSQL Client/Server database.

The principle is as follows:



Benefit: The site data is accessible in real time.

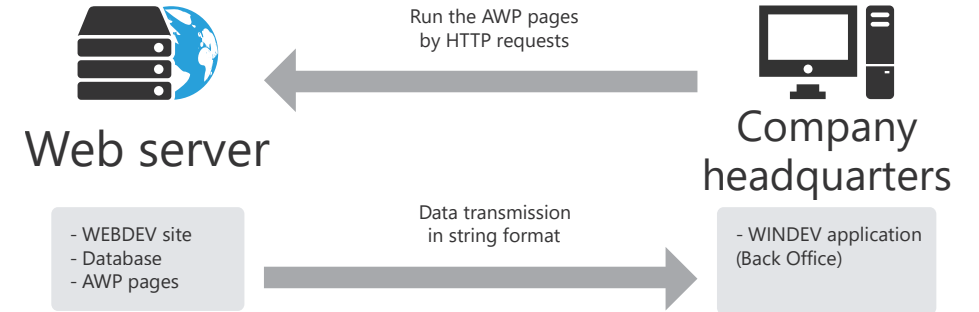
Drawback: Necessity to manage security in optimized way: using a secured connection to access data, opening a specific port to access data (port 4900 by default), configuring the firewall, ...

This solution is recommended for a site installed on a dedicated server.

2 Retrieving the site data

2.1 Via an interrogation AWP page

WEBDEV allows you to create AWP pages. The AWP pages are independent pages that can run processes from the parameters passed in command line to the page. The AWP pages can be used to retrieve the data entered on the site.



The AWP pages found on the Web server are interrogated on a regular basis by the HTTP requests run by the Back Office application. The data is returned in string format and processed by the WINDEV application.

Benefits:

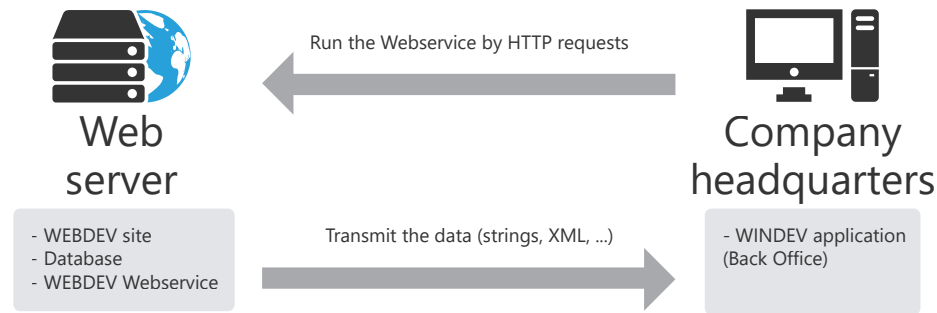
- Solution that can be easily implemented.
- Equivalent to a custom Web service.
- No additional executable is required on the server. No additional session is run on the server.

Drawback: 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 found on the Web server is interrogated on a regular basis by the HTTP requests run by the Back Office application. The data is returned in XML format, as strings (or other) and processed by the WINDEV application.

Benefits:

- Solution that can be easily implemented.
- No additional executable is required on the server. No additional session is run on the server.

Drawback: Unidirectional data retrieval: no data update toward the server.

This solution can be used on a shared server.

2.3 By FTP:

WEBDEV allows you to regularly save the data files of 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 data files
- decompress the files (via a WINDEV application)
- process the data found in the files.

Benefits:

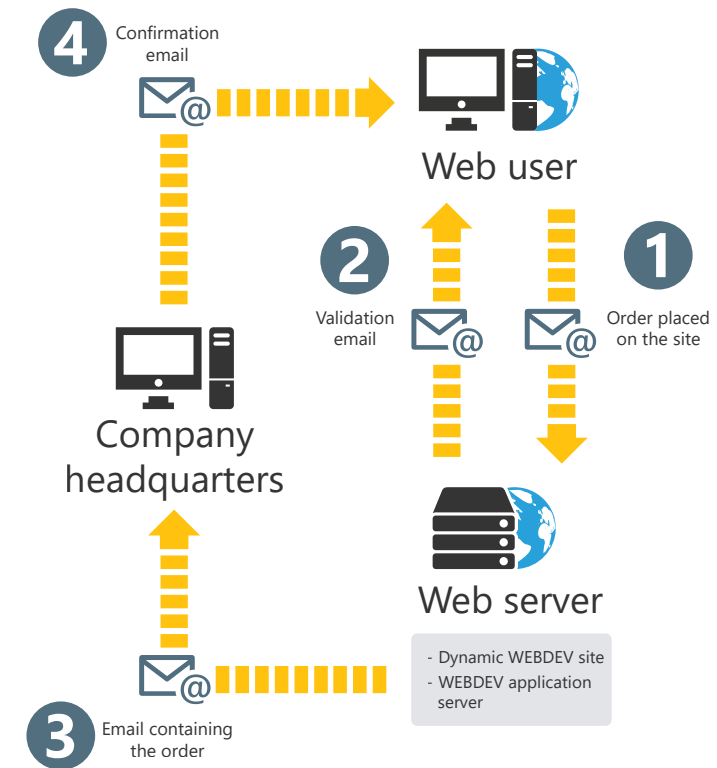
- Solution that can be easily implemented.
- No additional executable is required at the hosting company.

Drawback: All the data files found in the application are retrieved.

This solution can be used on a shared server.

2.4 By emails:

The principle:



1. The Web user places an order on the site.
2. The WEBDEV site informs the Web user that his order was successfully saved. The Web user is informed that a confirmation email will be sent to him.
3. The WEBDEV site sends an email to the company headquarters. This email contains the order placed by the Web user.
4. An email is sent by the company to the Web user to inform him that his order will be processed.

Benefits:

- Solution that can be easily implemented.
- No additional executable is required on the server.

Drawback: 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 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)

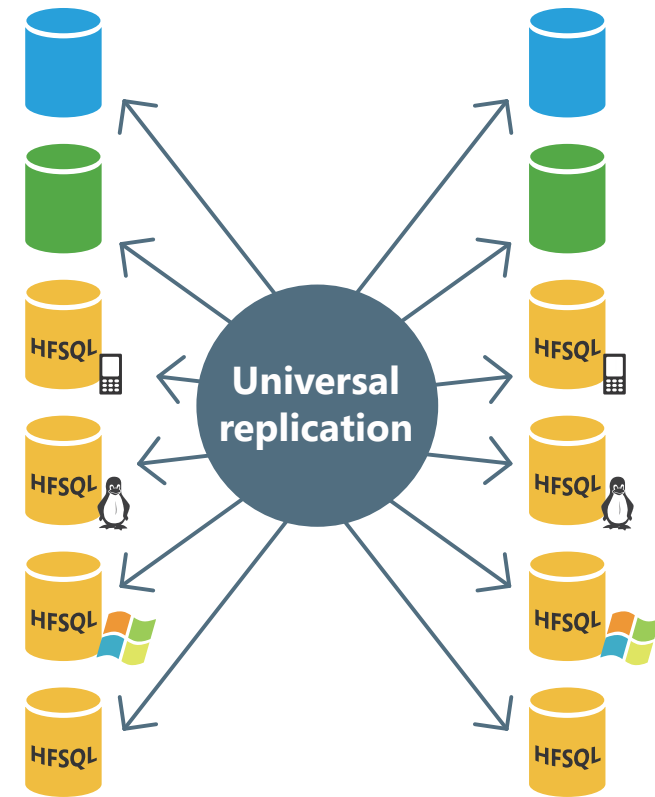
Caution: The replication constraints must be taken into account as soon as the project creation. Indeed, several rules must be followed when describing the analysis and the file items.

The replication can be performed by email or via the assisted universal replication. See the online help for more details.

Universal replication

The universal replication is used to update databases of identical format or databases of different formats (HFSQL Classic, Oracle, SQL Server, ...). You can for example perform a synchronization between a HFSQL Classic database and an Oracle database.

The universal replication is using a centralized model: all the databases are synchronized with a master database. Then, the master database applies the modifications to the other databases.



The synchronization can be adapted to special cases. For example, you have the ability to retrieve the records regarding a specified product or the records created on a specific date, manage the conflicts, ...

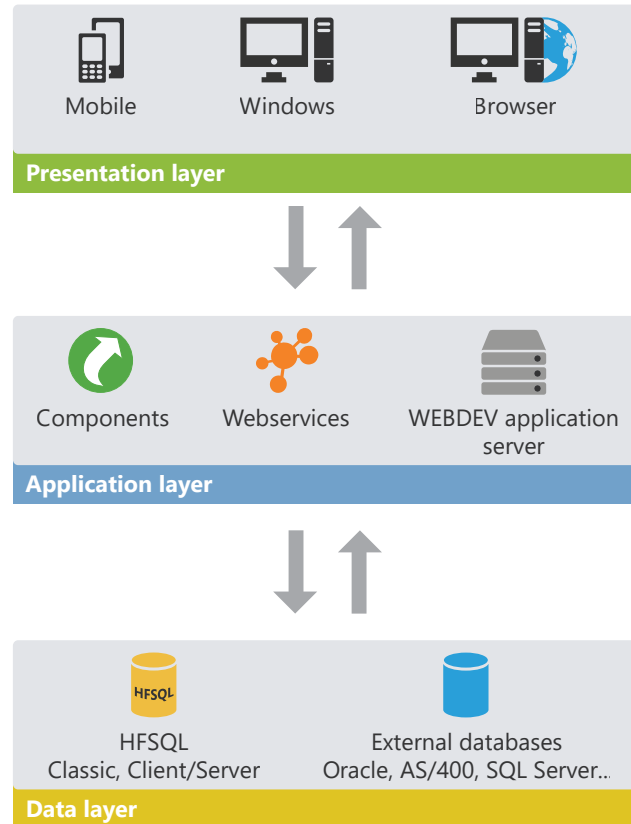
These changes must be done by programming via *HRpIFilterProcedure*.

3-tier architecture

The 3-tier architecture is a model for application architecture. Its basic principle consists in splitting an application into 3 layers:

- the presenter layer: corresponding to the display
- the process or application layer: corresponding to the business processes of application
- the layer for accessing persistent data

The reason for separating the layers is to simplify the maintenance and the future evolutions of application. This provides better security because the access to the database is allowed via the Processes layer only. It also optimizes the teamwork and the multi-target development.



PART 6

Running the test of a Web site

Running a site test: The elements to test

Running the test of a site is a main step when developing a site. The test is used to check the operating mode and the usability of the site.

The main points to check in a WEBDEV site are as follows:

- **the operating mode of site:**
sequence of pages, execution of code typed, ...
- **the site style:**
use different browsers (Internet Explorer, Chrome, FireFox, Edge, ...), use different screen resolutions, resize the browser, ...
- **the specific Web features:**
print, cookies, browser "Back" key, ...
- **the access to the site by different Web users:**
managing logins and passwords, managing concurrent accesses to data files, ...
- **the regression of a site during an update:**
validate the modifications performed in a site and check whether the non-modified features are still accessible.
- **the stress test** (for a site installed on the server):
validate the maximum number of Web users who can access the site.

Note: To find out the coverage of tests run, WEBDEV proposes to use the "Code Coverage". The code coverage is the coverage measurement for the tests run on a site. Each code line run is considered as being "checked".

Caution: The code coverage does not allow you to find out whether tests are successful: it only allows you to find out whether the code line was run.

The code coverage is used to detect the code lines that have never been run.

How to run a site test?

Several tools can be used to run the test of a WEBDEV site:

- the main WEBDEV editor.
- the WEBDEV administrator. On the development computer, the test of dynamic sites can be run from a test page created by the administrator.
- WDTTestSite, tool for running stress tests.

