In which order should the 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.
Organization of guide

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

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 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
- 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
- Site SEO ........................................................................................................................................... 79
Summary

Part 3 - Development Environment
The WEBDEV editors ........................................................................................................................................ 85
Project dashboard ......................................................................................................................................... 89
WEBDEV, WINDEV, WINDEV Mobile: 100% compatible ............................................................................ 90
Project configuration ..................................................................................................................................... 91
Multiple generation ....................................................................................................................................... 92
Source Code Manager (SCM) ...................................................................................................................... 93
Internal component ...................................................................................................................................... 100
External component ..................................................................................................................................... 103
Generation modes ........................................................................................................................................ 111
Project Monitoring Center ........................................................................................................................... 114
Managing requirements ............................................................................................................................... 115
Managing tasks ............................................................................................................................................ 116
Managing business rules .............................................................................................................................. 117

Part 4 - Advanced Concepts
RAD RID ....................................................................................................................................................... 121
The UML model .......................................................................................................................................... 122
User groupware .......................................................................................................................................... 128
Multilingual sites ........................................................................................................................................ 134
10 tips about ergonomics ............................................................................................................................ 138
Overlaying the controls ................................................................................................................................. 142
Customizing the aspect of a site: skin elements .......................................................................................... 143
Choosing a type of button/link .................................................................................................................. 145
Preventing from going back to a page .......................................................................................................... 146
Security benefit: managing the "Back" button .............................................................................................. 147
Cookies: information stored on the computer of Web user ....................................................................... 153
Protecting the access to the site: passwords ................................................................................................. 154
TLS/SSL: Encrypting information on the Web .............................................................................................. 155
Secure payment with provider .................................................................................................................. 156
Sending emails ............................................................................................................................................ 157

Part 5 - Databases
Analysis: Database structure ......................................................................................................................... 161
The various types of accessible files ............................................................................................................ 170
HFSQL Classic ............................................................................................................................................ 172
HFSQL Client/Server ................................................................................................................................ 173
HFSQL Client/Server clusters ...................................................................................................................... 174
HFSQL: the physically created files ............................................................................................................ 175
Associate controls with data ...................................................................................................................... 176

The queries .................................................................................................................................................. 178
The embedded queries ............................................................................................................................... 179
The Table/Looper control ............................................................................................................................ 180
Retrieving data from a site ............................................................................................................................ 182
Universal replication ................................................................................................................................... 187
3-tier architecture ....................................................................................................................................... 188

Part 6 - Running the Test of a Web Site
Running a site test: The elements to test .................................................................................................... 191
How to run a site test? .................................................................................................................................. 192
The debugger ............................................................................................................................................... 194
When can I use the debugger? ................................................................................................................... 195

Part 7 - Deploying a Web Site
Deploying a static site .................................................................................................................................. 205
Deploying a dynamic WEBDEV site ............................................................................................................ 207
On-lining a dynamic WEBDEV site ........................................................................................................... 211

Part 8 - Hosting WEBDEV Sites
What is the WEBDEV administrator used for ............................................................................................. 221
Server configuration ..................................................................................................................................... 227
Dedicated server or shared server? ............................................................................................................ 228
Which type of server to choose? .................................................................................................................. 230
Hosting Control Center ............................................................................................................................... 231
Dynamic site on specific configurations ..................................................................................................... 232
Traffic statistics of dynamic sites ............................................................................................................... 235
Monitor your sites, servers, ......................................................................................................................... 236

Part 9 - Appendices
WLanguage functions specific to WEBDEV 24 ........................................................................................ 239
Examples and components supplied with WEBDEV ................................................................................ 248
DeveloP 10 TIMES FASTER

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.

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

Server at the hosting company. The HTML pages are stored on this server.
Static or dynamic site?

Several types of sites can be created:
• static sites,
• dynamic sites.

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

<table>
<thead>
<tr>
<th>Static site</th>
<th>Dynamic site</th>
</tr>
</thead>
<tbody>
<tr>
<td>The content of pages is fixed, it is defined once and for all.</td>
<td>The data displayed in the pages changes. In most cases, the pages are used to:</td>
</tr>
<tr>
<td>A static site cannot interact with the data.</td>
<td>• perform processes and/or calculations by programming,</td>
</tr>
<tr>
<td></td>
<td>• display the data stored in a database.</td>
</tr>
<tr>
<td></td>
<td>• display images and interactive text.</td>
</tr>
</tbody>
</table>

WEBDEV offers several types of dynamic sites:
• WEBDEV site in Session mode.
• WEBDEV site in AWP mode.
• PHP site.

Static WEBDEV site: the WEBDEV Application Server is not required.

Dynamic WEBDEV site: the WEBDEV Application Server or the PHP language is required.

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

WEBDEV allows you to create:
• static 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 (in Session or AWP mode). This type of site requires a WEBDEV Application Server on the server. It can be used to handle several databases.
• dynamic PHP site. This type of site requires a PHP engine on the server. This type of site is recommended for the dynamic sites hosted by a public hosting company.
The different types of pages

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

- **Static pages.** This type of page is used to display fixed data.

- **WEBDEV dynamic pages in Session mode.** These pages are used to dynamically display the information found in a database. The content of the page depends on the displayed record. When displaying a dynamic page in session mode, the associated page context is automatically created on the server. This page context mainly contains the global variables and the variables for positioning in the data files.

- **Dynamic WEBDEV AWP pages (Active WEBDEV Page).** These pages are used to dynamically display the data found in a database. The content of the page depends on the displayed record. An AWP page is a dynamic WEBDEV page without a context persisting on the server. The AWP page context is temporary. It is created in a temporary session.

- **Dynamic PHP pages.** These pages are used to dynamically display the data found in a database. These pages can only be used in a PHP site.

The following pages present these different types of pages.
Principle of a dynamic page (Session or AWP)

Development

Create a dynamic page in WEBDEV (".WWH" file)

Server

HTML page  WEBDEV engine  Database

Computer of the Web user

Dynamic page in execution in the browser. The data displayed in the pages depends on the operations performed by the Web user.

DISPLAY

Principle of a PHP type page

Development

Create a PHP page in WEBDEV (".WWH" file)

Server

PHP page  MySQL database

Computer of the Web user

PHP page in execution in the browser. The data displayed in the pages depends on the operations performed by the Web user.

DISPLAY
Principle for displaying a dynamic WEBDEV site

The WEBDEV session

The following operations are performed when displaying a WEBDEV page in Session mode:
1. Request page display.
2. Start WEBDEV engine. This engine will remain on the server until the end of application.
3. Create the application context. This context will remain on the server until the end of 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 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 contexts of data files, ...

The page contexts remain in memory on the server until the end of the 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. You don’t need to program anything.
**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 WEBDEV site in Session mode, each displayed page owns a persistent page context for the entire lifetime of the session on the server.

**Operating mode of AWP sites**

The following elements are automatically created on the server when an 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 is sent to the Web user, the temporary page context and the temporary session are destroyed. There is nothing left in memory on the server.

The temporary AWP page context contains all 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 contexts of data files, ...

When the AWP page is 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
It is possible to pass information from one page to another via the URL. The URL has the following format: "http:\\Web server\\..\\mypage.awp?NameParam1=Value1&NameParam2=Value2".
This method allows a better page referencing 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 timeout of AWP contexts is not exceeded. The timeout of AWP contexts is defined in the WEBDEV administrator ("Configuration" tab, "Timeout 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 cart, characteristics of a product, list of cities, map, etc.) are modified, only these elements will be refreshed. The server will not have to send the entire page to the user’s computer.

This technology presents several benefits:

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

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

- **Automatic and 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 sufficiently recent browsers support AJAX technology. **AJAXAvailable** is used to determine if the AJAX technology is supported by the current browser. If a process that uses the AJAX technology is run on a browser that does not support this technology, the process is run "as if" it did not use the AJAX technology (the entire page is refreshed for example).

---

**Automatic and immediate AJAX**

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

For example, a site page is used to find out the different characteristics of a country (capital city, currency, flag, location, etc.). 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

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

- 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. When creating a PHP page, an additional code is displayed: a PHP code run on the server.
- New WLanguage functions specific to the Web are available.
- Some WLanguage functions not relevant for the Web do not exist anymore.
- Some WLanguage functions can only be used in an executable process in server code.
- Some WLanguage functions can only be used in an executable process in browser code.
- New types of controls specifically designed for Web applications are available:
  - Formatted display control,
  - Java Applet control,
  - Scrolling Banner control,
  - Navigation Bar control,
  - Captcha control,
  - Cell control,
  - Breadcrumb 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

<table>
<thead>
<tr>
<th>The development computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 1 computer: PC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The server-operating system</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 1 server: PC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The Web server software</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 1 Web server software: IIS, Apache, ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The PHP engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 1 PHP engine (version 4.3.2 or later)</td>
</tr>
</tbody>
</table>

### Developing a site

- **1 WEBDEV "Development"**
- **the site itself**
- **the data** *(dynamic site only)*
- **the data (optional)*
- **at least one browser**: Internet Explorer, Firefox, Chrome, ...

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

The development of a Web site using 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 the 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 characteristics specified when creating the project can be modified later.

2 Creating the project

To create a project:
1. Click on among the quick access buttons of WEBDEV. 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 (session mode or AWP), dynamic PHP site, etc.
   • the creation mode of the project: you have the ability to create a blank project or a project based on the Application RAD.
   • whether the project will be used 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 found in a control, in a page, in a report, ... can be translated.
3. Specify whether the project is associated with a database or not. If yes, it can be an existing database or a new one.
4. Validate the wizard. The created project becomes the current project.

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

3 Dashboard and project elements

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 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.

4 Operations performed on a project

Let’s see the main operations that can be performed on a project:
• Archive a project,
• Restore a project,
• Duplicate a project,
• Copy or delete a project,
• Rename a project.

See the online help for more details.
WEBDEV covers the entire development cycle of a site:

**Design**
- Specifications
- UML modeling
- Import of existing files

**Development**
- Create the project
- Create the analysis
- Development
  - Project RAD
  - Page RAD
  - Queries
  - Reports
  - Input of source code

**Test and generation**
- Test and debug of the site
- Site generation:
  - Session site
  - AWP site
  - PHP site
  - Webservice

**Deployment**
- Deploy the site on a WEBDEV Application Server
- Final test
- Open the site to the users

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 project and the analysis are created via specific wizards. The development can be done in RAD mode (Rapid Development Application) with automatic generation of the code and GUI or it can result from the manual creation of 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**: A dynamic WEBDEV site (in Session or AWP mode) is deployed on a WEBDEV Application Server. When the site is deployed, it is possible to run the last tests before making the site available to the users.
The following steps are required to create a page with WEBDEV:

1. Creating a page in the WEBDEV page editor.

2. Entering code (optional step).

3. Saving the page.

The following files are automatically created:

- **Available types of pages**
  - **Session WEBDEV page**
  - **Static WEBDEV page**
  - **AWP page**
  - **PHP page**

**Generated files**

- **Source file of pages in WEBDEV format (.WWH)**
- **Generic HTML WDL**
- **.HTML Generic**
  - 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.
- **.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.
- **.AWP AWP page**
  - This file is generated from the WWH file and it contains the description of controls, the server code, and the browser code. 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.

**Legend:**

- **.WWP: 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, which is generated when the WEBDEV site deployed (in Session or AWP mode), contains the server code of the project pages. This file will be found on the server.
Editing a page: zoning mode

Editing a page in zoning mode is easy and powerful: all you have to do is split the page into "zones" via the "Pencil" tool. These 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. It is an alternative (of 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 page address is typed in the browser, the environment used is automatically detected and the corresponding page (computer or mobile device) is displayed in the browser. The address displayed in the browser does not change: only the content displayed is adapted to the system.

All the links and 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 "desktop" use first. When reducing the display resolutions (page size), the content and the features are moved, resized or hidden.

In the WEBDEV editor, the representation of different resolutions is performed via slices. A slice represents a browser size for a specific device. WEBDEV includes 3 slices:
- The Mobile slice corresponding to smartphone devices.
- The Tablet slice corresponding to tablet devices.
- The Desktop slice 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 the control templates.
Page templates

WEBDEV allows you to create page templates. A template is intended to contain the elements common to a set of site pages.

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

A page template 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, etc.

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, etc.

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.
- 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 on 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" type.
3. Specify whether the blank page should be created in the main or in the project's RAD.
4. Validate the creation of the page. A blank page is created.
5. The save window is automatically displayed.
6. Specify the page title and its name. The page corresponds to a *.WHY* file. By default, this file will be created in the main project directory. This name will be used to handle the page.
7. Validate the save window.
8. Create the controls in the page.

**Notes:**
- By default, the page created corresponds to the active type of site (static, Session, AWP or PHP). 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 on 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. In the tabs, choose the type of page to create:
   - the "Standard" tab allows you to create:
     - a blank page without any controls.
     - a blank page based on a page template, if the project contains a page template.
     - a page based on a predefined template.
   - the "Internal page" tab is used to create internal pages.

**Notes:**
- The "Error pages" tab allows you to create pages to customize the errors displayed on the site.
- The "RAD" tab is used to create RAD pages that allow data display and input. This type of page can be based on a data file, on a query, etc.
- The "RID" tab is used to create RID pages that allow data input. Only the interface of the page will be generated. The programming must be done by the developer.
- Depending on the selected type of page, enter the requested information in the different steps of the wizard.
- Validate the page creation.
- The window for saving an element appears automatically. Specify the page title and its name, then validate.

### 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 tab 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.
- 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:** Web pages intended to be viewed on the Apple iPhone phone may benefit from special enhancements: splash screen at launch, 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, etc.),
• ask to confirm a deletion,
• ...

To open a popup page from another page, you can use the browser function PopupDisplay. See the online help for more details.

7 Refreshing a Session 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, simply use the PageRefresh WLanguage function.

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

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

2. Re-displaying the page according to its context found on the server.

Using PageDisplay to refresh a page is not recommended because it takes 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 about the page:

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

8.2 Automatic operating mode

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

• A page context is opened when the Session page is displayed in the browser.
• The context of a page is updated according to the information typed by the Web user in the browser. This update is performed during the page validation (via a "Submit" button or with PageSubmit).
• The existing page contexts are closed when PageUse is used. The page context corresponding to the page to display with this function is then 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, etc.),
• by programming in WL.

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 "x" from a button, a link, an image, etc.:
1. Display the element description ("Description" from the popup menu of element).
2. In the "Action" area, select the type of action: "Display the xxx page".
3. Select the destination of action: current page, current browser, ... (specific frame for a frameset).
The action that was previously selected will be performed in this destination.
4. Validate.

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): Declaring the 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.
• Load (unload) page (browser code): Browser code run when the page is displayed in the browser.
• Unload the page (browser code): Browser code run when a new page is displayed in the browser.

9.3 Opening a page by programming
Several WL functions can be used to open a page.
• PageRefresh: Refreshes the page displayed according to its context.
• PageDisplay: Opens and displays a new page in the browser of the Web user.
• PageDisplayDialog: Opens and displays a new page modally in the user’s browser.
• PageUse: Closes all current pages (and their contexts) and opens a new page.
• PopupDisplay: Displays a popup in the page.
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 object that triggers the page opening (button, link, etc.).
• the target defined for the current page.

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 "Add other processes to the page", at the end of the list of processes. The window of optional processes is displayed.
3. Select the optional process to add and validate.

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,
• —

12 The menu options

12.1 Overview
A menu includes one or more options and sub-options. Each option is used to run a WL 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 Managing 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 option caption,
• 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 WL code with a menu option. Only the menu options without sub-menu can start a WL process. See the online help for more details.
# The different types of standard controls

<table>
<thead>
<tr>
<th>You want to...</th>
<th>Use ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display a text, a title, ...</td>
<td>Static control</td>
</tr>
<tr>
<td>Propose a caption with a specific layout (images, links, ...).</td>
<td>Rich Text Area control</td>
</tr>
<tr>
<td>Define a color area.</td>
<td>Cell control, HTML Table</td>
</tr>
<tr>
<td>Display a price, a quantity, an address, a date, a time, ...</td>
<td>Formatted display control</td>
</tr>
<tr>
<td>Select a value from a list (country, city, color, ...).</td>
<td>Radio Button control, Combo Box control, List Box control</td>
</tr>
<tr>
<td>Select several values from a list (message recipients, files to download, etc.).</td>
<td>Check Box control, List Box control</td>
</tr>
<tr>
<td>Display an animation (animated Gif).</td>
<td>Image control</td>
</tr>
<tr>
<td>Display a page according to the area that was clicked in an image.</td>
<td>Clickable Image control</td>
</tr>
<tr>
<td>Display a Flash animation.</td>
<td>Flash control</td>
</tr>
<tr>
<td>Use existing HTML code.</td>
<td>HTML control</td>
</tr>
<tr>
<td>Concepts</td>
<td>Concepts</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Display a video.</td>
<td>Video control</td>
</tr>
<tr>
<td>Display a page from another site in one of your pages.</td>
<td>IFrame control</td>
</tr>
<tr>
<td>Select and display a date on a calendar.</td>
<td>Calendar control</td>
</tr>
<tr>
<td>Display the appointments in scheduler or organizer format.</td>
<td>Scheduler control, Organizer control</td>
</tr>
<tr>
<td>Display a Column chart, a Line chart, a Pie chart.</td>
<td>Chart control</td>
</tr>
<tr>
<td>Display a page from your site inside one of your pages.</td>
<td>Internal page</td>
</tr>
<tr>
<td>Display an automatic menu (that is built while the site is browsed).</td>
<td>Breadcrumb control</td>
</tr>
<tr>
<td>Allow the Web user to give or view a rate.</td>
<td>Rating control</td>
</tr>
<tr>
<td>Display thumbnails.</td>
<td>Thumbnail control</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display a set of images in gallery format.</td>
<td>Gallery Looper control</td>
</tr>
<tr>
<td>Display Flex files.</td>
<td>Flex control</td>
</tr>
<tr>
<td>Include SilverLight controls.</td>
<td>SilverLight control</td>
</tr>
<tr>
<td>Expand/Collapse a display area.</td>
<td>Drawer control</td>
</tr>
<tr>
<td>Upload one or more files.</td>
<td>Upload control</td>
</tr>
<tr>
<td>Display several contents successively.</td>
<td>Sliding Banner control</td>
</tr>
<tr>
<td>Display the dashboard displaying various information in thumbnail format.</td>
<td>Dashboard control</td>
</tr>
<tr>
<td>Display a list of objects as a scrolling horizontal list.</td>
<td>Linear Looper control</td>
</tr>
<tr>
<td>Use external resources available on Internet (Angular JS components, Bootstrap snippets, ...).</td>
<td>Web Component control</td>
</tr>
<tr>
<td>Create a visual effect in a page by presenting an image partially covered by another one.</td>
<td>Peeling Corner control</td>
</tr>
</tbody>
</table>
Part 2: Developing a site

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 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,
- Drawer,
- Scrolling banner,
- Tab,
- Internal Page control,
- Control Template control,
- Supercontrol,
- HTML table,
- Menu,
- Pager,
- Site map,
- Breadcrumb,
- HTML control,
- IFrame control,
- Flash control,
- Flex control,
- Silverlight control,
- Java Applet control,
- Row,
- “Web component” control,
- Layout control.

1.1 Creating a new control

To create a control:
1. Select the type of control to create using the corresponding icon in the “Creation” pane of the WEBDEV menu.
2. The name 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,
- Line,
- HTML control,
- Java Applet control.

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 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 see and/or modify the characteristics of one or more controls in the modifier. See the online help for more details.

2.1 Displaying the characteristics

To display the description window of a control:
- double-click the control.
- select “Description” from the popup menu of 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 for more details.

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

GUI tab
The “GUI” tab is used to define the parameters of 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 a view) to with 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 multifile.

Content tab
The “Content” tab is available for 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 List box, Combo Box and Table controls 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 enter 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 control aspect: 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.
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 a control,
- modify the size of a 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 can only be used to see a control. For example, when selecting several controls, the white handles indicate that the selected controls cannot be resized.
- This type of handles is also displayed when a page is read-only for the developer: the controls cannot be modified.

3.2 Available operations
The page editor is used to:
- Select one or more controls (with the lasso for example).
- Group selected controls.
- Move a control.
- 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).
- The gray handles indicate the first selected control during a multiple selection. This control will be the reference control.

3.3 Displaying 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 the pages.
- the border guides, used to define a border of identical size on each side of the page.

Note: To see the page with or without its guides, select "Show guides" 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.

Handing 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 Aligning the controls
The alignment of controls allows you to create "professional" and outstanding interfaces. Several WEBDEV tools can be used to create 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 customizable zoom.

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.

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 that 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 change the zoom factor with the mouse wheel while holding down the Ctrl key.
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 runtime: 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 window.

- the management of width and/or height: this parameter is used to manage the behavior of the control when resized. The height or the width 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 description window ("Description" from the popup menu),
   - via "Anchor" from the popup menu (right click) of control.
3. Choose the desired type of anchoring in the window that appears.

Three types of anchoring can be defined:

- position anchoring: allows you to define the position of the control when resizing. The control can:
  - keep its height,
  - adapt to content,
  - change along with the browser.

- anchor in width: allows you to define the width behavior of the control when resizing. The control can:
  - keep its width,
  - adapt to content,
  - change along with the browser.

- anchor in height: allows you to define the height behavior of the control when resizing. The control can:
  - keep its height,
  - adapt to content,
  - change along with the browser.

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

5. Validate. If the "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 anchoring of controls, you can also 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 looper. The records found in a database can be displayed on each row.

Creating the looper in the WEBDEV editor

Displaying the looper 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:

1. Actions that can be performed locally (on the computer of the Web user)
   - simple input check, ...
   - read information in the database
   - calculations/processes by programming, ...

2. Actions that imply going back to the server computer (Session or AWP pages)
   - read information in the database calculations/processes by programming, ...

To manage these two types of actions, the WEBDEV code editor differentiates two types of code:
• Server code (yellow or pink code in the code editor): This code is written in WLanguage (yellow code) or in PHP (pink code, only available in PHP pages). This code is run on the server. This code is 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 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. Simply click on "Add other processes to xxx" at the bottom of the code window: the list of additional events is displayed.

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 enough:

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

Here is the equivalent process in JavaScript:

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

</SCRIPT>
```

WLanguage code is easier to write, understand and maintain. It is much more reliable. 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**, entered in WLanguage (or in Javascript) in the code editor. This code is included in the HTML page and it will be run on the computer of Web user.
- **a server click code**, entered in WLanguage (only in dynamic pages). This code will be run on the server.

<table>
<thead>
<tr>
<th>Browser click code</th>
<th>Server click code</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Click code" /></td>
<td><img src="image2.png" alt="Click code" /></td>
</tr>
</tbody>
</table>

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

The aspect of the 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 a directory.

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.

### Web server

On the browser of the Web user

When validating the page, the "uploaded" file is copied into the hard disk of the server.

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

### Note:
WEBDEV also offers an advanced Upload control that requires a Flash player on the Web user’s computer. This control is used to manage a progress bar, to select several files, etc.

---

### Table: Action Sequence of codes run

<table>
<thead>
<tr>
<th>Action</th>
<th>Sequence of codes run</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send values to the server (submit)</td>
<td>Associated action:</td>
<td>Page context automatically updated on the server (Session or AWP dynamic pages only).</td>
</tr>
<tr>
<td></td>
<td>Execute browser codes only</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Execute server and browser codes</td>
<td></td>
</tr>
<tr>
<td>Code run:</td>
<td>1. Browser click code of control.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Browser code of page submit (if it exists).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Sending the values found in the page controls to update the page context on the server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Server code of control.</td>
<td></td>
</tr>
<tr>
<td>Reinitializing pages (reset)</td>
<td>Associated action: Display a page of the site</td>
<td>Page context automatically updated on the server (Session or AWP dynamic pages only).</td>
</tr>
<tr>
<td></td>
<td>Code run:</td>
<td>Automatic page display.</td>
</tr>
<tr>
<td></td>
<td>1. Browser click code of control.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Browser code of page submit (if it exists).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Sending the values found in the page controls to update the page context on the server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Server code of control.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5. Page display.</td>
<td></td>
</tr>
<tr>
<td>Do not send anything to the server</td>
<td>Associated action: Execute browser codes only</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code run:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. Browser click code of control.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Associated action: Execute server and browser codes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code run:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. Browser click code of control.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Server code of control.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Associated action: Display a page of the site</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code run:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. Browser click code of control.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Server code of control.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Page display.</td>
<td></td>
</tr>
</tbody>
</table>
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.](image)

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.

![Using the template in a page: The elements that belong to the template are bordered in blue and identified by a yellow square.](image)

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).

![Development computer Report sources](image)

![Server Print modes](image)
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 the Web user, we advise you to propose at least two print modes:

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

Report in the report editor of WEBDEV

Print in HTML

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

Which page should be displayed?

The page to display can be defined:
• in the page editor (no programming is required).
• in the code editor, in a process of the button (or link or clickable image, etc.).

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 type of action, select "Display a page of the site".
• 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, etc.).

Where should the page be displayed?

A page is displayed in a specific "target": current page, new browser, etc.
The target can be defined:
• in the page editor: no programming is required.
• in the code editor, in a process of the button (or link).

Several WLanguage functions allow you to open a new page (or frameset). The main functions are:
• PageDisplay (for pages),
• FramesetDisplay (for framesets),
• PopupDisplay (for popup pages),
• PageDisplayDialog (for modal page display).

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

In a button, to define the target of the page to display:
• Display the description of button or link ("Description" from the popup menu of control).
• In the list of targets, select a preset target or a frame (in the case of 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.
The ChangeTarget WLanguage function is used to modify the target of an action by programming. This function must be used in browser code.
Choosing a target

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

- a preset target: four preset targets are proposed:
  - Current browser (_top),
  - Current frame (_self),
  - Parent frame (_parent),
  - New browser (_blank). In this case, the new browser can correspond to a new tab or a new browser with specific characteristics (menu bar, message bar, etc.).
- 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)**

Initial page  
New page displayed in the entire browser

**Current frame (_self)**

Initial page  
New page displayed in the same frame

**Parent frame (_parent)**

Initial page (2 nested framesets)  
New page displayed in the page containing the second frameset

**New browser (_blank)**

Initial page  
New page displayed in a new browser window
Performing a process on several successive pages

In a business site, placing orders can be performed on several pages:
• page 1: Selecting the products to order.
• page 2: Showing the basket.
• page 3: Validating the order and entering the customer details.

In this case, the write-to-file operation must ONLY be performed in the code of the button for validating the order: all the order elements must be stored until the final validation. Don’t write into 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 gives you the ability to modify the style book of a site by only making a few modifications to the styles used.

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

Note: “Custom CSS” tab enables you to enter 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 in a site:

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

- **site anchored in width**:
Page content takes up all 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.

Site SEO

To visit your site, the Web users must be able to find it. To do so, your site must come up in search engines when a Web user types keywords corresponding to your site.

To improve your site’s SEO, WEBDEV proposes:

- an SEO wizard. This wizard lists all the optimizations that can be performed to improve your site’s SEO. To start the wizard, on the "Project" pane, in the "Web" group, expand "SEO" and select "SEO wizard".
- 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: SEO is a technique whose rules are dictated by search engines (Google, Bing, etc.). These rules are not always explicit and evolve very quickly. Don’t hesitate to read books or to visit sites dedicated to this topic.
SEO of a site in practice

1 SEO principles

In order for your WEBDEV site to be visited, the Web users must be able to find it. To do so, your site must come up the Web user types keywords corresponding to your site in a search engine.

1.1 Improving the SEO of a WEBDEV site

To improve the SEO of your pages, we recommend that you use AWP pages.

To optimize a dynamic site, use:
- a home page.
- a section of the site in static mode. The static site will be used to perform the SEO and to start the dynamic site.
- a section of the site in AWP mode. The AWP site will be used to perform the SEO 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 optimized while the AWP contexts cannot.

2 Using (or not) a static or AWP page of a site for SEO

To optimize a static or AWP page of your site:

1. Display the description window of the page.
2. In the "Details" tab, a specific area is used to manage the page SEO. This solution allows you to optimize 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.

3. It is possible to:
   - Do not use the current page for SEO.
   - Optimize the current page by associating it with expressions and keywords ("Edit the expressions and the keywords").

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

   Tips:
   - The first three keywords are the most important ones. Use common keywords and pertinent keywords.
   - The keywords must not be repeated.
   - Use variations (singular/plural, noun/verb, ...).
   - Note: These keywords can be dynamically modified by "Keywords".

5. Validate the page SEO and description windows.

3 Optimizing a page and entering its keywords

To optimize a static or AWP page of your site:

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

   Tips:
   - Use short sentences, limit the number of words (up to 200 characters).
   - This description must entice the Web users to access the page directly. You can also choose not to optimize a page.
   - Display the requested page in the editor.

Note: This description can be dynamically modified using "Description".

4 Using the SEO wizard

To help you improve the SEO of your sites and pages, WEBDEV proposes an SEO wizard. This wizard can be used for a specific page or for all the pages of your project. This wizard 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 SEO wizard for all the pages of a site:
1. On the "Project" pane, in the "Web" group, expand "SEO" and select "SEO wizard".
2. The SEO wizard starts. Double-click a suggestion to perform the corresponding optimization. The "Refresh" button is used to update the list of suggestions. The results can be sorted by page or by type of advice.
5 Registering your site in the search engines

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

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

Two types of search engines are available:

• the "Automatic" search engines:
  These search engines are based on automatic programs for analyzing the content.
  How can I get my site optimized?
  An "Optimize your site" link is often available for this type of search engine.
  Then, all you have to do is specify the address of the site and the email of the Webmaster. The site will be automatically monitored (according to keywords and to the site content) and optimized.

• the "Phone book" search engines:
  These search engines optimize sites that have been previously checked by human beings.
  How can I get my site optimized?
  An "Optimize your site" link is often available for this type of search engine.
  In most cases, the SEO procedure is as follows:
  1. Choosing a category for the site (leisure, culture, etc.).
  2. Filling out a questionnaire about the site to optimize: Internet address of the site, email of the WebMaster, description of the site, ...
  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 the developer requirements, unified in a single environment:

- project editor.
- data model editor.
- UML editor.
- query 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 allows you to manage the project using a comprehensive dashboard. Various Widgets allow you to see:

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

2.2 Data model editor

The data model editor 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 the Conceptual Data Model (CDM) then automatic generation of the analysis.

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

- WYSIWYG editor ("What You See Is What You Get"). You can directly view all the data files and the links of your project.
- Creation and description of the data files and their items.

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.

Let’s see some characteristics of the UML editor:

- Use of 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.
- Check of referential integrity.
- Automatic analysis generation.
- Zoom in the data model editor.
- Interaction between the different panes and the analysis elements.

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 immediately run.

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

Let’s see some characteristics of the query editor:

- WYSIWYG editor ("What You See Is What You Get"). You can 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 paste, copy 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.
- Real-time interface checker to simplify the positioning of controls.
- Ability to type the captions of controls in the work area of the 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 the "Reports and Queries Guide".

Let’s see some characteristics of the report editor:

- WYSIWYG editor ("What You See Is What You Get"). You have the ability to view the reports as they will be printed.
- Simplified report creation without writing a single code line.
- Drag and Drop to copy, paste or move controls from a report to another one.
- Skin templates of reports and preset control styles.
- Real-time interface checker to simplify the positioning of controls.
- Use of a form in a 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 is used 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.
• Preview of the different processes of a window, control or report found in the project.
• Insertion of processes specific to the mouse or keyboard use.

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.
• Management of texts.
• Management of 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 the progress 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 different audits,
• list of elements checked out from the 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 the projects belonging 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 elements, excluding the GUI (pages and windows), are 100% compatible and sharable between 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 on "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.

It is possible to share the elements found in the SCM:
- 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:

1. A project is created on Local Computer 1.
2. A version of the project is checked out from the SCM server.
3. Modifications are made on the project.
4. The modified version is checked in.
5. Another version of the project is checked out from the SCM server.
6. Modifications are made on the project.
7. The modified version is checked back in.

If a project element (window, 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 repository 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.

1. The Source Code Manager

1.1 Overview

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

 setSize 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, ...).

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

• make teamwork easier and faster,
• store the history of modifications and versions,
• view the source code of the development team in an automatic way.

For team between 1 and 300 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 SCM repository: procedures, classes, pages, reports, components, analyses, ...

This repository 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 used in offline mode (in train, plane, ...).

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").

1.2 Principle

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

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

To work on a project element (page, report, etc.), the developer must check out the element from the repository, modify it and check it 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 integrated messaging).
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 the 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 repository to use.
   Specify whether you want to use:
   • a repository found on a network share (SCM repository). Specify the directory of the SCM repository (network file system or shared directory).
   • an SCM repository found on a HFSQ Client/Server. The SCM repository will be in HFSQ Client/Server format. The SCM repository will be created.
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 the 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, this option proposes to retrieve the latest version of project elements. By default, the latest version of elements is automatically retrieved.

• Propose to check in the elements 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".

• the name and password of an HFSQ database administrator.
• an SCM Drive repository. Specify the information for identifying to your SCM Drive: email, password and team.
• an SCM repository found in PCS Cloud. Specify the name of cloud platform used and the project to open.

Note: If no SCM repository has been created yet, click on the "Create a repository" button.

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

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.
• Make backup copies of the repository on a regular basis (at least once a week).
• Store the server in a "secure" area and use a firewall.

2 Creating a repository for SCM

2.1 Overview

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

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

The repository can be created:
• in a shared directory on network: all the SCM users must have full rights on this directory. The repository will be in HFSQ Classic format.
• on a HFSQ Client/Server server: the repository will be in HFSQ Client/Server format. In this case, you must specify:
  • the server,
  • the database,
  • the port used,
  • the name and password of an HFSQ database administrator.

2.2 When should an SCM repository be created?

The SCM repository must be created only once. WEBDEV allows you to create this repository at different times:

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

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

2.3 Backups

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

Part 3: Development environment
4.2 Checking out an element

The different check-out modes

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

• **the standard mode**: if you display a SCM element that is not checked out, a panel indicates that the 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 a 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: nobody can check out this element until it is checked back in. The element can be checked out for test only.
   - for test: the element can be modified but the modifications will not be checked back in.
   - multiple: the element can also be checked out by other users. In this case, the differences between the different element versions can be viewed when the element is checked back in.
3. Validate. The element is opened. The title bar indicates that the element is checked out.

4.3 Check 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 repository with the local (checked out) element.
• access the element history in the repository.
• You can check in the modifications made to the element while keeping the element checked out (“Keep the element checked out” option).

4.4 Management modes of project

Two management modes are available with the SCM:

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

**Master and guest**

The Source Code Manager distinguishes between 2 types of users:

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

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

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

To switch from master mode to guest mode, under the “GDS” pane, in the “Project” group, scroll down to “Master / Guest” and select “Become guest on the project”.

To switch from guest mode to master mode, under the “GDS” pane, in the “Project” group, scroll down “Master / Guest” and select “Become master 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 to continue working on a project found in the SCM repository while being disconnected from the repository.

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 the modified elements back in.

See the online help for more details.

6 SCM administrator

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

It is used to:

• manage the SCM repositories (creation, connection to an SCM repository).
• manage the branches.
• manage files and directories present in a project in the repository (add, delete, rename files and directories).
• manage the different files of the repository (check-in, check-out, share, etc.).
• 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 repository, save it, restore it.
• add files of any type into the repository (.doc, .xls, .pdf, ...).
Internal component

An internal component is a group of project elements. This group 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).

Creating the 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 subdirectory in your project containing all the 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 elements (or "public" elements) 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 data and runtime contexts:
   - Use the project analysis or no analysis (full share): The internal component accesses the data files of the project. The internal component and the project use the same 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 the project. The internal component and the project use different runtime contexts. This mode is reserved to specific cases.

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 group is used to:

- Organize a project: the internal components can be used to group the project elements, by feature for example.
- Share the elements among several projects: The elements found in an internal component can be shared between several projects via the 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 the component elements in the interface of the 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.
   - 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.
5. Specify the management mode of data and runtime contexts. Three modes are available to manage data and runtime contexts:
   - Use the project analysis or no analysis (full share): The internal component accesses the data files of the project. The internal component and the project use the same runtime contexts.
   - Use the project analysis with different runtime context (advanced mode): The internal component accesses the data files of the project. The internal component and the project use different 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 the 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 uses an analysis. In this case, the analysis used by the internal component must be specified. You also have the ability to create a new analysis directly. This analysis will be associated with the internal component.

6. Validate the creation of internal component. At any moment, it is possible to:
• Modify the characteristics of the internal component via the description window of internal component.
• Handle the internal component and its elements.

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

3 Sharing the internal components (via SCM)

The internal components can be shared between projects via the SCM. To share an internal component via the SCM, the project containing the internal component must be found in the SCM. You can:
• create the internal component from a project found in SCM. The internal component will be automatically found in the SCM.
• import a project containing one or more internal components into SCM. The internal components will be automatically included in the SCM.

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

The elements generated by RAD for the internal component will be generated in the directory of 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.

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 WINDEV 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 this chapter, we will refer to “external component” as “component”.

An external component can be distributed to other WINDEV developers (free of charge or not). These developers will be able to easily include the feature proposed by the component in their site. The component will be included in the 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 via the project explorer or by programming.
• the mode for using the component (how to use the component, the parameters used to access the component features, ...). A short documentation (to be completed) is automatically generated and associated with the component.

Note: Several 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>.WDL
  Component file. Contains all component elements. 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>.WDO
  Description of the component (when developing sites only). This file is required to include the component in a project. This file must not be supplied in the setup procedure of the site that is using the component.

- <ComponentName>.WDI (optional file)
  Optional file. File in text format containing the list of additional elements supplied with the component (data files, .INI files, ...). See “The WDO file”, page 109.

1.3 What is an external component made of?
An external component contains:
• the different elements to distribute (pages, reports, ...). These elements can be accessible (or not) when the component is imported into a WEBDEV project.
• a short component description.

2 Creating and generating an external component

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

The component creation is performed in several steps:
1. Developing the component elements.
2. Creating the component.
3. Defining the advanced component options.
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 the component
  Example: the component is directly used on the computer of the developer who created it.
• simple distribution, by providing the necessary files.
  Example: the component is intended to be used within the same company, by several developers. In this case, the necessary files can be copied onto a network drive.
• setup procedure of 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>.WDL 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>.WDL 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>.WDL file and <ComponentName>.WDL 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 "The WDO file", page 109 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.
• 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.

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 105 for more details.

5 Deploying a site containing an external component

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, expand "Deploy the site".

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.

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 subdirectory of the project. 

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

5.2 Updating the components and the deployed sites

When updating a component and/or a deployed site, you have the ability to install on the deployment computer of the site:
• the site, the component and the files required for them to operate.
• the site and the files required for the site to operate.
• the component and the files required for the component to operate.

See the online help for more details.

Two methods can be used to update a component on the end-user computers:
1. Recompiling the host project
Recompile the project that uses the component and redistribute the site with the component. In this case, no version problem or compatibility problem will occur. The recompilation is required in the following cases:
• New features have been added 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 component versions.