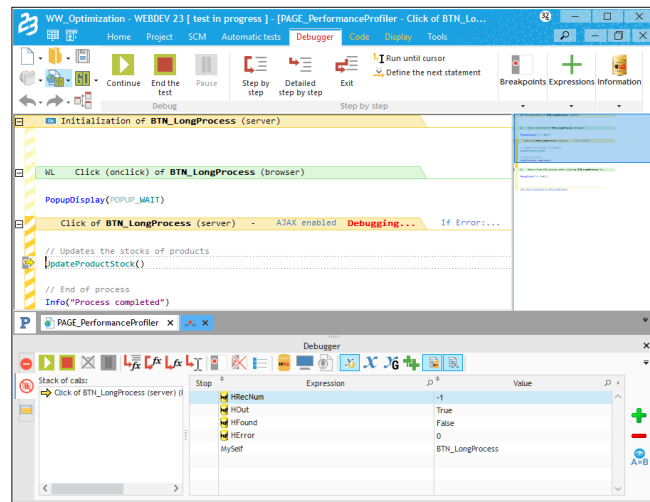
The table below presents how and when these test modes can be used.

Test mode	Type of test	Running the test
Publisher	<p>Running the test from the editor allows you to test:</p> <ul style="list-style-type: none"> • the site features, • the code used in the different processes. A powerful debugger allows you to monitor the execution of different processes. • the use of the site with different browsers (installed on the development computer). <p>This type of test:</p> <ul style="list-style-type: none"> • allows you to use the debugger. • is taken into account by the "Code Coverage". 	<ul style="list-style-type: none"> • On the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Trace the project". • On the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Debug the project from the home page". <p>The rights granted to the Web user (access rights, write rights and read rights) correspond to the rights granted to the developer.</p>

<p>WEBDEV administrator</p> <p>(dynamic WEBDEV site only)</p>	<p>Running the test from the administrator allows you to test:</p> <ul style="list-style-type: none"> • the site features, • the features specific to the Web (cookies, ...), • the use of the site in real conditions. 	<p>Start the WEBDEV administrator ("Start" menu) and click the "Test page" button ("Advanced" tab). This test allows you to use the WEBDEV site in conditions similar to the ones of a Web user. 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>
<p>WDTTestSite (dynamic WEBDEV site only)</p>	<p>WDTTestSite is used to run stress tests. WDTTestSite is used to start several simultaneous connections to a WEBDEV site. Each connection performs a set of actions in the WEBDEV site (preset scenario). 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"> 1. Create a test scenario (with WDTTestSite). 2. Install WDTTestSite and the scenario on the different computers to simulate an important number of accesses. 3. Specify the number of connections to establish. 4. Run the test scenario on each computer. <p>Notes: Each computer establishes a defined number of connections. Each connection performs the selected scenario.</p>
<p>Automatic tests</p>	<p>The automatic tests are used to run the test of procedures and classes found in a site at different development levels. 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. This type of test is taken into account by the "Code Coverage".</p>	<ol style="list-style-type: none"> 1. Display the procedure or the class in the project explorer. 2. Display the popup menu of the element (right mouse click) and select "Create a unit test".

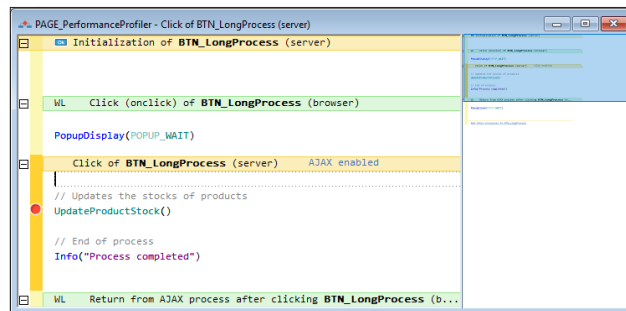
The debugger

To help you optimize your code, a debugger is supplied with WEBDEV. The debugger allows you to run your dynamic sites step by step, by viewing the WLanguage code run, the content of variables, ...



To start the debugger, use:

- **a breakpoint positioned in the code editor:** the debugger will be automatically started when the line preceded by a breakpoint is run. To add a breakpoint, all you have to do is click in front of the code line: a red bullet appears.





- **the STOP server keyword of WLanguage in your WLanguage code:** the debugger will be automatically started when this code line is run.
- **the "Trace the project" option:** the debugger is started when running the test of dynamic site. on the "Project" pane, expand "Test mode" and select "Trace the project".

When can I use the debugger?

The debugger can always be used when running a test from the editor. To do so, breakpoints must be positioned at the requested locations in order to debug the site step by step.

The different types of tests available from the editor are as follows:

- **"Go" on page:**
The test of current page in the editor can be run via  found among the quick access buttons of WEBDEV.
- **"Go" on project:**
The project test on the development computer can be run via:
-  found among the quick access buttons of WEBDEV,
- on the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Debug the project".
- **The deployed project:**
You have the ability to debug the deployed project. This feature is very useful to reproduce a problem that occurs in specific conditions. To start the deployed project, on the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Connect to a site and debug the connection".
Note: This option can be used with a development server or with a deployment server. The server must be configured to allow the remote debugging.
- **The deployed project is currently used:**
You have the ability to debug the deployed project that is currently used. This feature is very useful to reproduce a problem that occurs in specific conditions. Most common case: a problem occurs while you are using the site: to start the debugger, all you have to do is insert a breakpoint into the code of your project. To use a session of deployed project, on the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Debug an existing connection".
Note: This option can be used with a development server or with a deployment server. The server must be configured to allow the remote debugging.

Test of a site in practice

1 Overview

WEBDEV proposes several methods for running the test of your sites:

- test of entire project,
- test of single page,
- test of single query (see the "Reports and Queries" guide for more details),
- test of single report (see the "Reports and Queries" guide for more details),
- running the project step by step,
- test of site performances,
- regression test/automatic test,
- stress test.

The test of entire project is used to simulate the start of site. This allows you to test the entire site, even if its development is not ended. As soon as a problem occurs, you have the ability to start the debugger to find out and fix the problem.

The test of single page is used to run the current page. This allows you to run the test of your project from a given page or to check the operating mode of a page as soon as its development is ended. Like for the project test, the debugger can be started as soon as a problem occurs.

2 Running the test of WEBDEV project

2.1 Running the project test 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 project test can be run regardless of the current element in the editor.

Note: The test browser used to run the project test can be chosen:

- in the WEBDEV options: on the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WEBDEV". The browser can be selected in the "Web" tab.

The test of single query is used to run the current query. This allows you to check the operating mode of a query as soon as its development is ended.

The test of single report is used to run the current report. This allows you to check the operating mode of a report as soon as its development is ended. Like for the project test, the debugger can be started as soon as a problem occurs.

Running the project step by step is used to start the debugger when the site is started. This solution allows you to closely monitor how the site runs.

The performance test of your site is used to check and optimize the execution time of your site.

The regression test (or automatic test) is based on the execution of scripts. It is used to check that, when running your site, ... the existing features are still supported.

The stress test is used to start several simultaneous connections to the same dynamic WEBDEV site.

In addition to these methods, WEBDEV also proposes to find out the "Code coverage" of site, which means the coverage measurement for the tests run on a site. See the online help for more details.


- in the options of test mode: on the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Test browser".

Different types of test

To run the test of a static site from the editor:

1. On the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Debug the 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 from the editor, several methods available:

- On the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Debug the project" (or [Ctrl] + [F9]).
- Click  among 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 run the test of a static + dynamic site from the editor:

- **to run the test of static site part:** perform the operations corresponding to the test of a static site.
- **to run the test of dynamic site part:** perform the operations corresponding to the test of a dynamic site.

Dynamic site: Start

The following modules are automatically started when running the test of a dynamic WEBDEV site:

- The Web server installed on the computer and configured for WEBDEV when installing WEBDEV. The test cannot be run if the Web server is not started.
- The WEBDEV administrator (WD230ADMIN.EXE). The administrator is used to manage the connections to the Web server and to configure the WEBDEV sites.
Note: a project test can be run from the test page of administrator ("Advanced" tab of WD230ADMIN, "Test page" button).
- The WEBDEV engine (WD230AWP.EXE). The WEBDEV engine is used to manage the requests made by the Web users from their browser and to return the corresponding dynamic HTML page.
Note: The WEBDEV engine is started only if the project contains dynamic pages.
- The Internet browser.
The Internet browser is used to display the HTML

pages of WEBDEV site.

2.2 Running the project test from the WEBDEV administrator

Running the test from the WEBDEV administrator (WD230Admin) is used to test:

- the site features.
- the specific Web features (use of cookies, ...).


Note: The WEBDEV administrator can only be used to run the test of dynamic sites (or the test of 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 site test at least once from the WEBDEV administrator.

To run the test from the WEBDEV administrator:

1. Start the WEBDEV administrator: on the "Tools" pane, in the "Web utilities" group, click "WAdmin".
2. In the "Advanced" tab of WEBDEV administrator, click the "Test page" button.

To stop the test, display the WEBDEV administrator (click the icon  found in the taskbar) and click the "Disconnect" button ("Connections" tab).

Note: 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" pane, 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

WDTestSite is used to run stress tests: WDTestSite is used to start several simultaneous connections to a dynamic WEBDEV site.

Each connection performs a set of actions in the WEBDEV site (preset scenario).

See the online help for more details.

3 Running the test of a remote WEBDEV site

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 on the user computers directly.

From your office in London, you have the ability to debug a site running on a Web server in Taiwan. The debug operation is done without having to go anywhere, on the final configuration directly.

Two features are available:


- Starting and debugging the site on a remote application server.
- Debugging a site currently used on a remote application server.

For these two features, a specific configuration is required for the remote computer.

4 Running the page test

4.1 Running the page test from the editor

To run the page test from the editor:

1. Open the page whose test must be run.
2. Click  among the quick access buttons of WEBDEV menu. You also have the ability to press [F9].
3. The editor is automatically minimized and the page is run.

During the test, all the page features can be run. You will have the ability to open other pages for example.

4.2 Stopping the page test

Several methods can be used to stop the test:

- **1st method:**
Close the page whose test is currently run. WEBDEV displays the editor that was used at the beginning of test.
- **2nd method:**
 - Go back to the editor with the taskbar or with [Alt] + [Tab].
 - Confirm that the test must be stopped. WEBDEV displays the editor that was used at the beginning of test.

5 Tracing a project

5.1 Principles for debugging

Debugging an application consists in:

- checking the operating mode of a process,
- understanding the operating mode of an existing program,
- checking the value of variables,
- checking the operating mode of special cases in an application or in a site.

The debugger is used to perform these operations.

Note: WEBDEV also includes several trace tools (trace window, information box, ...). See “Debugging without debugger”, page 201 for more details.

5.2 Overview of debugger

The debugger is used to trace the WLanguage programs in order to help you improve these programs.

The source code run is viewed on the screen. The processes run are sorted in hierarchical order in the "Debugger" pane.

The value of variables can be viewed:

- individually in the rollover tooltip of each variable.
- in the "Debugger" pane.

5.3 Features of debugger

The debugger allows you to:

- find out the call stack
- view the content of variables or expressions
- run the code step by step with ability to skip blocks.
- use conditional breakpoints
- modify the code while continuing the execution,
- ...

5.4 Debugging without debugger

In some cases, running a program with or without debugger may be different.

Indeed, the debugger introduces pauses in the process execution during which several tasks are performed by WINDEV.

Therefore, the debugger cannot be used in a procedure called by timer or in the scrollbar code.

Note: See the online help for more details about the debugger limits.

To debug these applications, you may want to follow the evolution of a value, how different procedures are called, ...

This information can be:

- displayed on the screen
- stored in a trace file.

Caution: If the information is displayed on the screen, it must be displayed during the application tests only.

Displaying information

Two tools can be used to display information:

- **the information boxes:** *Info* function.
Caution: Displaying an information box is a locking operation.
- **the trace window:** *Trace* function.
The trace window is displayed in the top left corner of screen, without interrupting the program execution.

Managing the display of debug information

Displaying the debug information on screen is useful in test mode only.

Any unsuitable display must be removed before distributing an application.

To avoid any oversight, we advise you to manage the display of debug information via a global procedure.

For example:

```
PROCEDURE MyTrace (StringToTrace)
IF InTestMode() THEN
Trace (StringToTrace)
END
```

In this code, depending on the result of *InTestMode*, the trace window appears during the application test only.

Such procedure allows you to leave the call to the trace windows in the application code without any risk of displaying it on the end-user computers.