• ...

2. Distributing the .WDK file directly
Provide an upgraded version of component (.WDK file) to the end users without recompiling the project. In most cases, this possibility applies when:
• The new version is used to correct the problems of an earlier version,
• New features have been added to the component but they are not required for the site to operate.

6 Modifying an external component

6.1 Overview

A component that was created and generated can be modified at any time. This modification can correspond to:
• the addition of elements into the component.
• the deletion of elements from the component.
• the modification of rights on the component elements.
• the modification of one of the component elements.

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

6.2 The different types of compatibility

The management of compatibility is linked to the component versions. Indeed, if modifications have been made to the component, the sites that 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.
• the forward compatibility: using a new component version with the projects compiled with an earlier version may not be allowed.

Therefore, the projects must be recompiled in order to use the new component version.
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 overview of component. This general overview is typed when generating the component.
• 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

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 "Describe" from the popup menu of component.

To get the documentation specific to a component element, double click on it ("Project explorer" tab) or press F2 from its code.

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 the current project:

- <ComponentName>.WDO
- <ComponentName>.WDI

These two files must be distributed along with the component.

If the component uses additional elements (data files, etc.), the following elements must be added into the project EXE directory:
• 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 a WDO file?

The <ComponentName>.WDO file is a file in TXT format that can be created and modified at any time. This file can be created and modified with Notepad, the standard text editor of Windows. This file contains the list of external files (data files, text files, ...) used by the component and that must be supplied and installed with the component. These files must be copied into the EXE directory of the projects that is using the component.

This "WDO" file can contain:
• the full name of file.
For example: C:\Components\PickerComponent\InitialStatus.INI
• the name of file. This file will be sought in the current directory of component.
For example: InitialStatus.INI
• a file name that is using a relative path.
The possible syntaxes are:
• Directory\FileName.xxx to specify a subdirectory of the current directory
• .\FileName.xxx to specify the current directory
• ..\FileName.xxx to indicate the parent directory
For example: ..\PickerComponent\InitialStatus.INI

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

In the wizard, you must:

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

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

Generation modes

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

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

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

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

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

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

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, this database will be accessed via an ODBC driver (MyODBC, ODBC for Oracle, etc.) or via the Native MySQL Connector (also called Native Access).
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.

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 on 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 of the wizard, you can choose how your project will be generated. 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 WDDeploy.
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. You see the PHP pages generated by WEBDEV.
The compilation of a PHP site is used to:
• find out the WLanguage programming errors.

2.4 Deploying a PHP site
The method for deploying a PHP site is identical to the method for deploying a static site. When creating the setup procedure, WDDeploy 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 allows you 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 you to choose one of the following options:
- create the database of Control Centers. This database will be automatically created in HFSQL Classic format in the specified directory.
- 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 the project at any time.

1. Creating requirements made of:
- Tasks
- Requirements
-Quality

2. Performing the requirements:
- Project Monitoring Center
- Incidents
- Automatic tests

3. Monitoring requirements (Project Monitoring Center):
- Pane of Business Rules
- Managing the requirements
- Image catalog (custom images)

Tasks
Requirements
Quality
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 (window, 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: the calculation of a specific VAT rate, the rules for changing the status of a customer, the formula for calculating shipping costs, a sales commission, a discount rate, a decay coefficient, ...

A business rule can be simple or complex.

The business rules can come from:
- the specifications (corresponding to 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 business rules apply and the percentage of rules currently implemented.
RAD (Rapid Application Development) and RID (Rapid Interface Design) are used to create pages from:

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

In RAD generation the generated pages contain all the code necessary for their operation. The test of these pages can be run immediately with the data found on the development computer.

In RID generation, the generated pages contain only the controls linked to items in the analysis. The entire code required for these pages to operate must be written by the developer. Your custom code can be typed directly.
WEBDEV allows you to create nine types of UML models:

- **Class diagram**: describes the overall structure of a system.
- **Use case diagram**: represents the system features from the user’s point of view.
- **Object diagram**: represents a set of objects and their relationships at a given time.
- **Component diagram**: describes the physical and static architecture of a computer application.
- **Activity diagram**: represents the behavior of a method or the progress of a use case.
- **Sequence diagram**: represents the chronological order of messages sent and received by a set of objects.
- **Collaboration diagram**: represents the structural organization of objects that send and receive messages.
- **State-transition diagram**: represents a sequence of states.
- **Deployment diagram**: shows the distribution of devices (nodes) used in a system as well as the association between the executable programs and these devices.

### 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 nine types of UML models:

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

This chapter only provides an overview of UML. 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 diagram below presents the management of stocks:

A class diagram includes the following elements:

- **Class**: represents the application structures. Each class is divided into three compartments:
  - the 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 Stocks class contains the ListProducts 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 the operations:

- **Public**: the element is visible to all 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 Orders class is linked to the Product and Customer classes. A Customer can place several Orders. An Order contains several products. An order must necessarily contain at least one product.
  - **Dependency**: Use relationship that establishes that the instances of a class are linked to the instances of another element. For example, the Orders class uses the Stocks class: before adding a product into an order, you must check whether this product is available in stock.
  - **Generalization**: Relationship between a general class (parent) and a specific class (child) that derives from it. For example, the Sail Boat class and Speed Boat class are derived from the Boat class.

2.2 Use case diagram

A use case diagram is used to view the behavior of a system in such way that:
• the user can understand how to use each element.
• the developer can implement these elements. For example, the behavior of a cell phone can be described via a use case diagram.

A use case diagram includes the following elements:
• **Actor**: represents the role of application users. For example, a person who works in a bank will be the loan manager. If this person has an account in this bank, he will also play the role of Customer.
• **Use case**: describes a sequence of actions run by the application. For example, Place an order, Enter an invoice, Create a new Customer entry, ...
  - A use case describes the actions performed by an application but it does not specify how the application performs these actions.
• **Relationship**: describes the behavior of actors in relation to the use cases. Three types of relationships are available:
  - **Association**: Structural relationship between two linked elements.
  - **Dependency**: Relationship establishing that an element uses another one. For example, a bank customer may get cash from an ATM. In this case, the Get Cash action depends on the Customer.
  - **Generalization**: relationship used to organize the elements according to a hierarchy.

2.3 Object diagram

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

• **Object**: represents an object. It is an instance of a class. An object has characteristics (attributes) and behavior (methods). An object can also have other objects as attributes.

For example:
• two types of Customer actor are available: Individual Customer or Enterprise Customer.
• the identity check can be performed according to two methods: typing a password or checking a fingerprint.

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.
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:

![Activity diagram of a dam]

An activity diagram includes the following elements:
- **activity**: presents a specific step in a 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 clutch" and "Change gear" before "Release clutch".**
- **by indicating the activities that will be performed in parallel.**
- **object**: used to attach activities to the object that performs these activities. For example, the "Order" and "Pay" activities are attached to the "Customer" object; the "Teach" and "Check knowledge" activities are attached to the "Teacher" object.
- **sending signal**: represents the sending of a signal toward an object.
- **waiting for signal**: represents the wait for a signal coming from an object.
- **transition**: represents the move from an activity that is ended to another one. For example: "Too much water", "Enough money", ...
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.

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:
• Price list view,
• Price list modification,
• Order entry,
• Customer entry.

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 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 dynamic WEBDEV sites running on Windows or Linux. The user groupware is not available in static, PHP, AWP or Ajax sites.

2 How does the user groupware operate?

A site that is 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: All user information and rights are stored in HFSQL format data files.
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 the data files of user groupware.

Integration mode of user groupware

Several modes can be used to integrate the user groupware:

• Automatic user groupware: all user groupware programming is automatically integrated into your site. No customization can be done. The skin template of project is automatically applied to all the 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 page of the user groupware is launched before any other page of the application. The initialization code of project is run once the login page is opened and validated.
• Manual start: The login page will open only if the function gpwOpen is used. This option is used to run the initialization code of the project for example, 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 that will be used in "Automatic 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 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: "PGWPW2001".

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.

3.2 Running the site test

When running the test of a site that supports the user groupware, the first page displayed is the login page (regardless of the first page defined in your application). A single user exists by default: the supervisor. To connect as the supervisor, all you have to do is enter the following information in the login page:

• Name: SUPERVISOR.
• Password: SUPERVISOR.

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

Notes:
• 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. Simply use the WLanguage gpwOpen function to open the login page.
• The pages for managing the user groupware are run before the initialization code of the project.
4 Configuring the site that is using the user groupware

When the site is deployed, the user groupware is configured by the application supervisor. 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 simply:
1. Start the site and connect as supervisor:
   • Name: SUPERVISOR.
   • Password: SUPERVISOR.
2. Select "Configuring the groupware".
3. The page for managing the user groupware is displayed. This page is used to:
   • manage the users and the 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 start of the application.

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

4.1 Managing users

The user management consists in:
• creating users,
• creating groups,
• associating users with groups.
To create a new user, you must specify:
• the user’s last name (mandatory),
• the user’s first name,
• the user login. This login corresponds to the user’s identifier when he connects to the application.
• the user’s password. This password is optional and it can be entered by the user during his first connection (check the corresponding option).
The user can be defined as being a supervisor of the site.
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 grouped.
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 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.
The following states are available for each element:
• Default: the element behavior corresponds to the default behavior, defined in the application.

5 Tips for a site that is managing the user groupware

5.1 Using groups of controls

To simplify the configuration of groupware, 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).

Notes:
• Active: the element is displayed but no input can be performed.
• Grayed: the element is grayed. No input can be performed.
• Invisible: the element is not displayed.

4.3 Displaying statistics

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

4.4 Migrating data

From version 19, the user groupware has evolved. Different data files are used by the user groupware. The administrator of user groupware can migrate the existing data to the new format and therefore automatically retrieve the data used in the earlier versions. This migration can be performed regardless of the data format: HFSQL Classic, HFSQL Client/Server, Native Access.
### Multilingual sites

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

1. **Choosing languages**
   - Project
   - Analysis

2. **Translating project elements**
   - Direct translation
   - Translation via WDMSG, WDINT, WDTRAD, ...

3. **Programming**
   - Data files
   - Choosing the site language (Nation function)
   - Choosing the character set (ChangeCharset function)

4. **Library**
   - Choosing the library languages

5. **Setup program**
   - Available languages
   - Translating the setup program

### Multilingual sites in practice

**1 Overview**

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

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

1. **Choosing the languages supported by the project and the analysis.**
2. **Entering the different project elements (pages, code, ...) in the different 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.**

**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 messages associated with a control, ...).

When modifying the project languages, the modifications will be automatically taken into account:

- 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.

**Notes:**

- If the operating system supports several languages (Hebrew, Arabic, Greek, ...), the corresponding character set will be automatically used when typing translations 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.

**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 Windows control panel).

**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 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 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 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.

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

3 Translating the interface into 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 after translation.

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

To translate this information from WEBDEV:
1. 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 the project messages (captions of controls, ... ) and check them back in after translation. 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, ...), 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
Management of 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\Data\Preset pages\HFSQL - Automatic help pages.
2. Customize the management of errors to use the HFSQL pages for error management (WbErr).
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,

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 language of the application.
10 tips about ergonomics ...

1. Enhance the style of your home page.
The home page is a major element of your site. In some sites, it is just a connection page but in most Internet sites, this is truly the main page for attracting visitors (and future customers of the site). Its content and its presentation must be studied closely. Don’t hesitate to read books about user-friendliness of sites to help you.

2. Get in touch
   When Web users visit a site, they must be able to identify the person or the company in charge of the site: simply provide a name, a mailing address and an Internet address, ... allowing the Internet user to contact you.

   This page can show a map specifying a store’s location, opening hours, etc.

3. Are your pages very long? Give the Web user ways to go back to the top.
   If your pages are very long (more than 2 screens), use buttons or links to go back to the top of the page (to avoid using the scrollbar).

4. Harmonize the style and alignment of your buttons.
   Make sure 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.

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 Are there 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.

8 Are you using images? Make sure they are displayed properly in your pages.

If a display problem occurs, check whether:
• the image exists,
• the image is saved in a format recognized by the browser (GIF or JPG for example),
• the image is found in the sub-directory of the <ProjectName>_WEB project.

9 The Web user must type a quantity? Use a combo box to simplify the input of quantities.

10 Are you proposing a product catalog with the possibility to order?
Specify all the 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 has an "Overlay and move" option ("GUI" tab in the control description window). If the "Stackable" option is selected, the control will no longer be linked to the page positioning tables. The control can be moved and placed "over" every other control found in your page.

**Caution:** The control where "Stackable" is checked will move over the other controls. If this option has been enabled for several controls, the order in which they are overlaid can be defined via the "Up", "Down", "Foreground" and "Background" options of the "Modifications" tab.

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.

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.

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

You can choose among several button/link actions, depending on the operation to be performed on the page controls.

The action can be chosen in the control description ("General" tab).

Let’s see some examples of "standard" uses of buttons/links in a page:

• Send the data found in the current page to the server for processing
  Example: Registration form: the link is used to check the validity of information entered and to display a confirmation page.
  To perform this type of process, use a link with the action "Submit". The page data is sent to the server in order to be processed.

• Perform a process on the server, without retrieving the data of the current page
  Example: Menu page made of buttons: each button is used to start a page of the site.
  To perform this type of process, use a button with the action "Do not send anything to the server".

• Perform a process in browser code
  Example: Performing a simple calculation locally, an input check, ...
  To perform this type of process, use a button with the action "Do not send anything to the server".

• Display a page external to the WEBDEV site
  Example: Launch a search engine site via a button.
  To perform this type of process, use a button with the action "Do not send anything to the server".

• Reinitialize the page controls
  Example: Link used to clear the controls found in the current page.
  To perform this type of process, use a button with the action "Delete content of the controls on the browser".
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 using the browser “Back” button to go back to a page:
1. Display the page in the WEBDEV page editor.
2. On the “Page” pane, click on the group icon of “Edit” group ( ). The page description window is displayed.
3. Display the “GU” tab.
4. In the option “Using the browser “Back” button”, select “Forbidden”.
5. Validate.

Security benefit: managing the "Back" button

A browser allows you to navigate through the different pages of a site using the “Next” and “Back” buttons.

A specific page context is created on the server whenever a new Session page is displayed in the browser.

When the same Session page is used to display different data (page with browsing table, loopers or “Form with browse” page, etc.), the page context on the server changes according to the data displayed.

When the Web user clicks on the “Back” button, the browser does not inform the site or the server. The browser goes back to the previous page without the corresponding context.

No specific management is required when using a “Back” operation to go from one page to another.

A specific management is required when the same Session page displays different data.

Why manage the "Back" button? A simple example...

To understand why the “Back” button must be managed, let’s take a look at a simple example: a Session page that displays the list of suppliers.

Step 1: Displaying the list of suppliers

The context contains the list of suppliers displayed in the Table control.
Step 2: Clicking the "Customers" link.

The customers are displayed. The page context is updated on the server: the context contains the list of elements displayed in this Table control (list of customers).

Step 3: Clicking the browser "Back" button.

The Session page displayed on the browser corresponds to the page displayed in step 1. However, the browser did not inform the server that the "Back" button was used 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 control, the selected element will be the corresponding element in the context: a customer. A desynchronization occurs between the dynamic page viewed and the page context found on the server. The WEBDEV engine detects this desynchronization.

This is why the Back button must be managed.

How to manage the "Back" button?

To manage the "Back" button, you can use:

• the page synchronization code in Session mode. This code is called in case of desynchronization.
• a hidden Edit control (invisible Edit control) to identify the record displayed in the browser.
• the option "Call the page synchronization process if the user has used the browser "Back" button" on your buttons and/or links ("Advanced" tab of the button description).

You can choose and combine either of these options according to the type of action that can be performed in the page in Session mode.

Detailed example of how to manage the "Back" button

A dynamic page is used to browse the different products of a site. The buttons found in the page are as follows:

• Back to main menu
  This action requires no specific management of the "Back" button. This button does not handle data files: a desynchronization between the context and the page displayed has no effect.
• Delete the current product
  The "Delete the current product" action handles a data file. It is important that the deleted record corresponds to the record viewed by the Web user.
  To manage the "Back" button, we recommend that you use:
  • a hidden (invisible) control in the dynamic page: this control will contain the identifier of the record actually displayed in the browser.
  • the synchronization code: this code is used to update the context (search for the record viewed by the Web user).

The actions performed in the synchronization code will be:

1. Re-read the record corresponding to the identifier stored in the hidden control.
2. If the record is not found: an error page is displayed and the code of "Delete" button is not run.
3. If the record is found, the page context is automatically modified on the server: the code of "Delete" button is run.

Total security!
Managing the "Back" button in practice

1 Overview

The browser "Back" button allows the Web users to display the pages that have already been visited. In a WEBDEV site, if a 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 a server action: the server sends a response to the browser. The click on the browser "Back" button is used to display the previous 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.

1.2 Example of desynchronization

Let's see a site example:

- A browser page contains a browsing Table control linked to the ITEM file and a "Next" link.
- The ITEM file contains a single item, each record includes a letter of the alphabet.
- A page is used to display 6 rows of the Table control, the "Next" link is used to display the next 6 rows.

When opening the page, the Table control displays the 6 first records of the file (from 'A' to 'F'). Let's see the sequence of actions performed by the user:

1. Click on the "Next" link
   Result: the server is positioned on the next 6 records of the ITEM file and returns their contents to the browser. The browser displays the next page of the Table control with the 6 new contents ("G" to 'L').
2. Click on the browser "Back" button
   Result: the browser displays the page preceding the first action. The Table control displayed contains the letters 'A' to 'F'. The server was not contacted, therefore it is still positioned on the records 'G' to 'L'.
3. Click on the "Next" link
   Result: the server is positioned on the next 6 records of ITEM ("M" to 'R'). The browser is synchronized with the server and it displays the same elements: the Web user has the feeling that some information is missing.

This behavior can have unexpected consequences when modifying a file record (modification of a record other than the one viewed by the Web user for example).

Reminder: each action on the browser must trigger a server action: the server sends a response to the browser. The click on the browser "Back" button being a browser action independent of your WEBDEV site, the second condition may not be performed.

2 Preventing from using the "Back" button

If the browser "Back" button is used to display the previous page, this action will have no effect.

2.1 Operating mode

Disabling the "Previous page" feature of the browser inserts the following Javascript code into the generated HTML page:

```html
<SCRIPT LANGUAGE="JavaScript">
  history.forward()  
</SCRIPT>
```

When running the page in a browser, it will not be possible to go back to this page by using the "Back" button.

Notes:
- Clicking the browser "Back" button can make the page blink.
- This mechanism can fail if the browser [STOP] button is clicked before the forward() statement is run by the browser.

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 can be used to manage 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.
To implement the management of the synchronization in a page:
1. Display the description window of the page: on the "Page" pane, in the "Description" group, click "Description".
2. In the "GUI" tab, for "Using the browser "Back" button", select "Allowed".
3. Validate. This page will be automatically included in the history of browser pages: to go back to this page, click the browser "Back" button.

To implement the management of synchronization in all the project pages:
1. Display the project description: on the "Project" pane, in the "Description" group, click "Description".
2. In the "Advanced" tab, check "Use the mechanism for synchronizing pages".

4 Synchronization by programming

To manage the synchronization by programming:
1. Display the description window of the page: on the "Page" pane, in the "Description" group, click "Description".
2. In the "GUI" tab, for "Using the browser "Back" button", select "Allowed".
3. Configure (if necessary) the page controls for which the synchronization must not be managed. For each control that triggers an action on the server, you can specify whether the page synchronization must be managed (default option) or ignored during this action. To ignore the synchronization management, simply uncheck "Call page synchronization process if the user has used the browser "Previous" button" in the "Advanced" tab of the control description.
4. Enter the code required for custom management of synchronization in the page synchronization code. Use ChangeAction in the page synchronization code. This function is used to define the action that will be performed if the page is desynchronized.

Notes:
• The function ChangeAction is initialized with "No action" if a WLanuage function that allows displaying or re-displaying a page is used in the page synchronization code.
• To customize the synchronization message, all you have to do is enter in the synchronization code of the page:
  1. the custom message.
  2. the redisplay of the current page on the server (with PageRefresh for example).
• To perform a synchronization from the information found on the computer of the Web user, we recommend that you:
  1. Use a hidden control containing the identifier of the displayed and selected record.
  2. In the synchronization code, find the current record on the browser. This search is performed from its identifier found in the hidden control.
  3. Refresh the page.

To create cookies in a WEBDEV site, use CookieWrite.
To read the cookies in a WEBDEV site, use CookieRead.

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.
Protecting the access to the site: passwords

Managing the access to a site by password is required in two cases:
1. Protecting personal details.
2. Restricting the access to a site.

Protecting personal information

To access 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: When he first connects, the Web user identifies himself and enters 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 the resources required to manage personal passwords.

Limiting the access to the 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 online new products.

WEBDEV allows you to easily manage these different types of access to a site.

TLS/SSL: Encrypting information on the Web

By default, the data exchanged between the computer of Web user and the Web server is not encrypted. The authenticity of the server identity, the confidentiality and the integrity of exchanged data are not guaranteed.

To insure confidentiality, the most used method is the TLS/SSL protocol (Transport Layer Security, formerly Secure Socket Layer). This protocol guarantees the identity of 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 WEBDEV online help 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). All you have to do is call SSLActive in the browser code of the button (or link). As soon as the secure page is opened, all the actions will be performed in secure 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. Redirection for the page for order confirmation

Sending emails

In a Web site, emails make it possible to:
• simplify communication between Web users: sending emails to the site manager, mailshot, etc.
• validate an order: in a business site, an email is sent to the Web users in order to validate their orders.
• transfer data from a computer to another one: the new orders entered in a business site are transferred by email to the company headquarters, etc.

Two methods can be used to send emails from a site.

1. Sending the email from the Web user's computer

   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 the 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 (Session pages only)

In this case, the server manages the emails. The emails can be sent:
- in the WEBDEV site directly (for an email data transfer, or for an order confirmation email, for example);
- by the email spooler (supplied with WEBDEV).

Use this method for a custom management of emails:
- checking the content of the emails sent by the web users (*mailing-list* site),
- checking the validity of recipient address,
- encrypting the information sent by email,
- validating a process, etc.
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 (data files, items, etc.) used in your project.

The data model editor is used to easily create an analysis.

The analysis of a WEBDEV project corresponds to the LDM (Logical Data Model). The entire structure and data organization are described: the data is grouped by file (called data file). Each data file contains several data called items.

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 (data files, items, etc.) used in your project.

Two methods can be used to describe an analysis:
- **1st method**: Creating the analysis directly (which means a Logical Data Model (LDM)).
- **2nd method**: Creating the Conceptual Data Model (CDM), then automatically creating the analysis from the 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 the 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 the project analysis ("<Project name>\ANA directory"). This directory must be accessible in read/write.
   - the analysis caption that briefly describes the analysis subject.
   - 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 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 a data file starts.
2. Select "Create a new description of data file".
3. Specify:
   - the name of data file. This name is the logical name of the data file. It will be used to handle the data file.
   - the caption of data file that briefly describes the file subject.
   - 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.

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 a data file 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 a data file 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 data files whose description must be imported and validate.
5. The imported data file is automatically inserted into the current analysis.

**Solution 2: From the Windows explorer**

Drop the file description from the Windows explorer to the data model editor. For example:

- to transfer its description into the data model editor.

Here: an Oracle database on the server. All you have to do is drag the database name...
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, search direction, ...).
5. In the bottom section of the screen, specify the details about the shared information.
6. Validate the item description.

Note: You can create an item using the metatypes offered by WEBDEV. To do so, click the "+M" button on the right of the table. The list of available metatypes 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 164" for more details.

3 Characteristics of links defined in an analysis

3.1 Owner file and member file
When a link is defined between two data files, there is an owner data file and a member data file:
- the owner data file is the owner of the key,
- the member data file is a member of the analysis data files containing a copy of the key.
To manage the link between two data files, the key of owner data file is copied into the member data file.

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.

For example, the key of the Supplier data file is copied into each record of the Product data file. Several records of the Product data file can have the same key of the Supplier data file:
- Supplier is the owner data file,
- Product is the member data file.
This type of link is represented as follows:

3.2 The cardinalities
The cardinalities are used to count the links between the data files. The cardinality is defined according to the answers to the two following questions:
1. For each file record, how many records of the other file minimum is this record linked?
   - The answer 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 file record, what is the maximum number of records in the other data file to which this record is linked to?
   - The answer provides the second part of cardinality (maximum cardinality):
     - if the answer is "a single one", the cardinality is x,1.
     - if the answer is "several", the cardinality is 1,N.

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.
- 0,N cardinality: 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 the data files.

The referential integrity consists in checking that:
- if a record is deleted from the owner file, the corresponding records are also deleted from the member files,
- if a record is added into a member file, a corresponding record exists in the owner file,
- if a record is modified in the owner file, the unique key is not modified,...

The check of referential integrity depends on the nature of the link between 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

In a parallel link, each record in a data file (Product) is linked to a record in another data file (Info), and conversely.

To manage a parallel link, the identifier of each data file in one of the data files is copied into the Info data file. This identifier is also a unique key in the Info data file.

The parallelism of records in the data file is respected if the following operations are run simultaneously on the two data files:
- creating a record,
- deleting a record,
- reindexing with compression.

Optional link

For an optional link:
- Each record in a data file (Category) is associated with no record or with a single record in another data file (Group).
- Each record in 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 found in a data file (Product) has no associated record or a single associated record in another data file (Details).
- Each record in the other data file (Details) necessarily has 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 the Product data file is copied into the Detail data file. This key is also a unique key in the Detail data file.

The owner data file is Product and the member data file is Details.

Note: This type of link is rare because the two data files can be grouped into a single file.

Shared link

For a shared link, the same record in a data file (Supplier) can be shared by several records in another data file (Product).

To manage a shared link, the key of the Supplier data file is copied into the Product data file. It becomes a multiple key to increase the speed of integrity check.

The owner data file is Supplier, the member data file is Product.

Depending on the cardinality, we can distinguish between four types of shared links:
- Shared link with a 1,0 - 0,1 cardinality,
- Shared link with a 0,0 - 1,1 cardinality,
- Shared link with a 1,0 - 1,1 cardinality,
- Shared link with a 1,1 - 1,1 cardinality.

Shared link with a 0,0 - 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 0,0 - 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 0,0 - 1,1 link.

Shared link with a 1,0 - 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 1,0 - 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 that includes the two keys of linked data 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 ("Orders" data file) can contain one or more products. A product (Product data file) can be used in several orders.

In this case, a link file is required (OrderLine data file).

The OrderLine data file contains:
- a unique key containing the keys of the Product and Orders data files,
- 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:
• Duplicating/Copying an LDM: Duplicating an LDM allows you to have two identical LDMs with different names.
• Delete an LDM.
• Rename an LDM.
• Associate an LDM with a project.
• Enlarge or reduce 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.

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.

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) is not generated, the analysis description (the data files) cannot be used in the project.

To start generating the analysis, on the “Analysis” pane, in the “Analysis” group, click “Generation”.

The generation is performed in three steps:
• verification of modifications and generation of data files of the analysis description.
• automatic modification of accessible data files (files found in the "EXE" directory of project).
• Synchronizing 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.

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 indicate:
• one rectangle: few modifications have been made between the two versions.
• several rectangles: several modifications have been made between the two versions.

To see the details of modifications performed between two versions: Double-click the line containing the "small rectangles" or click on [Modifications].

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 physical files corresponding to the restored analysis version, 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.
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, etc.) 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 the databases and they are synchronized in real time.
- The read load can be balanced among the different servers.
- The physical configuration can evolve without any interruption for the client computers.
- If one of the servers crashes, the client is automatically redirected to an operating server.

---

**HFSQL: the physically created files**

The data model editor is used to describe the structure of data files. Different files are physically created depending on the information typed in the data model editor.

- Characteristics of the file (Analysis)
  - HFSQL Classic or Client/Server file
  - One or more search keys (unique or with duplicates)
  - One or more full-text indexes
  - One or more “Memo” items

- Files physically created
  - “.fic” file
  - “.ndx” file
  - “.ftx” file
  - “.mmo” file

Note: This diagram only presents the main created files. Other specific files can be created if the data file is using the logs, the transactions or the replication.
Associate 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 WLlanguage variables.

The method for displaying and retrieving the information is straightforward:
- The link between a control and an item or variable is defined in the page editor during the control description ("Link" tab).
- `PageToFile` is used to update the record or the variable with the data found in the page.
- `FileToPage` is used to update the data displayed in the page with either the information saved in the data file or the information saved in the variable.

Link between control and item

![Diagram showing the link between control and item](image)
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 control 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.

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 data 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 operations on the data (sort on any column, search 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 file records being loaded in memory, this type of control is recommended for the data files containing less than 100 000 records (to avoid memory overflow).
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 the 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:

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 Retrieving the data from a site>

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 is recommended for a site installed on a dedicated 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 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 during the project creation. Indeed, several rules must be followed when describing the analysis and the file items.

The replication can be performed by email or via the assisted universal replication. See the online help for more details.

Universal replication

The universal replication is used to update databases of 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 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 adaptations must be done by programming with HRplFilterProcedure.
3-tier architecture

The 3-tier architecture is a model for application architecture. Its basic principle consists in dividing an application into 3 layers:
- the presentation 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.
Running a site test: the elements to test

Running the test of a site is an essential step of its development. 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:

• **how the site works**: sequence of pages, execution of code typed, etc.
• **appearance of the site**: use different browsers (Internet Explorer, Chrome, Firefox, Edge, etc.), use different screen resolutions, resize the browser, etc.
• **Web-specific features**: print, cookies, browser "Back" key, etc.
• **access to the site by different Internet users**: management of logins and passwords, management of concurrent accesses to the data files, etc.
• **non-regression of a site during an update**: validate the modifications performed in a site and check whether the non-modified features are still accessible.
• **a 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 "Code Coverage". Code coverage is used to measure the coverage of 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 editor of WEBDEV.
- the WEBDEV administrator.
- WDTestSite, tool for running stress tests.

This module is required to run a dynamic WEBDEV site. On the development computer, the test of dynamic sites can be run from a test page created by the administrator.

The table below presents how and when these test modes must be used.

<table>
<thead>
<tr>
<th>Test mode</th>
<th>Type of test</th>
<th>Running the test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editor</td>
<td>Running the test from the editor allows you to test:</td>