The call to the trace procedure is similar to the use of *Trace*:

```
MyTrace ("Customer: "+...
Customer.CustomerNum)
```

Creating a trace file

During long processes (batch processes, ...), to check the operating mode of program, you must keep a physical trace of processes run (a text file for example).

The following procedure is used to manage the trace display:

- on screen (/DEBUG parameter in command line).
- in a text file (default mode).

```
PROCEDURE MyTrace (StringToTrace)
FilePath is int
FilePath = ...
fDataDirUser()+...
ProjectInfo(piProjectName)+".txt"
FileNum is int
DebugMode is boolean = False
IF Position(CommandLine(),...
"/DEBUG") > 0 THEN
DebugMode = True
END
IF DebugMode THEN
Trace (StringToTrace)
ELSE
FileNum = fopen(...
FilePath, ...
foCreateIfNotExist+...
foWrite + foAdd)
If FileNum <> -1 THEN
DateTimeTrace is DateTime
DateTimeTrace = SysDateTime()
DateTrace is Date
DateTrace = MyDate..Date
TimeTrace is Time
TimeTrace = MyDate..Time
fWriteLine (FileNum, ...
DateToString(DateTrace)+...
" - "+TimeToString(TimeTrace))
fWriteLine (FileNum, ...
StringToTrace)
fWriteLine (FileNum, " ")
fclose (FileNum)
END
END
```

Notes:

- The trace file is created by default in the data directory of user. This file is named like the project. 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 content of trace file:

```
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 run the test of your site.
- During this test, the performance profiler keeps track of all actions performed and the corresponding processes run.

At the end of test, the performance profiler displays:

- the 10 most time consuming operations.
- all the actions performed in the site whose test was run, sorted by duration (from the longest one to the shortest one).

You can select a process in order to analyze the reasons for its processing time in order to optimize it.

6.2 Starting the performance profiler

To start the performance profiler, on the "Project" pane, in the "Audit and performances" group, expand "Analyze the performance" and select "Analyze the performance".

The project is automatically run in test mode. The process to optimize can be run in your site.

To go back to the editor, all you have to do is close your application or your site.

Then, the performance profiler displays the monitoring result.

Note: The performance profiler should be used to optimize your site (before it is distributed for example).

6.3 Reading the result of performance profiler

The performance profiler presents the monitoring result in several tabs:

- the "Summary" tab presents the ten longest processes.
- the "Mapping" tab presents a graphical view of main processes.
- the "Details" tab presents all the processes run during the application test (from the slowest one to the fastest one).
- the "Calls" tab is used to view the details of operations performed in a process.

The following information is displayed for each process:

Function	Function, process or procedure run.
Total Time	Execution time of function.
Nb of calls	Number of calls made to the function (procedure or process).
Time 1 call	Execution time of a call to the function (procedure or process).
code %	Percentage of code run outside the call to a WLanguage function or outside the call to a custom function or procedure.
Parent	Process that called the function.

Note:

- The "Full execution" caption represents the total amount of time for running the 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 process to optimize is chosen according to several criteria:

- **the execution time of process.** The longest processes must necessarily be optimized.

- **the percentage of time spent in the process of function or procedure.** The higher this percentage is, the greater the number of processes that can be optimized in the code.

Note: If the process corresponds to a WLanguage function, it is fairly hard to optimize it.

7 Regression tests

7.1 Overview

Several test tools are available to guarantee the quality of your applications:

- The test mode (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 are used to easily check all the features proposed by your applications.

7.2 Automatic tests

Each test contains a scenario that can be directly edited in the product interface. This scenario is written in WLanguage and it can be modified at any time.

These tests can be run before each deployment in order to check the operating mode of a site after several modifications.

The following elements can be checked:

- the sets of procedures
- the classes

Each test is associated with a WLanguage code: the test scenario. This scenario can be viewed in the code editor. The code of tests can be modified.

The tests and the associated code are not distributed to the end users. Therefore, the number of tests for a site has no incidence on the size of site supplied to the end users.

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

7.3 WDTTestSite

WDTTestSite is used to run different tests on a WEBDEV site.

The following tests can be run by WDTTestSite:

- **Stress test:**
The stress test consists in simulating the connection of several Web users to a WEBDEV site. Each Web user runs a set of operations (scenario) simultaneously.
 - **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 that the concurrent accesses to 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 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 optimizing the WLanguage code.
- See the online help for more details (keyword: "WDTTestSite").

PART 7

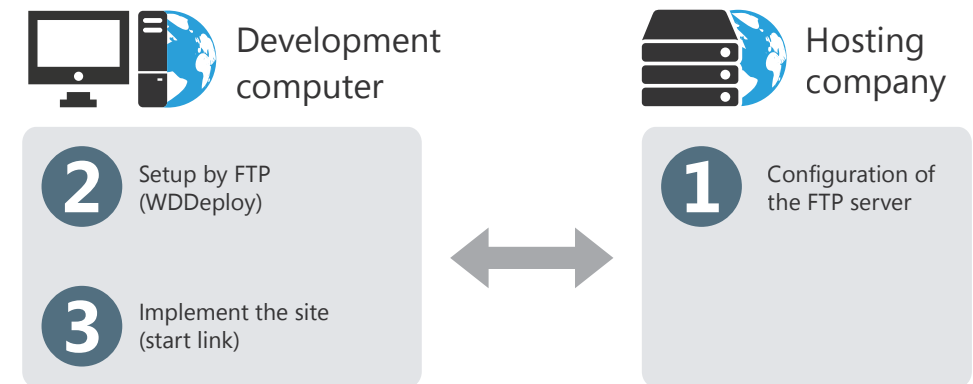
**Deploying a
Web site**



Deploying a static site

Principle

The deployment of a static site is performed by 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.

During the deployment by FTP, the developer must specify in WDDeploy:

- the information given by the hosting company.
- the files to install.

Deploying a static site, a semi-dynamic site or a PHP site

1 Overview

When developing a static site, a semi-dynamic site or a PHP site, one of the important steps consists in deploying this site on a Web server (at a hosting company or in Intranet).

- **deployment via network:** recommended when the server is directly accessible.

WDDeploy is used to simplify the deployment of your static sites.

Two deployment methods are available:

- **remote deployment (via FTP):** recommended when the server is not directly accessible.

Note: To deploy a dynamic site, see "Deploying a dynamic site", page 214.

2 Creating the setup

To deploy your static site:

1. On the "Project" pane, in the "Generation" group, click "Deploy the site". WDDeploy starts.

2. Create the profile corresponding to your site. This profile contains the following information:

- Location of local files ("`<ProjectName>_WEB`" sub-directory 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.

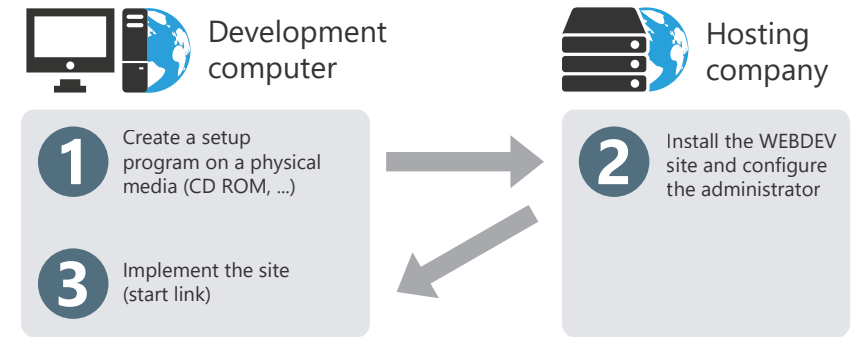
See the online help for more details.

Deploying a dynamic WEBDEV site

The different types of deployment

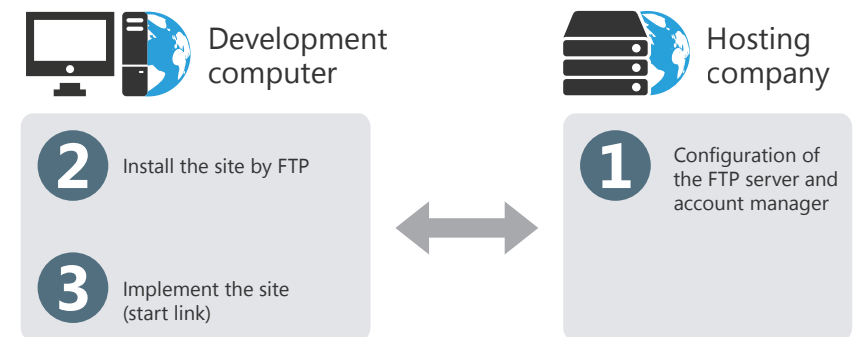
Three methods can be used to deploy a dynamic site:

- **deployment by physical media (CD, ...)** of WEBDEVsite, with creation of a setup version that can be supplied on CD to the hosting company

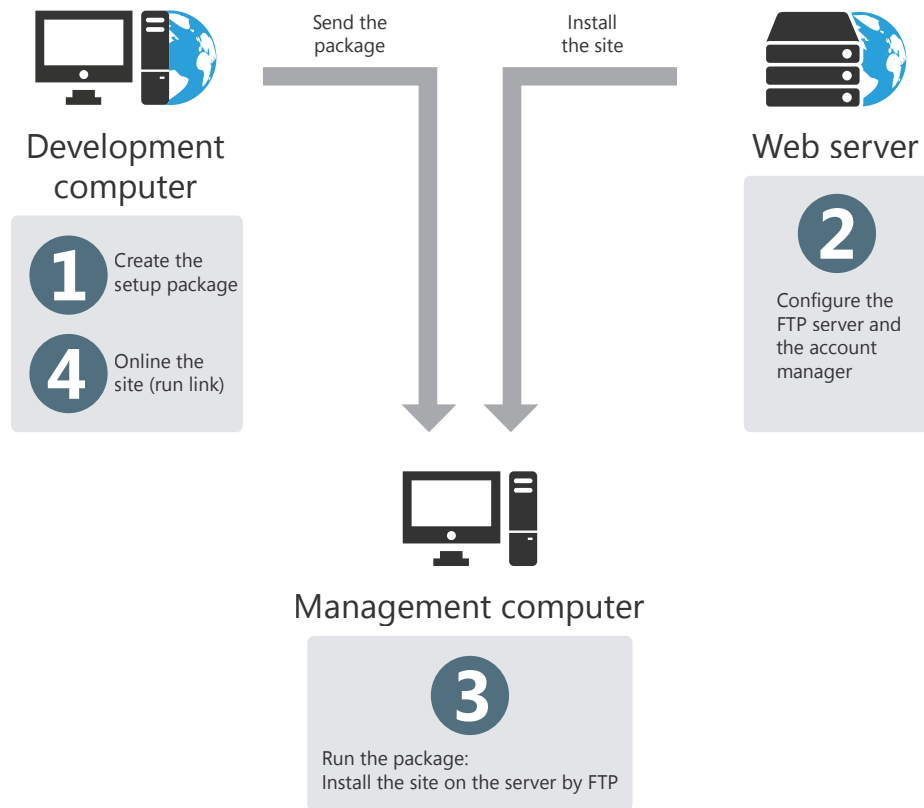


Note: a WEBDEV application server can be included in the setup supplied by physical media.

- **remote deployment from the development computer (file transfer by Internet, FTP)**



- *remote deployment from a management computer (file transfer by Internet, FTP)*



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 FTP) from the development computer is recommended if the deployment computer cannot be easily accessed. A standard FTP server must be found on the deployment computer.

Before performing a remote setup, the hosting company must configure the FTP server and the WEBDEV account manager.

The remote setup (by FTP) 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, FTP account, ...) are specified when the package is run on the administrator computer.

Deployment via physical media: necessary information

When deploying by physical media, the information regarding the setup and the site setting is supplied both in the wizard for setup creation and in the setup wizard.

When preparing the setup, you must supply:

- the setup directory,
- the files to install,
- the generation directory of setup.

During the setup, you must supply:

- the directory of data files,
- the directory of site files.

After the setup, the site characteristics (connection time-out, number of connections, ...) must be configured in the WEBDEV administrator (deployment version).

Deployment by FTP from the development computer: necessary information

No setup can be run without this information. Ask the provider to get this information.

Before deploying by FTP from the development computer, the hosting company must communicate the following information to the developer:

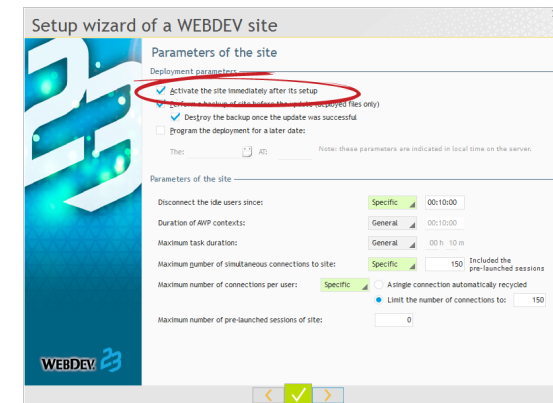
- Name of Web server where the setup will be performed (or its IP address)
- User name and associated password defined in the WEBDEV account manager
- User name and password defined in the FTP server

On-lining a dynamic WEBDEV site

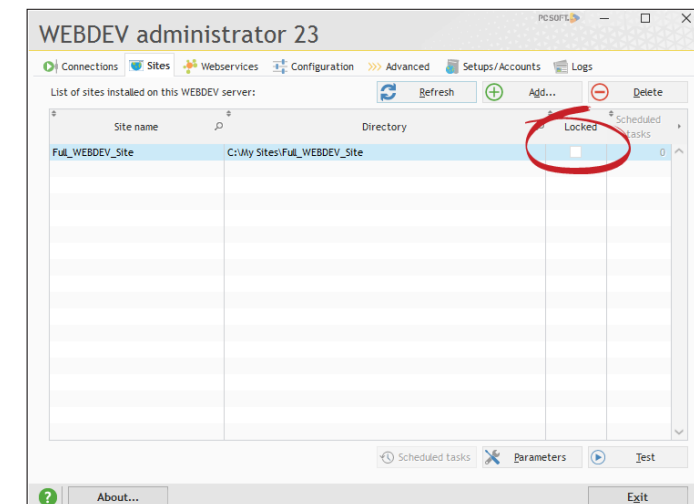
Your dynamic WEBDEV site can be accessed as soon as it is on-lined. On-lining a dynamic WEBDEV site consists in activating the site.

Several methods can be used to activate a dynamic WEBDEV site:

- during the setup by FTP: a check box is used to automatically activate the site after its setup.



- with the remote administrator.
- with the WEBDEV administrator found on the deployment computer ("Locked" option in the "Sites" tab).



When deploying by FTP, the developer must specify in the wizard:

- the information given by the hosting company
- the files to install
- the name of sub-directory of data files
- the different parameters for connecting to the site

Notes:

- A delayed setup can be performed at the specified date and time.
- The setup and the update by FTP are secured by password and data encryption.

Deployment by FTP from a management computer: necessary information

When creating the setup program, no specific information is required.

When running the setup program of 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)
- User name and associated password defined in the WEBDEV account manager
- User name and password defined in the FTP server

Notes:

- A delayed setup can be performed at the specified date and time.
- The setup and the update by FTP are secured by password and data encryption.

Deploying a dynamic site

When developing a dynamic site, one of the important phases consists in deploying this site on a Web server (at a hosting company or in Intranet).

Two deployment methods are available:

- **remote deployment (via FTP):** recommended when the server is not directly accessible.
- **deployment by physical media: CD, ...:** recommended when the server is directly accessible.
- **deployment by package:** recommended when the server is not directly accessible and if the server characteristics are unknown.

1 Implementing a library

Before performing a setup, all the objects found in your dynamic site must be included in a library. A library is a file that groups all the elements created during the development (description of database and pages, compiled source codes, ...). The HTML pages and the images are not included in the library. The library is automatically implemented when creating the setup procedure.

Once the library is created, the wizard for setup creation starts. You can choose to perform:

- **a setup by FTP (remote setup)**
Before performing a remote setup, the hosting company must have:
 1. Created and configured your FTP account on the server.
 2. Created and configured your WEBDEV account on the server.
The hosting company must supply your logins and passwords for these different accounts.
- **a deployment in PC SOFT Cloud:**
PC SOFT proposes a hosting platform based on Cloud Computing: you deploy without having to worry about the hardware constraints and you are invoiced according to the real use of your applications.
A PC SOFT Cloud account is required to perform this type of setup. The wizard proposes to create this account if necessary.
- **a setup by FTP via the test hosting service of PC SOFT:**
PC SOFT proposes to host your test sites. The

Notes:

- For each Web server that hosts WEBDEV sites, the hosting company must own a license for the WEBDEV application server.
- To create the site setup, on the "Project" pane, in the "Generation" group, click "Deploy the site".

To deploy a static site, see "Deploying a static site, a semi-dynamic site or a PHP site", page 208.

setup is performed from the environment. This allows you to run the site test in real conditions. A PC SOFT developer account is required to perform this type of setup. The wizard proposes to create this account if necessary.

- **a deployment package:**
A deployment package is an executable containing all the elements required to deploy a WEBDEV site. Then, this package can be run from any computer to install the Web site on the server by FTP.
To create this type of setup, there is no need to know the characteristics of FTP account and WEBDEV account found 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 not already done).
 2. Install your WEBDEV site by running the executable program *Install* supplied with your setup.

Note: You also have the ability to create a setup via stand-alone physical media, containing the setup of a WEBDEV application server limited to 10 connections. This solution is ideal for proof-of-concepts presented on a laptop.

2 Remote deployment (by FTP)

2.1 The steps

The deployment steps are as follows:

1. Installing the WEBDEV application server at the hosting company (optional step). See the documentation about the WEBDEV application server for more details.
2. WEBDEV account (in the administrator) and FTP account (in the FTP server) created by the hosting company.
3. The hosting company supplies the names and associated passwords as well as the name (IP address) of server.
4. Creating the remote setup from the editor: on the "Project" pane, in the "Generation" group, expand "Deploy the site" and select "Deploy the site remotely". The necessary files are copied into the transfer directory.
5. On-lining the WEBDEV site.

2.2 Creating the setup

You must define in the wizard:

1. **The information regarding the remote computer.** This information must be supplied by the hosting company:
 - **Server address** (name of a computer accessible by network, IP address or Internet address).
 - **Characteristics of WEBDEV account** (user name and password).
 - **Characteristics of FTP account** (user name and password).

To easily identify the server characteristics for a forthcoming update, give a name to this server. This name will be displayed (followed by the server address) in the window for choosing the type of setup.

2. **The files to install.** By default, the wizard selects the library, the data files, the images and the HTML pages.

3. Configuring the setup: Choose 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 server ones. Make sure that they are valid.
- **Activating the site immediately after setup:**
The site will be available as soon as it is installed: you will be able to connect to it by using the relevant 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 as soon as it is installed.
These parameters can be modified if necessary:
 - by the deployment administrator found on the server.
 - by the remote administrator, that can be used by the site manager.

5. If an analysis update was performed, select the **automatic modification of data files** during the setup.

Note: 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 is using the **Native AS/400 Access**. 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 for user N	Data directory for user N
Directory defined by the hosting company. In this directory, a sub-directory will be created for each site of user N.	Directory defined by the hosting company. In this directory, a sub-directory will be created for the data files of each site of user N.
The following elements are installed in this site sub-directory: HTML pages, the library (".WDL"), the image directory.	The following elements are installed in this data sub-directory: the ".FIC" files, the ".NDX" files, the ".MMO" files.
Example: The directory of sites for user N is "C:\UserN\WEBDEV sites". 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". User N installs his site named "Boat". By default, the data files will be installed in the "D:\UserN\WEBDEV data\Boat" directory.

3 Deployment by deployment package

3.1 The steps

The deployment steps are as follows:

1. Creating the deployment package from the editor: on the "Project" pane, in the "Generation" group, expand "Deploy the site" and select "Deploy the site remotely". In the wizard, select "Create a remote deployment package". A setup file is created.
2. Transmitting the deployment package (by Internet, CD, ...) to the administrator of Web server (hosting company for example)
3. Installing the WEBDEV application server at the hosting company (optional step). See the documentation about the WEBDEV application server for more details.

4. WEBDEV account (in the administrator) and FTP account (in the FTP server) created by the hosting company.
5. Running the deployment package on a computer and installing the WEBDEV site.
6. On-lining the WEBDEV site.

3.2 Creating the setup

You must define in the wizard:

1. **The files to install.** By default, the wizard selects the library, the data files, the images and the HTML pages.

2. If an analysis update was performed, select the **automatic modification of data files** during the setup.

Note: 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.

3. **Configuring the site.** These options are used to configure the options for connecting to the site as soon as it is installed.

These parameters can be modified if necessary:

- by the deployment administrator found on the server.
- by the remote administrator, that can be used by the site manager.

4. Specify whether your site is using the **Native AS/400 Access**. If you are using a license for a limited number of computers, the number of current connections must be managed in the WEBDEV site (INI file or registry, ...).

5. **The default profile of Web server to use.** These parameters can be typed or modified by the person who will install the deployment package.

6. **Configuring the setup:** Choose 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 setup computer of 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 server ones. Make sure that they are valid.
- **Activating the site immediately after setup:**
The site will be available as soon as it is installed: you will be able to connect to it by using the relevant 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 Web server on which the WEBDEV site will be installed (characteristics of FTP account of server and characteristics of WEBDEV account of server).
3. Validate the site setup.

4 Deployment by physical media

This deployment is performed in three steps:

1. Creating the setup version from the editor: on the "Project" pane, in the "Generation" group, expand "Deploy the site" and select "Create a setup by physical media".
2. Installing (if necessary) the WEBDEV application server on the Web server. See the documentation about the WEBDEV application server for more details.
3. Installing the site at the hosting company.

4.1 Creating the setup

You must define in the wizard:

1. **A title and a version name** for your setup. A title can be typed in each language that can be selected during the setup.

2. **A default directory.**

3. **A layout** for the setup window. The layout can be customized with the "Customize" button.

4. **The files of your site.**

Note: the library, HTML, Java, JavaScript and image files are selected by default.

5. **The appendix files:**

- The license (file in txt format) will be displayed when starting the setup.
Note: the setup will be canceled if the license is not validated by the person in charge of setup.
- The **"Readme"** file: this file can correspond to different formats (txt, htm, doc, hlp, pdf, ...). Regardless of the selected format, the user must have a software allowing him to view this file.

- **An executable** that will be run at the end of setup. **Caution:** Don't forget to include the files linked to this executable (DLLs, parameter files, other executables, ...).

Note: a license file and a "Readme" file can be selected for each setup language.

6. If an analysis update was performed, select the **automatic modification of data files** during the setup.

Note: if this option is selected, the update will be proposed during the setup.

7. If programs not developed with WEBDEV must access the HFSQL files of your site, choose to install an ODBC driver.

Note: if this option is selected, installing the driver will be proposed during the setup.

8. Specify whether your site is using the **Native AS/400 Access**.

If you are using a license for a limited number of computers, the number of current connections must be managed in the WEBDEV site (INI file or registry, ...).

9. Print the **deployment documentation** :

The **deployment documentation** specifies the WEBDEV license as well as all the modules installed during the deployment. It also specifies the registry keys generated for the WEBDEV site. When the site is installed on the server, these keys can be updated in the registry or saved into a file for further examination.

The deployment documentation must be supplied to the person in charge of installing your WEBDEV site (hosting company or intranet manager).

10. Modify (if necessary) the **generation directory**. By default, the setup is generated in the "INST" sub-directory of your project.

11. Generate the setup.

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. Once the setup is ended, WEBDEV proposes to:

- run the test of created 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 your site is installed at a hosting company, this one can propose two possibilities:

- **Install your site on a "shared" server.** This server manages the sites of several customers. This server is configured when installing the WEBDEV application server. The WEBDEV administrator (WD230ADMIN.EXE) is used to configure each WEBDEV site installed on this server. **Caution:** The memory resources of server are shared by all sites.
- **Install your site on a dedicated server** (recommended solution). You are the owner of 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.

Note: 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** (supplied with the setup of 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 2 to 7 (Microsoft Internet Information Server):** Microsoft server.
- **Apache Version 1.3.x and 2.x:** free Web server supplied by the Apache Group.

Three cases may occur:

- **Case 1: your Web server is not listed:** select "Other server". See the online help for more details about the server configuration.
- **Case 2: your Web server is displayed in the list but it is grayed:** your Web server was not detected yet. It can be selected by checking "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 must 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> specifies the root directory of Web server of computer. According to the selected Web server, the root directory is detected and displayed. If the root directory of Web server is not displayed, it can be typed in the gray area or it can be found via the "..." button.

Caution: if the area is filled, we advise you not to modify it.

6. **Update the data files** (optional step, for a site update only).

To update the data files associated with the WEBDEV site, choose to run the automatic modification of data files. Contact the site provider if you don't know whether this update must be performed or not.

Caution: The site 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 proposed address to run the test of your site locally. This option can be used only if the Web server was automatically configured.
- Click "Copy" to retrieve the proposed address. See "On-lining the dynamic site", page 220 for more details.
- Click "Done" to end the setup.

10. **Configuring the Web server:**

Two cases may occur if the Web server was not automatically configured:

- **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: see the online help for more details.

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.

Run the test of WEBDEV site from the administrator via the [Test page] option ("Advanced" tab).

Notes:

- The WEBDEV administrator (WD230ADMIN.EXE) must be started and run in background task. It can be installed as service in Windows NT.
- See "The WEBDEV administrator in practice", page 224 for more details about the WEBDEV administrator.

Important: The resource required per connected Web user

For each connected Web user, you must plan for:

- about 400 KB of RAM per connection, in addition to the memory required by the server.
- about 1 MB of disk space (virtual memory) per connection, in addition to the space required by the site.

For example, for 20 parallel connections, you must plan for:

- RAM: 67.8 MB (7.8 MB of RAM for this site, 60 MB for the NT server).
- Virtual memory: 84 MB (20 MB for the connections, 64 MB for the NT server).

5 On-lining the dynamic site

To allow the Web users to access your site installed at a hosting company, create a link in your home page allowing to start your dynamic WEBDEV site.

If the setup was performed by physical media, the address to use was supplied at the end of setup. Replace <localhost> by the address of your site.

Example used to start an "OURAPP" site from the PC SOFT site:

```
http://www.pcsoft.fr/WD230AWP/  
WD230AWP.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 toward the WEBDEV site, which means toward the address for example:

```
"http://205.51.231.57/WD230AWP/  
WD230AWP.EXE/CONNECT/  
prettyeyes"
```

Example of META tag that must be included in the <HEAD> section of HTML page:

```
<meta http-equiv="refresh" content="0  
;URI=http://205.51.231.57/WD230AWP/  
WD230AWP.EXE.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>/  
<ProjectName>_WEB/
```

PART 8

Hosting WEBDEV sites

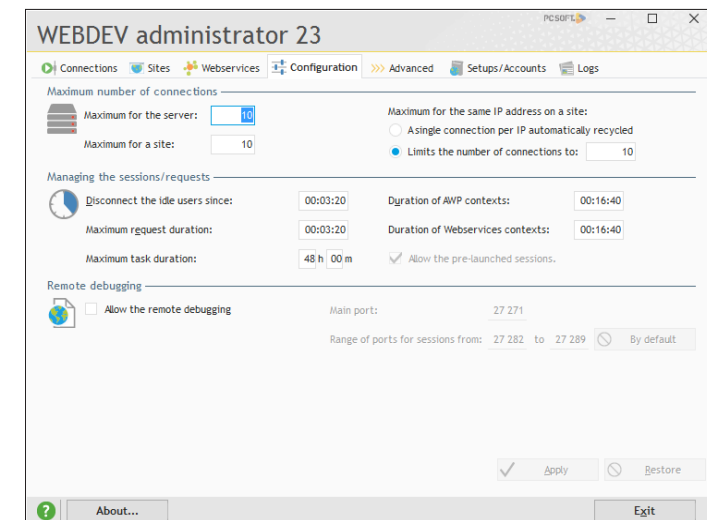


What is the WEBDEV administrator used for?

In development (for test) and in deployment, the WEBDEV administrator is required to run dynamic WEBDEV sites.

When developing a dynamic WEBDEV site, the administrator is mainly used to:

- run the test of dynamic sites via the test page ("Advanced" tab, "Test page" button)
- manage the webservices ("Webservices" tab)
- define the parameters for connecting to the dynamic sites for test: 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 the remote debugging ("Configuration" tab).
- ...



In deployment, the WEBDEV administrator is used to:

- manage the WEBDEV accounts ("Setups/Accounts" tab).
- manage the configuration of the different dynamic sites installed on the server: number of authorized connections, ... ("Sites" tab).
- customize the error messages displayed in the different dynamic sites installed on the server ("Advanced" tab).
- ...

The WEBDEV administrator in practice

1 Overview

The WEBDEV administrator is a module used to manage the dynamic WEBDEV sites.

When developing a dynamic WEBDEV site, the administrator is mainly used to:

- run the site test via the test page.
- define the parameters for connecting to the site for tests: disconnection time, ...
- delete the current test connections .
- change the Web server used for tests.
- perform a diagnostic if a problem occurs when starting a WEBDEV site.
- ...

Two types of WEBDEV administrators are available in deployment:

- **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 his dynamic WEBDEV sites remotely from any computer.

2 Interface of WEBDEV administrator

The WEBDEV administrator includes a window containing several tabs:

- "Connections" tab.
- "Sites" tab and "Webservices" tab.
- "Configuration" tab.
- "Advanced" tab.
- "Setups/Accounts" tab.
- "Logs" tab.

This chapter presents the WEBDEV Development administrator only.

For more details about the deployment tools, see their specific help.

Starting the WEBDEV administrator

To start the WEBDEV Development administrator:

- from WEBDEV Development version: on the "Tools" pane, in the "Web utilities" group, click "WAdmin".
- select "Programs .. WEBDEV 23 .. WEBDEV administrator" from the "Start" menu.

The WEBDEV administrator is automatically started during the first site test.

Then, the administrator can be run in background task.

By default, the WEBDEV administrator allows ten simultaneous connections to the same dynamic WEBDEV site (GO icon).

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 **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:
 - the connection identifier,
 - the site affected by the connection,
 - the identity of connected client. At run time, the client is identified by its IP address or by its Internet address,
 - the current connection time,
 - the idle time of connection.

- **Automatic refresh:** this option is used to automatically refresh the displayed data.

The "Connections" tab is also used to:

- 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 Sites installed

The "Sites" tab gives information about the dynamic WEBDEV sites installed on the computer.

The "Sites" tab returns the list of dynamic WEBDEV sites installed on the computer and managed by the WEBDEV administrator. For each site, the following information is displayed in a table:

- the site name.
- the full setup directory of site on the computer.
- the name of corresponding "project" file.
- the site status (locked or not). You have the ability to check the box in order to directly lock the access to the site during the update.
- the number of scheduled tasks used by the site.

The "Sites" tab is also used to:

- **Delete** the selected site from the table. It consists in deleting the site from the administrator only: the WEBDEV site will still be found on disk.
- **Refresh** the display 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 one of the sites found on the computer. By default, the configuration taken into account is the one of "Configuration" tab. You also have the ability to 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 will be locked for update. See the "Configuration" tab for more details about these parameters. To validate the new setting, press the "Apply" button. You also have the ability to manage the scheduled tasks associated with the selected site ("Scheduled tasks" button). This setting is available when the site is deployed.

- Run the test of sites installed on the computer (in development version only): Select the site and click "Run the test".

2.3 Webservices installed

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 webservice name.
- the full setup directory of webservice on the computer.
- the name of corresponding "project" file.
- the webservice status (locked or not). You have the ability to check the box to lock the access to the webservice during its update.
- the number of scheduled tasks used by the webservice.

The "Webservices" tab is also used to:

- **Delete** the selected webservice from the table. It consists in deleting the webservice from the administrator only: the WEBDEV Webservice will still be found on disk.
- **Refresh** the display of 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** each one of the Webservices found on the computer. By default, the configuration taken into account is the one of "Configuration" tab. You also have the ability to 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 will be locked for update. See the "Configuration" tab for more details about these parameters. To validate the new setting, press the "Apply" button. You also have the ability to manage the scheduled tasks associated with the selected Webservice ("Scheduled tasks" button). This setting is available when the Webservice is deployed.

2.4 General configuration

The general configuration corresponds to the default connection options for the WEBDEV sites found on the server and managed by the administrator.

The "Configuration" tab is used to:

- Configure the connections.
- Configure the sessions and requests.
- Manage the log file to get the traffic statistics.
- Manage the remote debugging.

The connections can be configured via the following options:

- **Maximum number of connections on the server:** Maximum number of simultaneous connections allowed for all the sites managed by the administrator (a connection = a Web user).
- **Maximum number of connections on a site:** Maximum number of simultaneous connections allowed for each site managed by the administrator. This number can be modified for each site in the "Sites" tab.
- **Maximum number of connections for a site user:** Maximum number of simultaneous connections allowed for a given Web user to each site managed by the administrator. If this parameter is equal to zero, any Web user who is trying to simultaneously connect to the same site will be automatically disconnected then reconnected. If this parameter is greater than zero, an error message will be automatically displayed after the x simultaneous connections. This number can be modified for each site ("Sites" tab). The option "A single connection automatically recycled" is used to limit the user connection to a single connection. The same connection will be re-used.

The sessions and requests can be configured via the following options:

- **Disconnect the users who have been inactive since:** This is the maximum connection time allowed without any action performed by the Web user. If 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 duration of AWP contexts. As soon as the specified duration is over and if no new request

was performed, the context file is deleted.

- **Duration of webservice contexts:** Validity duration of contexts corresponding to webservices.
- **Maximum request duration:** Maximum time-out allowed between the beginning of an action performed by the Web user (via a button or a link) and the display of 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 WEBDEV scheduled task or delayed task. The task will be automatically stopped when this duration is exceeded.
- **Allow the cached sessions:** Used to manage the cached 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. See the online help for more details.

The WEBDEV administrator is used to manage a specific log file allowing you to monitor the traffic statistics.

- **Generate a log file for the traffic statistics (.log):** Allows you to generate a log file (".LOG" extension) used to view the traffic statistics of WEBDEV sites via WDStatistic. See the help about WDStatistic for more details.
- **Save in the global file and in the file of each application:** This option is used to save the statistics both in the log file of application and in the log file of server. Indeed, if the log file is defined both at server level and at application level ("Sites" tab), information may be "lost".

Note: If the log file is enabled, the log of errors can be viewed in the "Logs" tab of administrator.

Manage 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.

2.5 Advanced options

The "Advanced" tab is used to:

- Choose one of the Web servers installed on the current computer to run the WEBDEV sites.
- Perform a diagnostic regarding the configuration of current computer.
- Specify the name or IP address of current computer.
- Display in the browser a page used to start all the WEBDEV sites installed on the current computer.
- Manage the search for expired pages.
- Manage the prints.
- Manage the error messages.
- Manage the emails in asynchronous mode.
- Manage the server sockets.
- Allow (or not) some specific functions.
- Forbid the change of IP while navigating.
- Forbid the access to AWP context identifiers from Javascript.

Server used

The "Server" button is used to choose one of the Web servers installed on the current computer to run the WEBDEV sites. Caution: This server will be used for all the dynamic WEBDEV sites installed on this computer.

If the server used is not found in this list, choose "Other". You will have to manually configure the Web server used. See the online help for more details about the configuration of specific servers.

Diagnostic

The "Diagnostic" button is used to check the configuration of current computer. This diagnostic is used to check:

- whether the FTP/IP protocol is installed
- whether a Web server (HTTP server) is started
- whether the 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. See the online help for more details.

Name or IP address of current computer

The WEBDEV administrator allows you to specify a name or an IP address to identify a specific computer. You also have the ability to specify the port number. For example: localhost:8080.

This computer will be used:

- when clicking the "Diagnostic" button.
- when clicking the "Test page" button.
- when running the test of a site, page, report, ... from the WEBDEV editor ("GO" icon).

Test

The "Test page" button is used to display in the browser a page allowing you to start each one of the dynamic WEBDEV sites and dynamic WEBDEV webservices installed on this computer.

See "Test of a site in practice", page 198 for more details about the tests of a WEBDEV site.

Finding the expired pages

This option is used to enable and configure the search for expired pages on the WEBDEV sites managed by the WEBDEV application server.

Printer used by default (Intranet sites only)

The "Prints" 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.

Error message

The "Errors" button displays the different error messages that can be displayed in the browser of Web user. You have the ability to customize the error message and the HTML page where the error message is displayed.

Email spooler

If "Disable the email spooler" is unchecked, your sites will be able to send emails without locking the execution of other processes (asynchronous mode). The asynchronous mode will have to be enabled when starting the email session (via **EmailStartSMTPSession** or **EmailStartSession**).

If the asynchronous mode is enabled, all outgoing emails will be redirected to a "Spooler". The emails are queued up before they are sent. The execution of Email functions do not lock the rest of your program anymore. **EmailStatus** is used to find out the email status.

Note: If the WEBDEV administrator is closed, the email spooler is cleared: the pending emails are not sent and they are removed from the spooler. If "Disable the email spooler" is checked while emails are still found in the spooler, these emails will not be lost: the administrator keeps sending them but no new email is accepted by the spooler.

Caution: The asynchronous mode can only be used when starting a session on an SMTP server (*EmailStartSMTPSession* for sending emails *EmailStartSession*). This mode is ignored in the other cases.

Sockets

If the option "Allow the server sockets" is checked, your sites will be able to handle the server sockets (via the Socketxxx functions of WLanguage).

Forbid the change of IP while navigating

If this option is checked, the IP address associated with the session cannot change while navigating. This is used to protect against "session hijack" attacks (attack that consists in pretending to be a legitimate user connected to the server).

Forbid the access to thenAWP 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 the option "Forbid the access to the AWP context identifiers from JavaScript" is checked, the cookies used will be "HTTPOnly" cookies, that 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.

Manage the fCopyFileWebFolder and fDeleteFileWebFolder functions

The option "Allow fCopyFileWebFolder and fDeleteFileWebFolder" must be checked if these functions are used in the Web site.

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 Web users of the site for example).

Caution: The copy is taken into account by the WEBDEV administrator on the server (WD230ADMIN.EXE). The Windows account that runs it must have sufficient rights on the target location of copy.

2.6 Setups/Accounts

The "Setups/Accounts" tab is used to:

- Configure the server for the site setups and updates.
- Manage the log of setups.
- Manage the WEBDEV accounts.

Setup/Update

These options are accessible in deployment version. They allow the hosting company to authorize and configure the site setups and updates remotely.

The option "**Lock the server (during an update)**" simplifies the site update by forbidding any connection from a new user. Whenever attempting to connect to one of the sites found on the server, the Web user receives a message signaling that the site is temporarily unavailable and asking him to retry later. The Web users who are already connected can continue to use the current WEBDEV site.

2.7 Logs

The "Logs" tab is used to see the details of logs over a specific period.

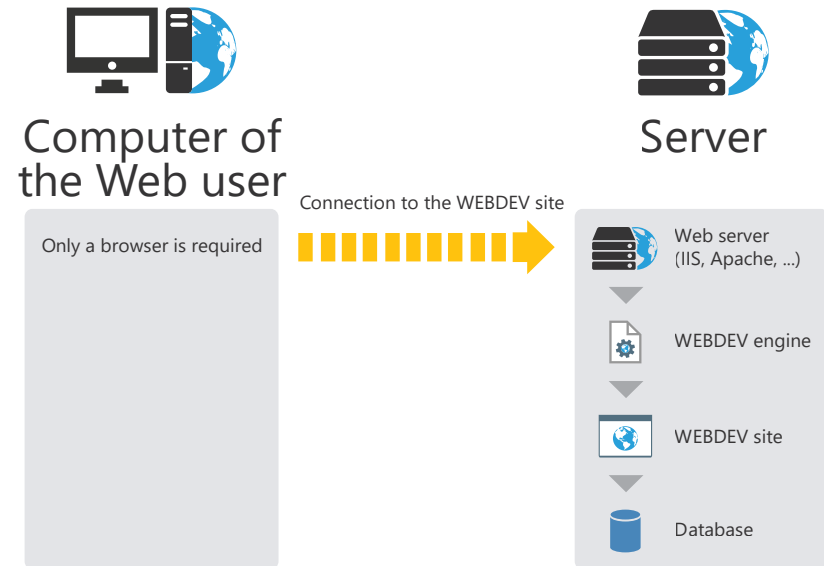
Caution: To use this feature, the logs must have been enabled for the site. The logs can be enabled:

- in the "Configuration" tab to manage a general log,
- in the "Sites" tab ("Parameters" button) and in the "Webservices" tab ("Parameters" button) to manage a specific log.

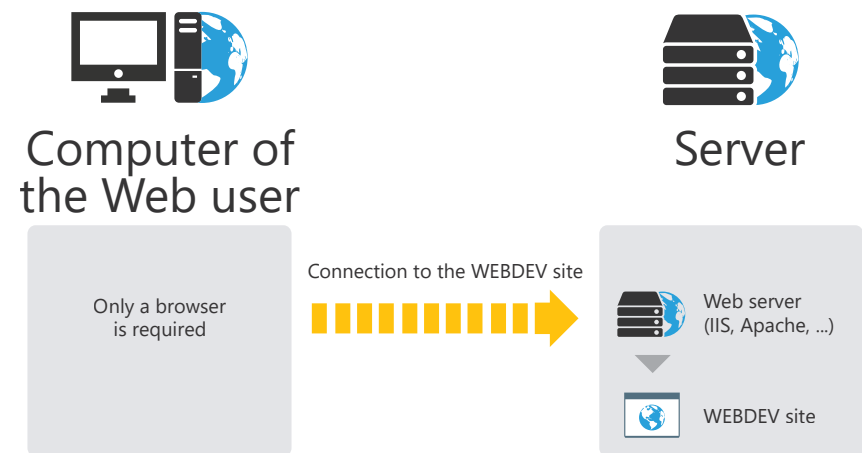
The log of errors is automatically created when the log of statistics is enabled.

Server configuration

For a dynamic WEBDEV site:



For a static or semi-dynamic site:



Dedicated server or shared server?

The hosting companies propose two types of servers for installing your WEBDEV sites:

- **shared server:**
server whose resources are shared by several clients (client = hosted company)



Web server

Common part

- WEBDEV application server
- Operating system
- Registry
- Web and FTP server

Disk space
Client 1

Disk space
Client 3

Disk space
Client 2

Disk space
Client 4

Characteristics:

The hosting company provides a server to several clients.

The hosting company:

- configures the FTP server.
- defines the maximum number of connections for all the dynamic sites belonging to the same client (via the WEBDEV account manager).

The setup and updates of a dynamic or static site can be performed by FTP.

- **dedicated server:**
the server reserved for the client. The hosting company provides the access to Internet (IP address, wire, ...)The computer is either rented by the hosting company, or provided by the client.



Web server

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, ...).

The client must:

- provide the server (or rent it)
- configure the computer (system, ...)
- install the Web server
- install and configure the WEBDEV sites (by FTP or by CD-ROM)

Which type of server to choose?

The table below presents the benefits and drawbacks of different types of servers. Regardless of the server type, a deployment license is required for each server. Each server is hosting one or more dynamic WEBDEV sites.

Server	Benefits/Drawbacks	When should this server be used?
Shared	<p>Main benefits:</p> <ul style="list-style-type: none"> - Lower cost - Server managed by the hosting company <p>Main drawbacks:</p> <ul style="list-style-type: none"> - Memory resources shared by all sites found on the server. - If a site is locked, all the sites found on the server are locked. - Stand-alone executables can rarely be used. - 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. 	<p>A shared server is recommended for:</p> <ul style="list-style-type: none"> - the sites with few hits (up to 20 or 30 simultaneous connections), - the sites requiring little memory (no large calculations performed on the server), - the sites in startup phase.
Dedicated	<p>Main benefits:</p> <ul style="list-style-type: none"> - The server resources are available for the sites installed on this computer. - Custom management of data download: <ul style="list-style-type: none"> - replication by emails, - WDREPLIC, - copying files by FTP. <p>Main drawbacks:</p> <ul style="list-style-type: none"> - High cost - Server entirely managed by the client in most cases. 	<p>A dedicated server is recommended for:</p> <ul style="list-style-type: none"> - sites with many hits, - the sites requiring a lot of disk space and memory, - the sites managing sensitive data, - the dynamic sites.

Hosting Control Center

Mainly intended for the hosting companies and for the Webmasters, the Hosting Control Center allows you to easily host the 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.

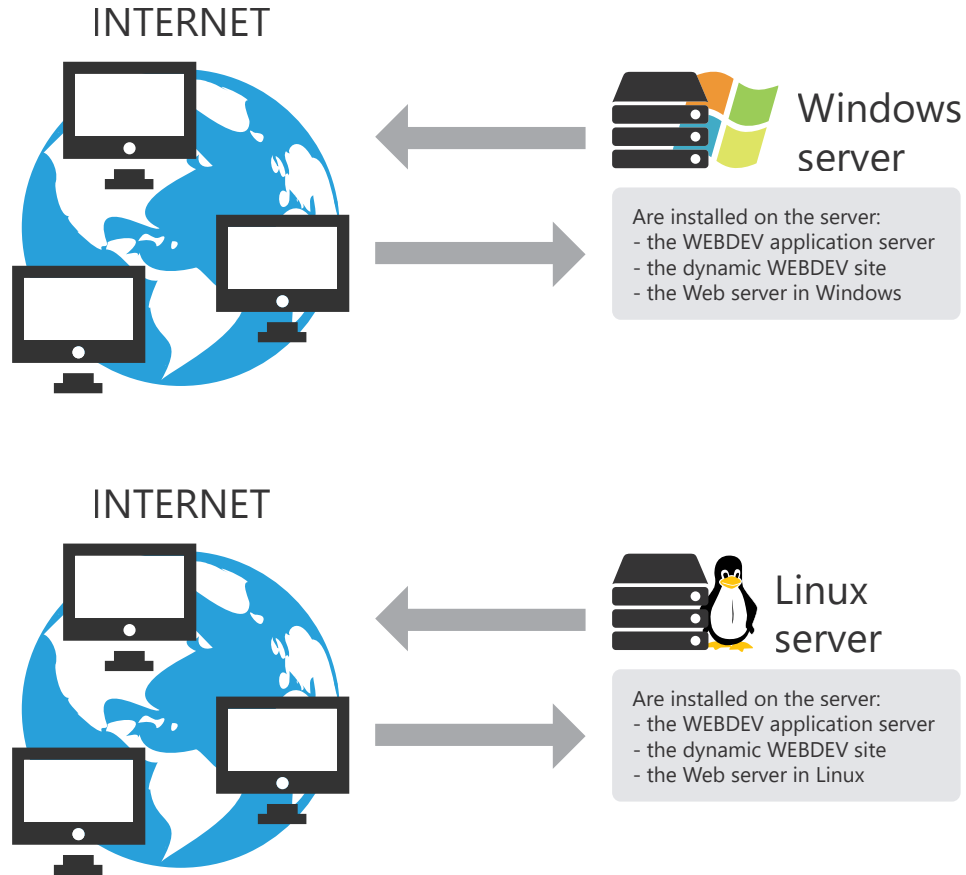
The setting of following elements is centralized:

- Windows accounts,
- Groups of FTP users,
- Groups of WEBDEV clients,
- Home Directory,
- FTP alias,
- WEBDEV accounts,
- Data directories,
- Virtual Web sites, ...