<td>• On the &quot;Project&quot; pane, in the &quot;Test mode&quot; group, expand &quot;Test mode&quot; and select &quot;Trace the project&quot;.</td>
</tr>
<tr>
<td></td>
<td>• the site features.</td>
<td>• On the &quot;Project&quot; pane, in the &quot;Test mode&quot; group, expand &quot;Test mode&quot; and select &quot;Debug the project from the home page&quot;.</td>
</tr>
<tr>
<td></td>
<td>• the code used in the different processes. A powerful debugger allows you to monitor the execution of different processes.</td>
<td>The rights granted to the Web user (access rights, write rights and read rights) correspond to the rights granted to the developer.</td>
</tr>
<tr>
<td></td>
<td>• the use of the site with different browsers (installed on the development computer).</td>
<td>This type of test: • allows you to use the debugger. • is taken into account by the &quot;Code Coverage&quot;.</td>
</tr>
<tr>
<td></td>
<td>This type of test:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• allows you to use the debugger.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• is taken into account by the &quot;Code Coverage&quot;.</td>
<td></td>
</tr>
<tr>
<td>WEBDEV administrator (dynamic WEBDEV site only)</td>
<td>Running the test from the administrator allows you to test:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• the site features.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• the features specific to the Web (cookies, ...).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• the use of the site in real conditions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Start the WEBDEV administrator (&quot;Start&quot; menu) and click the &quot;Test page&quot; button (&quot;Advanced&quot; 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.</td>
<td></td>
</tr>
</tbody>
</table>

Notes:

- Each computer establishes a defined number of connections.
- Each connection performs the selected scenario.

Automatic tests

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".

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".
3. Create a test scenario (with WDTestSite).
4. Install WDTestSite and the scenario on the different computers to simulate an important number of accesses.
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, etc.

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 insert 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 placed 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 the current page in the editor can be run via 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: The deployed project can be debugged. This feature is very useful to reproduce a problem that occurs under specific conditions. To start the deployed project, on the "Project" 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 being used: The deployed project can be debugged while it is used. This feature is very useful to reproduce a problem that occurs under specific conditions. The most common case: a problem occurs while you are using the site; you can start the debugger immediately by setting a breakpoint in your 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 a 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),
• step-by-step project execution,
• test of performances of your site,
• regression test/automatic test,
• stress test.

The test of the entire project is used to simulate the startup of the site. This allows you to run test of entire site, even if its development is not finished yet. As soon as a problem occurs, you have the ability to start the debugger to identify and fix the problem.

The test of a single page is used to run the current page only. 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 has 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 test of a project 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, expand "Test mode" and select "Test browser".

Different types of tests

To run the test of a static site from the editor:
1. On the "Project" pane, in the "Test mode" group, expand "Test mode" and select "Debug the project" (or Ctrl + F9).
2. The editor is automatically minimized.
3. The browser specified in the WEBDEV options is opened and the site home page is displayed.

To run the test of a dynamic site (Session or AWP) from the editor, several methods are available:
• On the "Project" 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 test the static part of the site: perform the operations corresponding to the test of a static site.
• to test the dynamic part of the site (Session or AWP): perform the operations corresponding to the test of a dynamic site.

Dynamic site: Start

The following modules are automatically started during the test of a dynamic WEBDEV site (Session or AWP model):
• 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 (WD240Admin.EXE).
• The WEBDEV engine 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 the administrator ("Advanced" tab of WD240Admin, "Test page" button).

To stop the test, display the WEBDEV administrator (click the icon 24 in the taskbar) and click "Disconnect" ("Connections" tab).

To run the test from the WEBDEV administrator:
1. Start the WEBDEV administrator: on the "Tools" pane, in the "Web utilities" group, click "WDAdmin".
2. In the "Advanced" tab of WEBDEV administrator, click the "Test page" button.

To run the test from the WEBDEV administrator:
1. Start the WEBDEV administrator: on the "Tools" pane, in the "Web utilities" group, click "WDAdmin".
2. In the "Connection" tab, select the site and click the "Test" button.

2.3 Stress test/Regression test

WTestSite is used to run stress tests: WTestSite is used to start several simultaneous connections to a dynamic WEBDEV site (Session or AWP).

Each connection performs a set of actions in the WEBDEV site (preset scenario).

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 directly on the user computers.

From your office in London, you have the ability to debug a site running on a Web server in Taiwan. The debug operation is done without having to go anywhere, on the final configuration directly.

4 Running the test of a page

4.1 Running the page test from the editor

To run the test of a page from the editor:

1. Open the page whose test must be run.
2. Click among the quick access buttons of WEBDEV menu. You can also use the F9 key.
3. The editor is automatically minimized and the page is run.

During the test, all page features can be run. You will have the ability to open other pages for example.

4.2 Stopping the page test

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 press Alt + Tab.

Confirm that the test must be stopped. WEBDEV displays the editor that was used at the beginning of test.

4.3 Not verifying the page test

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.

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, etc.). See "Debugging without the debugger" (page 201) for more details.

5.2 Overview of debugger

The debugger is used to trace WL 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 is used to:
• find out the call stack,
• view the content of variables or expressions,
6 Performance test

6.1 Overview
The performance profiler allows you to check and optimize the execution time of your site. Its principle is straightforward:
• You run the test of your site.
• During this test, the performance profiler keeps track of all the actions performed and the corresponding processes run.
At the end of 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 analyzer, 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.

Note: The performance profiler displays the result of the analysis 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 processes run during the application test (from the slowest one to the fastest one).
• the "Calls" tab is used to see the details of operations performed in a process.

6.3 Reading the result of performance profiler
The performance profiler presents the result of the analysis 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 processes run during the application test (from the slowest one to the fastest one).
• the "Calls" tab is used to see the details of operations performed in a process.
The following information is displayed for each process:

<table>
<thead>
<tr>
<th>Function</th>
<th>Function, process or procedure run.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Time</td>
<td>Execution time of function.</td>
</tr>
<tr>
<td>Nb of calls</td>
<td>Number of calls made to the function (procedure or process).</td>
</tr>
<tr>
<td>Time 1 call</td>
<td>Execution time of a call to the function (procedure or process).</td>
</tr>
<tr>
<td>code %</td>
<td>Percentage of code run outside the call to a WLanguage function or outside the call to a custom function or procedure.</td>
</tr>
<tr>
<td>Parent</td>
<td>Process that called the function.</td>
</tr>
</tbody>
</table>

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.
• WDTestSite 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 the 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 WDTestSite
WDTestSite is used to run different tests on a WEBDEV site.
The following tests can be run by WDTestSite:
• 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 version still operates properly once the site was updated.
• Test of a site in multi-user mode:
The test of a site in multi-user mode is used to check whether concurrent accesses to the data files are managed properly. This test consists in simulating the simultaneous connection of several Web users to a WEBDEV site. Each Web user runs a set of operations (scenario) simultaneously.
• Comparison of different servers:
WDTestSite is used to compare the speed of different servers. To do so, run the same scenario on different servers and compare the execution time of this scenario.
• Optimization of processes developed in WLanguage:
WDTestSite is used to compare the execution time of a scenario before and after the WLanguage code was optimized. See the online help for more details (keyword: "WDTestSite").
Deploying a Web site

DEVELOP 10 TIMES FASTER
Deploying a static site

Principle

The deployment of a static site is performed by FTP:

1. Configuration of the FTP server
2. Setup by FTP (WDDeploy)
3. Implement the site (start link)

Information required for the deployment

No setup can be run without this information. Contact your hosting company to get this information.

Before performing a deployment by FTP, the hosting company must communicate the following information to the developer:
• Name of Web server where the setup will be performed (or its IP address).
• User name and password defined in the FTP server.

When deploying by FTP, the developer must specify in WDDeploy:
• the information provided by the hosting company.
• the files to install.
Deploying a static or PHP site in practice

1 Overview

When developing a static 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).

Two deployment methods are available:
- remote deployment (by FTP): recommended when the server is not directly accessible.
- deployment by network: recommended when the server is directly accessible.

WDDeploy is used to simplify the deployment of your static sites.

Note: To deploy a dynamic site, see Deploying a dynamic WEBDEV site in practice.

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-ROM, etc.) of the WEBDEV site, with creation of a setup version supplied on CD-ROM to the hosting company.
- remote deployment directly from the development computer (by file transfer via Internet in HTTP mode).

Note: WEBDEV application server can be included in the setup supplied by physical media.

Note: It is also possible to perform a deployment by file transmission in FTP mode. In this case, the hosting company must configure an FTP server.
Part 7: Deploying a Web site

- remote deployment from a management computer (file transfer through Internet, FTP).

Development computer

1. Create the setup package

2. Configure the FTP server and the account manager

Web server

3. Run the package: Install the site on the server by FTP

Management computer

4. Online the site (run link)

Note: It is also possible to perform this deployment by file transmission in FTP mode. In this case, an FTP server must be configured on the Web server.

Choosing a deployment method

The choice of deployment method mainly depends on the constraints imposed by the hosting company and on the location of deployment server (remote or local).

The setup via physical media is recommended if the Web server can be easily accessed (the Web server and the development team are located in the same premises, for example).

The remote setup (by HTTP) from the development computer is recommended if the deployment computer cannot be easily accessed.

Before performing a remote setup, the hosting company must configure the WEBDEV account manager.

The remote setup (by HTTP) from a management computer is recommended if the characteristics of the server are not known by the developer when creating the setup program. In this case, the setup program is called a "Package".

If a site is intended for several clients, you have the ability to create a single package and to distribute it to all the relevant clients.

The server settings (address, etc.) are specified only when the package is run on the management computer.

Deployment via physical media: necessary information

When deploying by physical media, the information 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 directory for generating the setup.

During the setup, you must supply:
- the directory of data files,
- the directory of site files.

After setup, the characteristics of the site (connection time-out, number of connections, etc.) 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 performing an HTTP deployment directly from the development computer, the host 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.
- The OS account used to run WEBDEV sites (it is associated with the WEBDEV account in the account manager).
When deploying by HTTP, the developer must specify in the wizard:
• the information provided by the hosting company,
• the files to install,
• the name of 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 HTTP 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 the site (from a management computer), the hosting company will have to specify:
• Name of Web server where the setup will be performed (or its IP address).
• User name and associated password defined in the WEBDEV account manager.
• The OS account used to run WEBDEV sites (it is associated with the WEBDEV account in the account manager).

Notes:
• A delayed setup can be performed at the specified date and time.
• The setup and the update by HTTP are secured by password and data encryption.

On-lining a dynamic WEBDEV site

Your dynamic WEBDEV site (in Session or AWP mode) can be accessed as soon as it is on-lined. On-lining a WEBDEV site consists in activating the site.

Several methods can be used to activate a WEBDEV site:
• when performing the setup by HTTP: a check box is used to automatically activate the site after setup.
• with the remote administrator.
• with the WEBDEV administrator found on the deployment computer ("Locked" option in the "Sites" tab).
Deploying a dynamic site in practice

When developing a dynamic site (in Session or AWP), one of the important phases consists of deploying this site on a Web server (at a hosting company or in Intranet).

Several deployment methods are available:

- remote deployment (by HTTP): recommended method when the server is not directly accessible.
- deployment by physical media (DVD,...): recommended method when the server is easily accessible.

1 Implementing a library

Before performing a setup, all the objects found in your dynamic site (in Session or AWP mode) must be included in a library. A library is a file that 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 remote setup (by HTTP or FTP)
- in order to perform a remote setup, the hosting company must have:
  1. Created and configured your FTP account on the server (for setups by FTP).
  2. Created and configured your WEBDEV account on the server.

The hosting company must provide you with your logins and passwords for these different accounts, as well as the name of the account that runs the sites associated with the WEBDEV account.

- deployment in the PC SOFT Cloud: PC SOFT offers a hosting platform based on the Cloud Computing model: you deploy without worrying about hardware constraints and you are billed based on your applications' real usage.
- an installation by FTP via the test hosting service of PC SOFT: PC SOFT proposes to host your test sites. The setup is easily performed, directly from the environment. Once your site is developed, you have the ability to run its test directly. A PC SOFT developer account is required to perform this type of setup. 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. This package can be run from any computer to install the Web site on the server by HTTP or FTP.
- immediate transfer: A server address (name of a computer accessible by network, IP address or Internet address),
- a setup by physical media:

2 Remote deployment (by FTP)

The deployment steps are as follows:

1. Install the WEBDEV application server at the hosting company (optional step). See the documentation about the WEBDEV application server for more details.
2. Creation of the following elements by the hosting company:
   • a WEBDEV site.
   • an FTP account (in the FTP server) only if the deployment is performed by FTP.
3. The hosting company supplies the names and associated passwords as well as the name (IP address) of the server.
4. Creating the remote setup from the editor: on the "Project" pane, in the "Generation" group, click "Deploy the site".
5. To deploy a static site, see "Deploying a static or PHP site in practice", page 206 for more details.

2.1 The steps

Before performing a setup, all the objects found in your dynamic site (in Session or AWP mode) must be included in a library. A library is a file that 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 remote setup (by HTTP or FTP)
- in order to perform a remote setup, the hosting company must have:
  1. Created and configured your FTP account on the server (for setups by FTP).
  2. Created and configured your WEBDEV account on the server.

The hosting company must provide you with your logins and passwords for these different accounts, as well as the name of the account that runs the sites associated with the WEBDEV account.

- deployment by package: recommended method when the server is not directly accessible and when its characteristics are unknown.
  Notes:
  • For each Web server that hosts WEBDEV sites, the hosting company must own a license for the WEBDEV application server.
  • To create the setup of a site, on the "Project" pane, in the "Generation" group, click "Deploy the site".
  • To deploy a static site, see "Deploying a static or PHP site in practice", page 206 for more details.