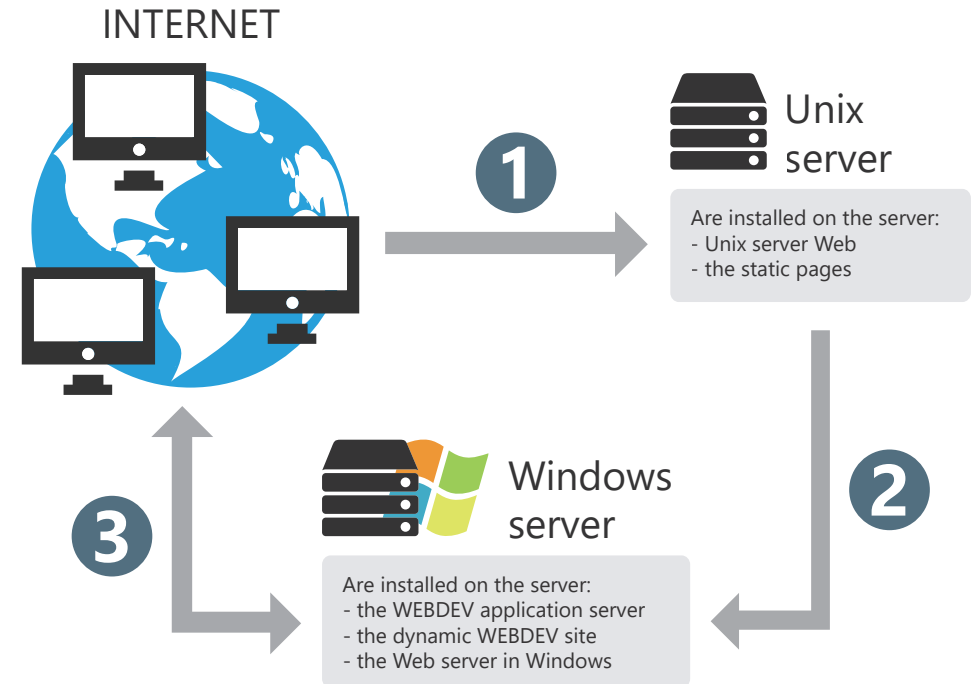
The WEBDEV Hosting Control Center operates in the Windows environment with the IIS Web server (Internet Information Server, all versions from version 5).

Dynamic site on specific configurations

1. Setup on a Web server in Windows (2000 or later) or a Linux server



2. Setup on a Windows server, with an access to Internet by UNIX

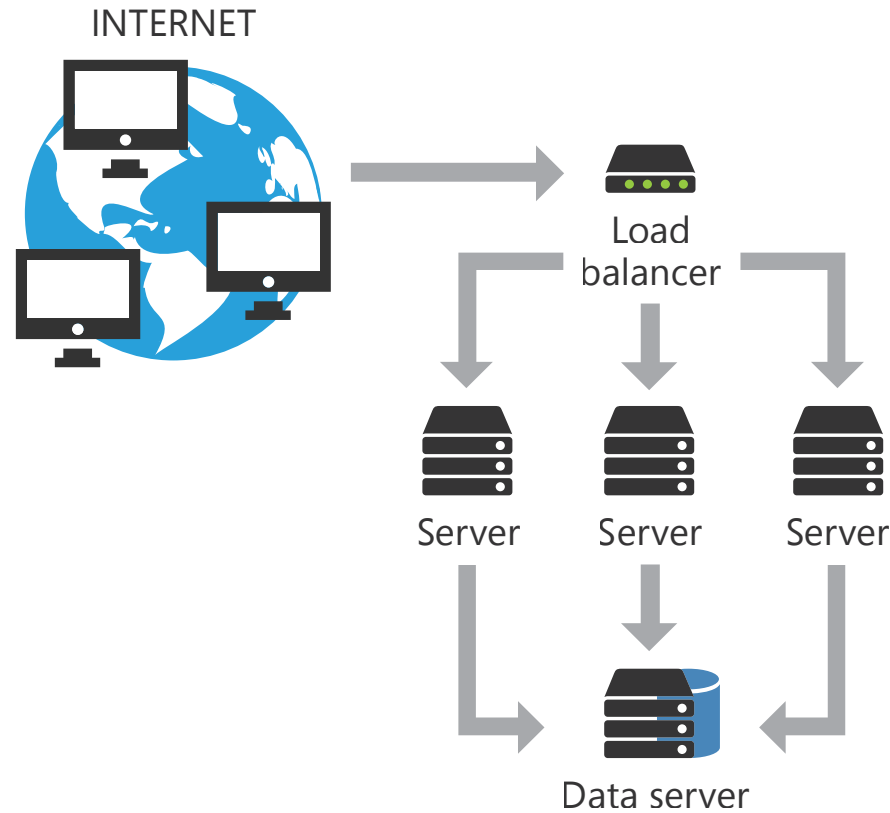


1. Opening a static page. The static page is found on the UNIX server.
2. Starting the dynamic WEBDEV site (found on the Windows server) via a link found in the static HTML page.
3. Standard operating mode of WEBDEV site.

Note: the same configuration can be performed 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 dynamic WEBDEV sites on the data server.

Traffic statistics of dynamic sites

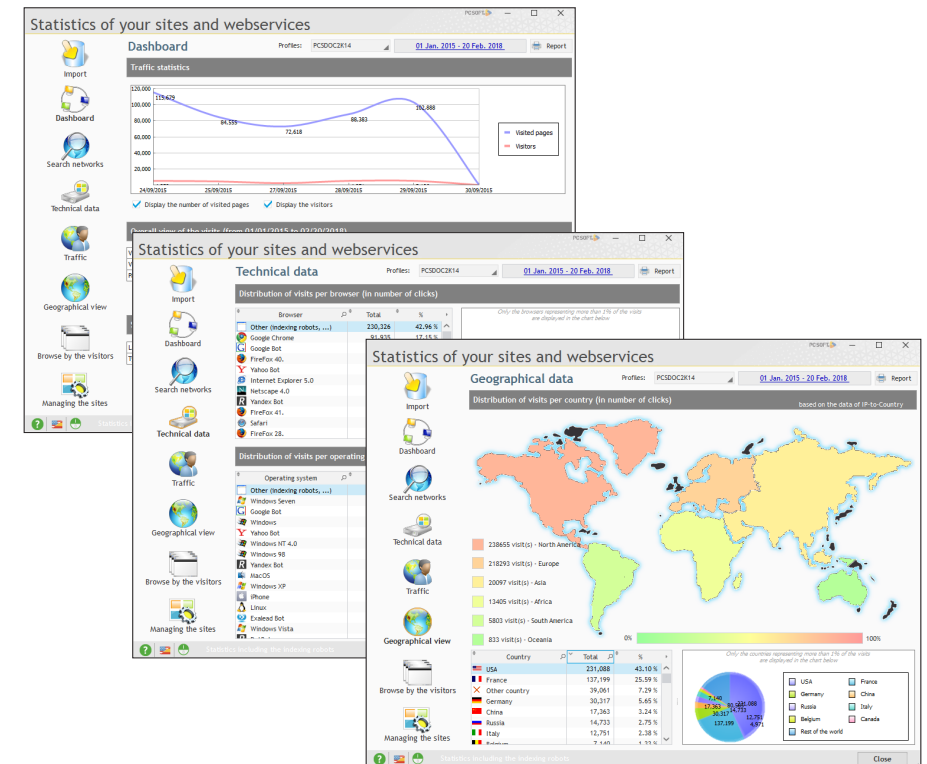
It is often important to get accurate statistics about the site visits.

How do you find out who visited your site, which pages have been view, which applications have been use and from which site were they coming from?

WEBDEV is supplied with a statistical tool for the dynamic pages: WStatistic. This tool can be installed on any computer.

Examples of statistics calculated by WStatistic: number of connections per day, visited pages, operating systems and browsers of Web users, ...

Examples of statistics for the dynamic sites:

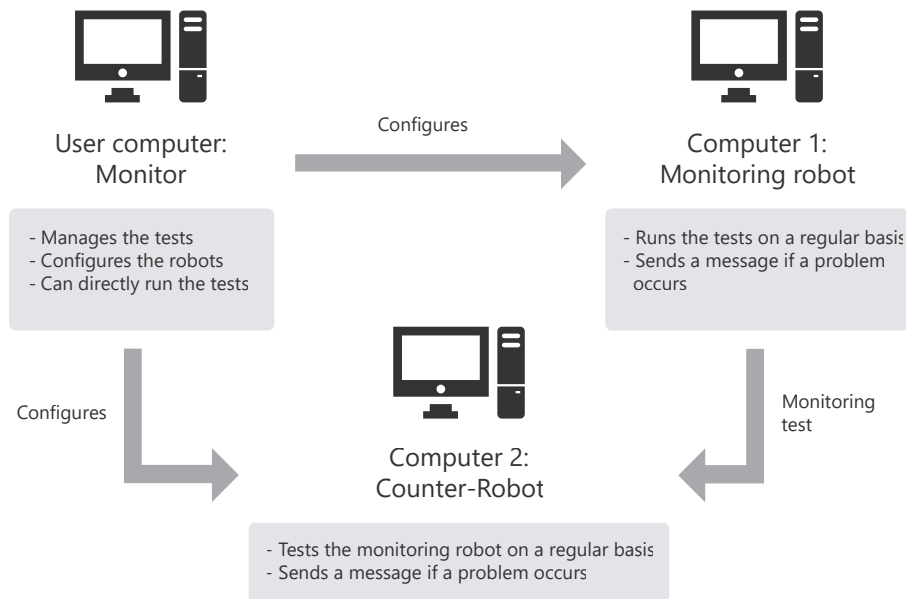


Monitor your sites, servers, ...








A site depends on several external parameters: Web site, HFSQL server, network, ... To optimize the management of incidents, WEBDEV proposes to use a monitoring robot. Made of three executables started on different computers, the monitoring robot is used to run different tests: Internet tests, network tests, ... If problems occur when running a test, different methods can be used by the monitoring robot to inform you:

- Message sent into PC SOFT messaging (WDBAL).
- Email
- Execution of another program.

Furthermore, a sound alert can be implemented on the monitor.



Available types of tests

 Internet test (test of a Web site)	 Test of 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 23

(Details in the WLanguage online help)

AddFavorite	Adds an Internet address into the list of favorites for the Web user.
AJAXAsynchronousCallPending	Used to find out whether a server procedure called by AJAXExecuteAsynchronous is currently run.
AJAXAvailable	Used to find out whether the AJAX technology is supported by the current browser.
AJAXCancelAsynchronousCall	Cancel the automatic execution of browser procedure called by AJAXExecuteAsynchronous.
AJAXExecuteAsynchronous	Runs a server procedure without refreshing the page.
AJAXExecute	Runs a server procedure without refreshing the page.
AppServerAddScheduledTask	Adds a new scheduled task to the application.
AppServerDeleteScheduledTask	Deletes a scheduled task.
AppServerInfoScheduledTask	Returns information about a scheduled task.
AppServerListScheduledTask	Returns the list of scheduled tasks for the application.
AppServerModifyScheduledTask	Modifies an existing scheduled task.
AppServerRunScheduledTask	Immediately starts the execution of scheduled task.
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.
AssistedInputConfigure	Used to configure the filter and the opening of assisted input.
BannerFirst	Displays the first plane of Scrolling Banner control.
BannerLast	Displays the last plane of a Scrolling Banner control.
BannerNext	Displays the next plane of Scrolling Banner control.
BannerPrevious	Displays the previous plane of Scrolling Banner control.
BannerStartScrolling	Enables the automatic scroll on a Scrolling Banner control.
BannerStopScrolling	Stops the automatic scroll on a Scrolling Banner control.
BrowserBlade	Returns the number of the blade displayed in a page in Responsive Web Design mode.

BrowserClose	Closes the current browser window and stops the execution of current code.
BrowserHeight	Returns the height (in pixels) of the client area where the page is displayed.
BrowserIPAddress	Returns the IP address of the computer of Web user connected to the WEBDEV site.
BrowserIsConnected	Indicates whether the browser is connected to the net
BrowserMobile	Indicates whether the browser used by the Web user is found on a mobile device.
BrowserName	Returns the name of browser used by the Web user.
BrowserOpen	Opens a new browser window.
BrowserOS	Indicates the operating system announced by the browser of Web user.
BrowserPlatform	Returns the platform of browser used by the Web user.
BrowserRunApp	Starts the native Web browser of application
BrowserType	Returns the type of browser used by the Web user.
BrowserWidth	Returns the width (in pixels) of client area where the page is displayed.
CancelAWPContext	Deletes from the AWP context a variable added by DeclareAWPContext.
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 displayed via CellDisplayDialog.
CellDisplayDialog	Displays a cell in a page with a DDW effect (Dim Disabled Windows).
CertificateClientInfo	Returns information about the certificate used by the client computer.
ChangeAction	Used to specify the action to perform when the HTML page displayed in the browser is no longer synchronized with the page context on server.
ChangeTarget	Changes the target frame of the page.
ColorPalette	Reads a color found in the current palette.