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. Creation of the following elements by the hosting company:
   • a WEBDEV site.
   • an FTP account (in the FTP server) only if the deployment is performed by FTP.
3. The hosting company supplies the names and associated passwords as well as the name (IP address) of the server.
4. Creating the remote setup from the editor: on the "Project" pane, in the "Generation" group, click "Deploy the site".
5. To deploy a static site, see "Deploying a static or PHP site in practice", page 206 for more details.

2. Remote deployment (by FTP)

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. Creation of the following elements by the hosting company:
   • a WEBDEV site.
   • an FTP account (in the FTP server) only if the deployment is performed by FTP.
3. The hosting company supplies the names and associated passwords as well as the name (IP address) of the server.
4. Creating the remote setup from the editor: on the "Project" pane, in the "Generation" group, click "Deploy the site".
5. To deploy a static site, see "Deploying a static or PHP site in practice", page 206 for more details.

2. Remote deployment (by FTP)

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. Creation of the following elements by the hosting company:
   • a WEBDEV site.
   • an FTP account (in the FTP server) only if the deployment is performed by FTP.
3. The hosting company supplies the names and associated passwords as well as the name (IP address) of the server.
4. Creating the remote setup from the editor: on the "Project" pane, in the "Generation" group, click "Deploy the site".
5. To deploy a static site, see "Deploying a static or PHP site in practice", page 206 for more details.

2. Remote deployment (by FTP)

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. Creation of the following elements by the hosting company:
   • a WEBDEV site.
   • an FTP account (in the FTP server) only if the deployment is performed by FTP.
3. The hosting company supplies the names and associated passwords as well as the name (IP address) of the server.
4. Creating the remote setup from the editor: on the "Project" pane, in the "Generation" group, click "Deploy the site".
5. To deploy a static site, see "Deploying a static or PHP site in practice", page 206 for more details.

2. Remote deployment (by FTP)

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. Creation of the following elements by the hosting company:
   • a WEBDEV site.
   • an FTP account (in the FTP server) only if the deployment is performed by FTP.
3. The hosting company supplies the names and associated passwords as well as the name (IP address) of the server.
4. Creating the remote setup from the editor: on the "Project" pane, in the "Generation" group, click "Deploy the site".
5. To deploy a static site, see "Deploying a static or PHP site in practice", page 206 for more details.

2. Remote deployment (by FTP)

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. Creation of the following elements by the hosting company:
   • a WEBDEV site.
   • an FTP account (in the FTP server) only if the deployment is performed by FTP.
3. The hosting company supplies the names and associated passwords as well as the name (IP address) of the server.
4. Creating the remote setup from the editor: on the "Project" pane, in the "Generation" group, click "Deploy the site".
5. To deploy a static site, see "Deploying a static or PHP site in practice", page 206 for more details.
4. Configuring the site. These options are used to configure the options for connecting to the site after its setup. These parameters can be modified if necessary:

- by the deployment administrator found on the server.
- by the remote administrator, that can be used by the site manager.

5. If an analysis update was performed, select the automatic modification of data files during the setup.

Note: if this option is selected, the update will be proposed during the setup.

Summary of elements installed as well as their location:

<table>
<thead>
<tr>
<th>Site server</th>
<th>Data server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory defined by the hosting company.</td>
<td>Directory defined by the hosting company.</td>
</tr>
<tr>
<td>In this directory, a sub-directory will be created for each site of user N.</td>
<td>In this directory, a sub-directory will be created for the data files of each site of user N.</td>
</tr>
<tr>
<td>The following elements are installed in this site sub-directory: HTML pages, the library (<em>.WLD</em>), the image directory.</td>
<td>The following elements are installed in this data sub-directory: the <em>.FIC</em> files, the <em>.NDX</em> files, the <em>.MMO</em> files.</td>
</tr>
<tr>
<td>Example: The directory of sites for user N is &quot;C:\UserN\WEBDEV sites&quot;. User N installs his site named &quot;Boat&quot;. This site will be installed in the &quot;C:\UserN\WEBDEV sites\Boat&quot; directory.</td>
<td>Example: The data directory of user N is &quot;D:\UserN\WEBDEV data&quot;. User N installs his site named &quot;Boat&quot;. By default, the data files will be installed in the &quot;D:\UserN\WEBDEV data\Boat&quot; directory.</td>
</tr>
</tbody>
</table>

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, DVD, etc.) to the Web server administrator (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. Creation by the hosting company of a WEBDEV account (in the administrator) and creation of an FTP account (in the FTP server) if necessary.

5. Running the deployment package on a computer and installing the WEBDEV site.

6. On-lining the WEBDEV site.

3.2 Creating the wizard

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 after its setup. These parameters can be modified if necessary:

- by the deployment administrator found on the server.
- by the remote administrator, that can be used by the site manager.

4. Specify whether your site is using the Native AS/400 Access.

Reminder: if your site is using the Native AS/400 Access, the AS/400 DLLs must be installed on the Web server.

If you are using a license for a limited number of computers, the number of current connections must be managed in the WEBDEV site (INI file or registry, ...).

3.3 Installing a dynamic site

To install a site, you must:

1. Run the deployment package.

2. Specify the characteristics of the Web server on which the WEBDEV site must be installed (characteristics of the server FTP account if necessary and characteristics of the server WEBDEV account).

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 caption for your setup. A title can be typed in each language that can be selected during the setup.

2. A default directory.

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 after its setup. These parameters can be modified if necessary:

- by the deployment administrator found on the server.
- by the remote administrator, that can be used by the site manager.

4. Specify whether your site is using the Native AS/400 Access.

Reminder: if your site is using the Native AS/400 Access, the AS/400 DLLs must be installed on the Web server.

If you are using a license for a limited number of computers, the number of current connections must be managed in the WEBDEV site (INI file or registry, ...).
To install the WEBDEV site on a Web server (at a hosting company or 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 lock menu.

3. Choose the setup language: a setup wizard starts. Go to the next plane.

4. The setup automatically detects whether one or more of the following Internet servers are installed on the computer:
   - IIS 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 displayed in this list; 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. To select it, you have to do a configuration update. 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 server can be selected directly; select your server. You can ask to generate a configuration file in order to run the configuration later. The configuration file can be modified before it is run.

Depending on the selected server, the extension of configuration file will be "reg", "conf" or something else.

5. Select the setup directory: webroot indicates the root directory of the computer’s Web server. Depending on the selected Web server, the root directory is detected and displayed. If the root directory of 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. Updating the data files (optional step, only for a site update).

To update the data files associated with the WEBDEV site, choose to run the automatic modification of the data files. Contact the provider of the site if you don’t know whether you must perform this update or not.

Caution: The data files must not be currently used. Use the WEBDEV administrator to check that no current connection is in progress and to stop the sites (check “Site locked” in the “Sites” tab).

7. Install an ODBC driver for HFSQL if programs not developed with WEBDEV must access the HFSQL files of the site that will be installed (optional step).

8. The setup summary is displayed. Confirm your choices.

9. The setup is completed:
   - Click the 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.

10. Configuring the Web server:
    - If the configuration of the Web server was not performed automatically, two cases may occur:
        - **Case 1:** your server was detected and a configuration file was generated; this file can be viewed, modified (if necessary) and run.
        - **Case 2:** your server was not detected: 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 the WEBDEV site from the administrator via the [test page] option ("Advanced" tab).

Notes:
• The WEBDEV administrator (WD240ADMIN.EXE) must be started and run in background task. It can be installed as service in Windows NT.
• See The WEBDEV administrator in practice for more details about the WEBDEV administrator.

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 your WEBDEV site to be started.

If the setup was perform 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 a "OURAPP" site from the PC SOFT site:

http://www.windev.com/WD240AWP/
WD240AWP.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 following address for example:

"http://205.51.231.57/WD240AWP/WD240AWP.EXE/CONNECT/prettyeyes"

Example of META tag that must be included in the <HEAD> section of an HTML page:

<meta http-equiv="refresh" content="0;URL=http://205.51.231.57/WD240AWP/WD240AWP.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/
What is the WEBDEV administrator used for?

In development (for test) and in deployment, the WEBDEV administrator is required to run the dynamic WEBDEV sites.

When developing a dynamic WEBDEV site, the administrator is mainly used to:

- run the test of dynamic sites via the test page ("Advanced" tab, "Test page" button).
- manage the web services ("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 to manage dynamic WEBDEV sites (Session or AWP). 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.

To perform a diagnostic if a problem occurs when starting a WEBDEV site.

In deployment, two types of WEBDEV administrators are available:

- the WEBDEV Deployment administrator that allows the hosting company to manage the dynamic WEBDEV sites installed on a Web server.
- the remote WEBDEV administrator that allows the site manager to manage 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,
- "Setup/Accounts" tab,
- "Logs" tab.

2.1 Characteristics of current connections

The "Connections" tab gives various information about the current connections. This information can be consulted at any time on the server:

- The 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 can also be 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".

Starting the WEBDEV administrator

To start the WEBDEV Development administrator:

- from WEBDEV Development version: on the "Tools" pane, in the "Web utilities" group, click "WDAdmin".
- select "Programs . WEBDEV 24 . WEBDEV administrator" from the "Start" menu.

The WEBDEV administrator is automatically started during the first test of the site.

Then, the administrator can be run in background task.

By default, the WEBDEV administrator allows ten simultaneous connections to the same dynamic WEBDEV site (GO icon).

The WEBDEV administrator is automatically started during the first test of the site. For tests: disconnection time, ...

This refresh operation is used to check the access to the site during the update.

Use the "Apply" button.

For more details about the deployment tools, see their help.

For more details about the deployment tools, see their help.

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 can also be used to:

- Delete the site selected in the table.
- It is used to remove the site from the administrator: the WEBDEV site will still be found on disk.
- Refresh the list of sites found in the table.
- This refresh operation is used to check the presence of a site in the administrator after its setup.
- Add a site into the table.
- Configure each site found on the computer.
- The configuration used by default is the one defined in the "Configuration" tab.
- You can also create a specific configuration for the selected site ("Parameters" button).
- This configuration affects the number of authorized connections, the authorized idle time and whether the site must be locked during an update. See "General configuration", page 224 for more details.
- To validate the new setting immediately, click the "Apply" button.
- You can also manage scheduled tasks associated with the selected site (Scheduled tasks" button). This setting is available when the site is deployed.
- Run the test of the sites installed on the computer (in development version only): Select the site and click on "Run the test".

Starting the WEBDEV administrator

To start the WEBDEV Development administrator:

- from WEBDEV Development version: on the "Tools" pane, in the "Web utilities" group, click "WDAdmin".
- select "Programs . WEBDEV 24 . WEBDEV administrator" from the "Start" menu.

The WEBDEV administrator is automatically started during the first test of the site.

Then, the administrator can be run in background task.

By default, the WEBDEV administrator allows ten simultaneous connections to the same dynamic WEBDEV site (GO icon).

The WEBDEV administrator is automatically started during the first test of the site. For tests: disconnection time, ...

This refresh operation is used to check the access to the site during the update.

Use the "Apply" button.

For more details about the deployment tools, see their help.
### Part 8: Hosting WEBDEV sites

#### 2.4 General configuration

The general configuration corresponds to the default connection options for the WEBDEV sites found on the server and managed by the administrator. The "Configuration" tab is used to:

- Configure the connections.
- Configure the sessions and the requests.
- Manage the log file to get the traffic statistics.
- Managing 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 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 user connections on a site**: Maximum number of simultaneous connections allowed for a given Web user to each site managed by the administrator. If this parameter is set to zero, a Web user trying to simultaneously connect to the same site will be automatically disconnected then reconnected. If this parameter is greater than zero, an error message will be automatically displayed after the x simultaneous connections. This number can be modified for each site ("Sites" tab).
- **Allow the pre-launched sessions**: Allows to manage the pre-launched sessions on the WEBDEV application server. This option is used to optimize the connection time to the WEBDEV sites and Webservices found on the server.

See the online help for more details.

The WEBDEV administrator is used to manage a specific log file allowing you to follow the traffic statistics.

- **Generate a log file to manage the traffic statistics (.log)**: Allows you to generate a log file (".LOG" extension) used to see the traffic statistics of WEBDEV sites via WDSStatistic. See the help about WDSStatistic 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.

Managing the remote debugging

The WEBDEV administrator can allow (or not) the remote debugging of dynamic WEBDEV sites. If the remote debugging is allowed, the following parameters must be specified:

- **Main port (27 271 by default)**.

- **Duration of AWP contexts**: Validity period of AWP contexts. As soon as the specified duration is over and if no new request was performed, the context file is deleted.

- **Duration of Webservices 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 a response. When this time is exceeded, an error page is displayed but the Web user is not disconnected.

- **Maximum task duration**: Used to limit the runtime duration of a WEBDEV scheduled task or delayed task. The task will be automatically stopped when this duration is exceeded.

- **Allow the pre-launched sessions**: Makes it possible to manage the pre-launched sessions on the WEBDEV application server. This option is used to optimize the connection time to the WEBDEV sites and Webservices found on the server.

- **Range of ports for sessions (between 27 282 and 27 289 by default)**.

#### 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 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.
- Enable (or not) the debugging of the server.
- Range of ports for sessions (between 27 282 and 27 289 by default).

The "Diagnostic" button is used to check the Web server used. See the online help for more details.

**Server used**

The "Server" button is used to choose one of the Web servers installed on the current computer to run the WEBDEV sites. 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.

For more details on testing a WEBDEV site, see "Test of a site in practice", page 196.

**Searching for expired pages**

This option is used to enable and configure the search for expired pages on the WEBDEV sites managed by 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 messages**

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 must be enabled when starting the email session (with EmailStartSMTPSession or EmailStartSession).

If the asynchronous mode is enabled, all outgoing 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 to send emails or EmailStartSession). This mode is ignored in the other cases.

Sockets

If "Allow the server sockets" is checked, your sites will be able to handle the server sockets (via the Socketxxx functions of WLanguage).

Forbidden 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).

Forbidden the access to AWP context identifiers from JavaScript

In an AWP site, the site context is stored on the server. The identifier of this context is sent and stored in the browser via a cookie. If the option "Forbidden 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.

Managing fCopyFileWebFolder and fDeleteFichierWebFolder