ConfigureAWPContext	Defines the operating mode of AWP contexts.
ConnectionCount	Returns the number of instances of the WEBDEV site currently run on the server.
ContextClose	Closes a page context.
ContextExist	Used to find out whether a page context exists on the server.
ContextOpen	Opens a new page context without returning the information to the browser.
CookieRead	Retrieves the value of a cookie saved on the computer of Web user.
CookieWrite	Writes a cookie on the computer of the Web user during the next display of a WEBDEV page in the browser
CurrentPage	Returns the name of the page containing the WLanguage code currently run.
DeclareAWPContext	Declares a list of variables whose value will be persistent between the successive displays of AWP page
DeleteAWPContext	Deletes from the AWP context a variable added by DeclareAWPContext
DynamicServingConfigure	Indicates the environment that will be used by the “Dynamic Serving” to choose the set of pages to display
DynamicSiteDisplay	Displays a dynamic site (created by WEBDEV) in the browser of Web user from a dynamic or static WEBDEV page.
EmailOpenMail	Opens the default messaging software of the Web user.
EmailStatus	Returns the status of an email sent via an SMTP session 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	Delete an image file from the “_WEB” directory of application (or from one of its sub-directories).
FileDisplay	Returns a specific file to the browser.

FileToPage	Automatically initializes the controls of a page with the values of associated items found in the current record (loaded in memory) of HFSQL file.
FolderData	Automatically initializes the memory value of items found in a data file with the value of page controls and/or the value of WLanguage variables with the value of page controls.
FolderWeb	Returns the path of directory containing the images, the Javascript files, the Java applet files and the other files accessible from the browser, ...
FramesetDisplay	Displays a WEBDEV frameset in the browser of Web user.
FramesetRefresh	Refreshes a frameset displayed in the browser of the Web user from the context found on the server.
FramesetUse	Displays a WEBDEV frameset in the browser of Web user and closes all 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. The toolbar is not displayed anymore.
fWebDir	Returns the physical name of directory containing the images, the Javascript files and the Java files of the WEBDEV site.
GadgetCloseFlyout	Closes the popup area of Vista gadget.
GadgetDisplayFlyout	Displays a page of Vista gadget in a popup area.
GadgetLoadParameter	Loads a persistent value in a Vista gadget.
GadgetSaveParameter	Saves a persistent value in a Vista gadget.
GaugeActivate	Enables the refresh of Ajax indicator.
GaugeDeactivate	Stops the periodic refresh of Ajax indicator.
GaugeExecute	Starts a long browser process and makes an Ajax indicator move forward according to the progress of this process.
GglAnalyticsAddEvent	Adds an event beside Google Analytics
GglAnalyticsAddException	Adds an exception beside Google Analytics
GglAnalyticsAddPage	Adds a page beside Google Analytics

GglAnalyticsAddSocialNetworkAction	Adds a social network action beside Google Analytics
GglAnalyticsAddTiming	Adds a duration beside Google Analytics
gpwOpenConfiguration	Opens the page for configuring the user groupware.
grlImageSize	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 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 of generated images when printing in HTML format.
ImageArea	Returns the number of the image area clicked by the Web user (clickable image only).
ImageFirst	Displays the first image.
ImageLast	Displays the last image.
ImageNext	Displays the next image.
ImageOccurrence	Used to find out the total number of images.
ImagePrevious	Displays the previous image.
ImageScrollingPosition	Returns the image displayed.
ImageStartScrolling	Starts the automatic scroll of images.
ImageStopScrolling	Stops the automatic scroll of images.
ImageXPos	Returns the horizontal position of mouse cursor in relation to the relevant image control.
ImageYPos	Returns the vertical position of mouse cursor in relation to the relevant image control.
jQueryExecute	Runs a Javascript method of jQuery library on a page element.
jQuery	Runs a Javascript method (or several chained methods) of jQuery library on a page element.
JSEndEvent	Deletes the association between a WLanguage browser function and an event.
JSEvent	Associates a browser procedure with an event on an object in browser code.
JSInfoEvent	Used to handle the JavaScript properties of the browser event that triggered the code execution.
JSInterruptEvent	Interrupts the process of current event.

JSMethod	Used to run a Javascript method on an element found in the current page.
JSONExecuteExternal	Calls an external server URL that returns data in JSON format (JavaScript Object Notation). The data is processed in a specific procedure.
JSONExecute	Calls a server URL of the same domain that returns data in JSON format (JavaScript Object Notation).
JSPProperty	Used to handle specific features on the objects found in the current page.
LocalStorageAdd	Adds a value to the local storage.
LocalStorageAvailable	Indicates whether the local storage is available.
LocalStorageDeleteAll	Deletes all the values from the local storage.
LocalStorageDelete	Deletes a value 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 menu popup.
MenuAddURLOption	Adds a new menu option at the end of a menu. This menu option is used to display the page corresponding to the specified URL.
PageActivateDDW	Enables or disables the effect applied by WEBDEV to the inactive pages when displaying a modal page.
PageAddress	Used to get the Internet address of a WEBDEV page.
PageCloseDialog	Closes the current page. This page was opened by PageDisplayDialog.
PageDisplayDialog	Displays a page in modal mode. Used to establish a dialog with the user.
PageDisplay	Displays a page in the browser of Web user.
PageExist	Checks whether the specified page is currently displayed in the browser of Web user.
PageInitialization	Resets (or not) the controls found in the current page and runs the initialization processes of controls.
PageParameter	Returns the value of a parameter passed to the current page.
PagePosition	Scrolls a page up to position a control in the visible page section (top) in the browser.
PageRateDDW	Used to define the rate of gray for the inactive pages.

PageReadHeaderHTTP	Returns the HTTP header received by the current page
PageRefresh	Refreshes a page displayed in the browser of Web user from the context found on server.
PageSubmit	Validates the specified page and starts the execution
PageToASP	Sends the data found in a page currently displayed in the browser to an ASP server.
PageToEmail	Emails the data found in a page currently displayed in the browser.
PageToFile	Automatically initializes the memory value of file items with the value of page controls.
PageToJSP	Sends the data found in a page currently displayed in the browser to a JSP server.
PageToPHP	Sends the data found in a page currently displayed in the browser to a PHP server.
PageToSource	Automatically initializes the memory value of items found in a data file with the value of page controls and/or the value of WLanguage variables with the value of page controls.
PageUse	Displays a WEBDEV page in the browser of Web user and closes all current page contexts.
PageVisible	Indicates whether the page is visible to the user.
PageWriteHeaderHTTP	Adds an HTTP header. It will be sent during the next return to the browser.
PHPDisplay	Calls an external PHP script and returns the result page in the current browser window.
PHPExecute	Calls an external .php script and returns the result in a string. An HTTP request is performed.
PopupClose	Closes a popup page.
PopupDisplay	Displays a popup page.
PreviousPage	Returns the name of previous page.
rssDisplay	Builds a RSS stream and returns the content of RSS stream to the client. The result is displayed in the browser of Web user.
SaaSCheckService	Checks the access rights of the user to a service of SaaS site.
SaaSClientConnexion	Returns the characteristics of the connection to the client database.
SaaSConnectedUser	Returns the user who is currently connected via SaaSConnect.

SaaSConnect	Authenticates a user of a SaaS site beside the SaaS webservice that manages the site.
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 (.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.
SiteAddress	Returns the Internet address for connecting to a dynamic WEBDEV site found on the same server.
SiteMapPathAddLink	Adds a link into a site map path.
SiteMapPathDeleteAll	Deletes all the links from a site map path.
SiteMapPathDeleteLink	Deletes a link from a site map path.
SiteMapPathInsertLink	Inserts a link into a site map path.
SiteMapPathModifyLink	Modifies a link in a site map path.
SourceToPage	Automatically initializes the controls of a page with: - the values of associated items in the current record, - the values of associated WLanguage variables.
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.
UploadCurrentFileSizeSent	Returns the size (in bytes) already sent for the file currently uploaded via an Upload control.
UploadCurrentFileSize	Returns the size (in bytes) already sent for the file currently uploaded via an Upload control.

UploadCurrentFile	Indicates the file currently uploaded via the upload control. This function is useful for the multi-file Upload controls.
UploadDeleteAll	Clears the list of files to upload: no file will be uploaded on the server.
UploadDelete	Deletes a file from the list of files to upload: the file will not be uploaded on the server.
UploadFileName	Returns the name of a file “uploaded” by the Web user.
UploadFileSize	Returns the total size (in bytes) of a file found in an Upload control.
UploadSizeSent	Returns the total size (in bytes) of the files already sent by the current upload via an Upload control.
UploadSize	Returns the total size (in bytes) of the files already sent by the current upload via an Upload control.
UploadStart	Starts sending the selected files into an upload control. At the end of the upload, the list of files to send is cleared on the browser.

Examples and components supplied with WEBDEV

The examples supplied with WEBDEV are intended to help you learn the features of WEBDEV.

Their source code is presented in details.

Different types of examples are supplied 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 whose test can be run in the current project. This page presents the use of a function, group of functions, control, ...
- components: -these examples include an internal component, an external component and a use example.

These examples and components can be opened from the home window of WEBDEV.

- The complete examples, training examples and unit examples are found in the "Examples" sub-directories of setup directory of WEBDEV.
- The components are found in the "Components" sub-directory of setup directory of WEBDEV.

Additional examples are supplied with the Technical Support Newsletter (LST) or can be downloaded from our site (www.windev.com).

Complete examples

Example name	Description
eBusiness	Full eCommerce site that can be customized via a Web management interface. - The showroom/payment section of the site is developed in AWP in order to get the best possible referencing of products. - The administration section of the site is developed in standard WEBDEV session to guarantee the maximum security.
Honolulu	Free Intranet portal: messaging, forum, blogs, ...
Photo_Gallery	This example is a site of photo gallery
WAds	Management of classified ads. Multi-criteria search, addition, modification and deletion of classifieds, as well user management.
WebApp	This example is a Web application intended for the Smartphones and tablets ("WebApp"). Used to display a a list of products (with filter and search), to view and edit a product form. All server processes are performed in AJAX.
WebEstate	Site for selling and renting real estates.
WebFleet	Simulates the management of a computer fleet.
WebMillion	Uses a data file containing more than 1 million records.
WW_Association_AWP	Association site made of 2 main sections: - the "visitor" section, built in AWP mode therefore can be referenced. - the "member space" section, built with WEBDEV in standard mode and therefore secured. This site presents news, a photo gallery, the association overview, ...

WW_Association_PHP	Association site in PHP WEBDEV made of 2 main sections: - the visitor section, accessible to all. - the management section, reserved to the members of association. 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_Blog_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: - the amount of monthly payments for a given loan - the amount that can be borrowed for a given monthly payment - the income of an investment for a given monthly payment. 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_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.
WWDload	This site is used to propose files that can be downloaded by the users according to the group to which they belong.

Training examples

Example name	Description
WW_Auto_Error_Management	This example explains how to use the automatic management of WEBDEV errors.
WW_AutoZoom	This example displays an image miniature and when it is clicked, the image is enlarged without disrupting the page layout.
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_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_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	Explains how to draw in HTML5 and how to save the drawn image in browser code.
WW_WIKI	This example is a manager of WIKI documents. A WIKI document is a document that can be modified by all the persons who have access to this document.

WW_Wizard	This example includes a class and a page template that can be re-used to create and manage a Web wizard.
WWBrowserDialog	This example presents a dialog between two distinct browsers.
WWPocket	This example is a site that can be viewed on a Pocket PC. The site detects the resolution of the user and displays the pages consequently.

Components

WEBDEV is supplied with projects corresponding to components. These projects contain:

- a project configuration corresponding to the component.
- a project configuration corresponding to the use of component.
- an internal component.

Let's see the main components supplied with WEBDEV:

Component name	Description
CloudOfTags	Used display a cloud of tags in a dynamic WEBDEV application.
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.
PayBox	Used to install a system for secure payment via PayBox.
PayPal	Used to include PayPal payment buttons in your WEBDEV pages.
Secure payment	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 downloaded from our site (www.windev.com).