The option "Allow fCopyFileWebFolder and fDeleteFichierWebFolder" 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 (WD240ADMIN.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 available in deployment version. They allow the hosting company to authorize and configure the site setups and updates remotely.

The option "Lock the server (for an update)" is used to easily update a site by blocking 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.
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).
  - Characteristics:
    - The hosting company provides a server to several clients.
    - The hosting company:
      - configures the FTP server (required only for FTP deployments).
      - defines the maximum number of connections for all the dynamic sites belonging to the same client (via the WEBDEV account manager).
    - The setup and updates of a dynamic or static site can be performed by FTP.

- **dedicated server:**
  - the server is reserved for the client. The hosting company provides the access to Internet (IP address, wire, etc.).
  - The computer is either rented by the hosting company, or provided by the client.
  - Characteristics:
    - The hosting company provides the access to Internet (IP address, wire, etc.).
    - The client must:
      - provide the server (or rent it),
      - configure the computer (system, etc.),
      - install the Web server,
      - install and configure WEBDEV sites (via HTTP, FTP or 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.

<table>
<thead>
<tr>
<th>Server</th>
<th>Benefits/Drawbacks</th>
<th>When should this server be used?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared</td>
<td>Main benefits:</td>
<td>A shared server is recommended for:</td>
</tr>
<tr>
<td></td>
<td>• Lower cost.</td>
<td>• sites with few hits (up to 20 or 30 simultaneous connections)</td>
</tr>
<tr>
<td></td>
<td>• Server managed by the hosting company.</td>
<td>• sites requiring little memory (no large calculations performed on the server).</td>
</tr>
<tr>
<td></td>
<td>Main drawbacks:</td>
<td>• Sites in startup phase.</td>
</tr>
<tr>
<td></td>
<td>• Memory resources shared by all the sites found on the server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If a site is locked, all the sites found on the server are locked.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Stand-alone executables can rarely be used.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-secured access to data: Unauthorized people can access the data: the hosting</td>
<td></td>
</tr>
<tr>
<td></td>
<td>company and possibly other clients if the server is not configured properly.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tip: encrypt the data files to avoid any unauthorized use.</td>
<td></td>
</tr>
<tr>
<td>Dedicated</td>
<td>Main benefits:</td>
<td>A dedicated server is recommended for:</td>
</tr>
<tr>
<td></td>
<td>• The server resources are available for the sites installed on this computer.</td>
<td>• sites with many hits,</td>
</tr>
<tr>
<td></td>
<td>• Custom management of data download:</td>
<td>• sites requiring a lot of disk space and memory,</td>
</tr>
<tr>
<td></td>
<td>• replication by emails,</td>
<td>• sites managing sensitive data,</td>
</tr>
<tr>
<td></td>
<td>• WDREPLiC,</td>
<td>• dynamic sites.</td>
</tr>
<tr>
<td></td>
<td>• copy of files by FTP.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Main drawbacks:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• High cost.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Server entirely managed by the client in most cases.</td>
<td></td>
</tr>
</tbody>
</table>

Hosting Control Center

Mainly intended for the hosting companies and for the Webmasters, the Hosting Control Center helps you 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 the following elements is centralized:
- Windows accounts,
- Groups of FTP users,
- Groups of WEBDEV clients,
- Home Directory,
- FTP alias,
- WEBDEV accounts,
- Data directories,
- Virtual Web sites, ...

The WEBDEV Hosting Control Center operates in the Windows environment with the IIS Web server (Internet Information Server, all versions from version 5).
Dynamic site on specific configurations

1. Setup on a Web server in Windows (2000 or later) or a Linux server

The configuration is as follows:

- the WEBDEV application server
- the WEBDEV site (Session or AWP)
- the Web server on Windows
- the static pages
- Unix Web server
- the Web server on Linux

1. Opening a static page. The static page is found on the UNIX server.
2. Starting the dynamic WEBDEV site (found on the Windows server) via a link found in the static HTML page.
3. Standard operating mode of the WEBDEV site.

Note: the same configuration can be 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 the 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 viewed, which applications have been used and from which site were they coming from?

WEBDEV is supplied with a statistical tool for the dynamic pages: WDStatistic. This tool can be installed on any computer.

Examples of statistics calculated by WDStatistic: number of connections per day, visited pages, operating systems and browsers of Web users, ...

Examples of statistics 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 allows you 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.
- Running 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
- Test of news server
- Test of WINDEV application
- Test of SNMP
- Test of WINDEV application
- Custom test (in WLanguage)
### WLWage functions specific to WEBDEV 24

(Details in the WLanguage online help)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddFavorite</td>
<td>Adds an Internet address into the list of favorites for the Web user.</td>
</tr>
<tr>
<td>AJAXAsynchronousCallPending</td>
<td>Used to find out whether a server procedure called by <code>AJAXExecuteAsynchronous</code> is currently run.</td>
</tr>
<tr>
<td>AJAXAvailable</td>
<td>Used to find out whether the AJAX technology is supported by the current browser.</td>
</tr>
<tr>
<td>AJAXCancelAsynchronousCall</td>
<td>Cancels the automatic execution of the browser procedure called by <code>AJAXExecuteAsynchronous</code>.</td>
</tr>
<tr>
<td>AJAXExecute</td>
<td>Runs a server procedure without refreshing the page.</td>
</tr>
<tr>
<td>AJAXExecuteAsynchronous</td>
<td>Runs a server procedure without refreshing the page.</td>
</tr>
<tr>
<td>AppServerAddScheduledTask</td>
<td>Adds a new scheduled task to the application (site or Webservice).</td>
</tr>
<tr>
<td>AppServerDeleteScheduledTask</td>
<td>Deletes a scheduled task associated with the application (site or Webservice).</td>
</tr>
<tr>
<td>AppServerInfoScheduledTask</td>
<td>Reads the description of a scheduled task.</td>
</tr>
<tr>
<td>AppServerListScheduledTask</td>
<td>Returns the list of scheduled tasks for the application (site or Webservice).</td>
</tr>
<tr>
<td>AppServerModifyScheduledTask</td>
<td>Modifies an existing scheduled task.</td>
</tr>
<tr>
<td>AppServerRunScheduledTask</td>
<td>Immediately starts the execution of a scheduled task on a WEBDEV Application Server.</td>
</tr>
<tr>
<td>ASPDisplay</td>
<td>Calls an external ASP script and returns the result page in the current browser window.</td>
</tr>
<tr>
<td>ASPExecute</td>
<td>Calls an external .asp script and returns the result in a string.</td>
</tr>
<tr>
<td>AssistedInputConfigure</td>
<td>Used to configure the filter and the opening of assisted input.</td>
</tr>
<tr>
<td>BannerFirst</td>
<td>Displays the first plane of a Scrolling Banner control.</td>
</tr>
<tr>
<td>BannerNext</td>
<td>Displays the next plane of a Scrolling Banner control.</td>
</tr>
<tr>
<td>BannerPrevious</td>
<td>Displays the previous plane of a Scrolling Banner control.</td>
</tr>
<tr>
<td>BannerStopScrolling</td>
<td>Stops the automatic scroll on a Scrolling Banner control.</td>
</tr>
</tbody>
</table>
### Concepts

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BreadcrumbAddLink</td>
<td>Adds a link into a Breadcrumb control.</td>
</tr>
<tr>
<td>BreadcrumbDeleteAll</td>
<td>Deletes all links from a Breadcrumb control.</td>
</tr>
<tr>
<td>BreadcrumbDeleteLink</td>
<td>Deletes a link from a Breadcrumb control.</td>
</tr>
<tr>
<td>BreadcrumbInsertLink</td>
<td>Inserts a link into a Breadcrumb control.</td>
</tr>
<tr>
<td>BreadcrumbModifyLink</td>
<td>Modifies a link in a Breadcrumb control.</td>
</tr>
<tr>
<td>BrowserBlade</td>
<td>Returns the number of the blade displayed in a page in Responsive Web Design mode.</td>
</tr>
<tr>
<td>BrowserClose</td>
<td>Closes the current browser window and stops the execution of the current code.</td>
</tr>
<tr>
<td>BrowserHeight</td>
<td>Returns the height (in pixels) of the client area where the page is displayed.</td>
</tr>
<tr>
<td>BrowserIPAddress</td>
<td>Returns the IP address of the computer of the Web user connected to the WEBDEV site.</td>
</tr>
<tr>
<td>BrowserIsConnected</td>
<td>Indicates whether the browser is connected to the network.</td>
</tr>
<tr>
<td>BrowserMobile</td>
<td>Allows you to find out whether the browser used by the Web user is started on a mobile device.</td>
</tr>
<tr>
<td>BrowserName</td>
<td>Returns the name of the browser of the Web user (&quot;user agent&quot;).</td>
</tr>
<tr>
<td>BrowserOpen</td>
<td>Opens a new browser window.</td>
</tr>
<tr>
<td>BrowserOS</td>
<td>Indicates the operating system announced by the browser of the Web user.</td>
</tr>
<tr>
<td>BrowserPlatform</td>
<td>Returns the platform of the browser used by the Web user.</td>
</tr>
<tr>
<td>BrowserRunApp</td>
<td>Opens the default Web browser of the current device.</td>
</tr>
<tr>
<td>BrowserType</td>
<td>Returns the type of browser used by the Web user.</td>
</tr>
<tr>
<td>BrowserWidth</td>
<td>Returns the width (in pixels) of the client area where the page is displayed.</td>
</tr>
<tr>
<td>CancelAWPContext</td>
<td>Deletes from the AWP context a variable that was added by DeclareAWPContext.</td>
</tr>
<tr>
<td>CapsLockVerify</td>
<td>Checks whether the CapsLock key is pressed.</td>
</tr>
<tr>
<td>CaptchaDisplay</td>
<td>Displays a new Captcha in a Captcha control.</td>
</tr>
<tr>
<td>CaptchaVerify</td>
<td>Checks whether the value typed by the user corresponds to the string displayed in a Captcha control.</td>
</tr>
<tr>
<td>CellCloseDialog</td>
<td>Hides a cell displayed in the page via CellDisplayDialog.</td>
</tr>
<tr>
<td>CellDisplayDialog</td>
<td>Displays a cell in a page with a DDW effect (Dim Disabled Windows). This function is used to easily simulate a dialog box in browser code, by using a cell in a page.</td>
</tr>
<tr>
<td>CertificateClientInfo</td>
<td>Returns information about the certificate used by the client computer.</td>
</tr>
<tr>
<td>ChangeAction</td>
<td>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.</td>
</tr>
<tr>
<td>ChangeTarget</td>
<td>Changes the target of a button action.</td>
</tr>
<tr>
<td>ColorPalette</td>
<td>Returns a color found in the current palette.</td>
</tr>
<tr>
<td>ConfigureAWPContext</td>
<td>Configures the operating mode of AWP context.</td>
</tr>
<tr>
<td>ConnectionCount</td>
<td>Returns the number of instances of the WEBDEV site currently run on the server.</td>
</tr>
<tr>
<td>ContextClose</td>
<td>Closes a page context.</td>
</tr>
<tr>
<td>ContextExist</td>
<td>Allows you to find out whether a page context exists on the server (which means whether the page was opened).</td>
</tr>
<tr>
<td>ContextOpen</td>
<td>Opens a new page context without returning the information to the browser.</td>
</tr>
<tr>
<td>CookieRead</td>
<td>Retrieves the value of a cookie saved on the computer of Web user.</td>
</tr>
<tr>
<td>CookieWrite</td>
<td>Writes a cookie onto the computer of the Web user.</td>
</tr>
<tr>
<td>CurrentPage</td>
<td>Returns the name of the page containing the WLanguage code currently run.</td>
</tr>
<tr>
<td>DeclareAWPContext</td>
<td>Used to declare a list of variables whose value will be persistent between the successive displays of AWP pages.</td>
</tr>
<tr>
<td>DynamicServingConfigure</td>
<td>Indicates the environment that will be used by the &quot;Dynamic Serving&quot; to choose the set of pages to display.</td>
</tr>
<tr>
<td>DynamicSiteDisplay</td>
<td>Starts a dynamic WEBDEV site from a static page or from a dynamic WEBDEV page.</td>
</tr>
</tbody>
</table>
| EmailOpenMail                                | Opens the default messaging software:  
• of Web user on the browser computer.  
• of the user on the current Windows computer.  
• on the phone.                                                                                                                                       |
<p>| EmailStatus                                  | Returns the status of an email sent via an SMTP session started in asynchronous mode.                                                                                                                    |</p>
<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExecuteDelayedProcedure</td>
<td>Runs a delayed procedure (in the WEBDEV application server).</td>
</tr>
<tr>
<td>fCopyFileWebFolder</td>
<td>Copies an image file from the data directory of application to the &quot;_WEB&quot; directory of application (or to one of its sub-directories).</td>
</tr>
<tr>
<td>fDeleteFileWebFolder</td>
<td>Deletes an image file from the &quot;_WEB&quot; directory of the application.</td>
</tr>
<tr>
<td>FileDisplay</td>
<td>Displays a file in the browser of Web user.</td>
</tr>
<tr>
<td>FileToPage</td>
<td>Automatically initializes the controls of a page with the values of the associated items in the current record (loaded in memory) of the data file described in the data model editor.</td>
</tr>
<tr>
<td>FolderData</td>
<td>FolderData is kept for backward compatibility.</td>
</tr>
<tr>
<td>FolderWeb</td>
<td>Returns the path of the directory containing:</td>
</tr>
<tr>
<td></td>
<td>• the images,</td>
</tr>
<tr>
<td></td>
<td>• the Javascript files,</td>
</tr>
<tr>
<td></td>
<td>• the Java applet files,</td>
</tr>
<tr>
<td></td>
<td>• the other files accessible from the browser.</td>
</tr>
<tr>
<td>FramesetDisplay</td>
<td>Displays a WEBDEV frameset in the browser of Web user.</td>
</tr>
<tr>
<td>FramesetUse</td>
<td>Displays a WEBDEV frameset in the browser of Web user and closes all the current page and frameset contexts.</td>
</tr>
<tr>
<td>FreeAWPContext</td>
<td>Frees the AWP context in advance (on disk) to allow the other call on the same AWP context to be processed in parallel.</td>
</tr>
<tr>
<td>FullScreenDisable</td>
<td>Disables the &quot;full screen&quot; mode of browser.</td>
</tr>
<tr>
<td>FullScreenEnable</td>
<td>Switches the browser to &quot;full screen&quot; mode. The browser occupies the entire screen.</td>
</tr>
<tr>
<td>fWebDir</td>
<td>Returns the physical name of the directory containing the images, the Javascript files and the Java files of the WEBDEV site.</td>
</tr>
<tr>
<td>GaugeActivate</td>
<td>Enables the refresh of a Progress Bar via Ajax.</td>
</tr>
<tr>
<td>GaugeDeactivate</td>
<td>Stops the periodic refresh of a Progress Bar via Ajax.</td>
</tr>
<tr>
<td>GaugeExecute</td>
<td>Starts a long browser process and fill a Progress Bar via Ajax according to the progress of this process.</td>
</tr>
<tr>
<td>GglAnalyticsAddEvent</td>
<td>Adds an event beside Google Analytics.</td>
</tr>
<tr>
<td>GglAnalyticsAddException</td>
<td>Adds an exception beside Google Analytics.</td>
</tr>
<tr>
<td>GglAnalyticsAddPage</td>
<td>Adds a page beside Google Analytics.</td>
</tr>
<tr>
<td>GglAnalyticsAddSocialNetworkAction</td>
<td>Adds a social network action beside Google Analytics.</td>
</tr>
<tr>
<td>GglAnalyticsAddTiming</td>
<td>Adds a duration beside Google Analytics.</td>
</tr>
<tr>
<td>gpwOpenConfiguration</td>
<td>Opens the window or page for configuring the user groupware.</td>
</tr>
<tr>
<td>grImageSize</td>
<td>Defines the size of the image containing the chart.</td>
</tr>
<tr>
<td>HTMLClassAdd</td>
<td>Adds a class into the HTML classes of a control.</td>
</tr>
<tr>
<td>HTMLClassDelete</td>
<td>Deletes a class from the HTML classes of a control.</td>
</tr>
<tr>
<td>HTMLClassToggle</td>
<td>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.</td>
</tr>
<tr>
<td>IdentifierAWPContext</td>
<td>Returns the identifier of AWP context.</td>
</tr>
<tr>
<td>iDirImageHTML</td>
<td>Used to select the directory in which the images must be generated during a print in HTML format.</td>
</tr>
<tr>
<td>ImageArea</td>
<td>Returns the number of the image area clicked by the Web user.</td>
</tr>
<tr>
<td>ImageFirst</td>
<td>Displays the first image for the scroll.</td>
</tr>
<tr>
<td>ImageLast</td>
<td>Displays the last image for the scroll.</td>
</tr>
<tr>
<td>ImageNext</td>
<td>Displays the next image for the scroll.</td>
</tr>
<tr>
<td>ImageOccurrence</td>
<td>Returns the total number of images taken into account by the scroll.</td>
</tr>
<tr>
<td>ImagePrevious</td>
<td>Displays the previous image of scrolling.</td>
</tr>
<tr>
<td>ImageScrollingPosition</td>
<td>Returns the image displayed.</td>
</tr>
<tr>
<td>ImageStartScrolling</td>
<td>Starts the automatic scroll of images.</td>
</tr>
<tr>
<td>ImageStopScrolling</td>
<td>Stops the automatic scroll of images.</td>
</tr>
<tr>
<td>ImageXPos</td>
<td>Returns the horizontal position (X) of the mouse cursor in relation to the specified image control.</td>
</tr>
<tr>
<td>ImageYPos</td>
<td>Returns the vertical position (Y) of the mouse cursor in relation to the specified image control.</td>
</tr>
<tr>
<td>JQuery</td>
<td>Runs a Javascript method (or several chained methods) of the jQuery library on a page element.</td>
</tr>
<tr>
<td>JQueryExecute</td>
<td>Runs a Javascript method of the jQuery library on a page element.</td>
</tr>
<tr>
<td>JSEndEvent</td>
<td>Deletes the association between a WLanguage browser function and an event (implemented by JSEvent).</td>
</tr>
<tr>
<td>JSEvent</td>
<td>Associates a browser procedure with an event on an object in browser code.</td>
</tr>
<tr>
<td>JSInfoEvent</td>
<td>Used to handle the JavaScript properties of the browser event that triggered the code execution.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>JSInterruptEvent</td>
<td>Interrupts the process of current event.</td>
</tr>
<tr>
<td>JSMethod</td>
<td>Used to run a Javascript method on an element found in the current page.</td>
</tr>
<tr>
<td>JSONExecute</td>
<td>Calls a server URL that returns data in JSON format (JavaScript Object Notation).</td>
</tr>
<tr>
<td>JSONExecuteExternal</td>
<td>Calls an external server URL that returns data in JSON format (JavaScript Object Notation).</td>
</tr>
<tr>
<td>JSPROPERTY</td>
<td>Used to handle specific features on the objects found in the current page.</td>
</tr>
<tr>
<td>LocalStorageAdd</td>
<td>Adds a value to the local storage.</td>
</tr>
<tr>
<td>LocalStorageAvailable</td>
<td>Indicates whether the local storage is available or not for the current browser.</td>
</tr>
<tr>
<td>LocalStorageDelete</td>
<td>Deletes a value from the local storage.</td>
</tr>
<tr>
<td>LocalStorageDeleteAll</td>
<td>Deletes all the values from the local storage.</td>
</tr>
<tr>
<td>LocalStorageGet</td>
<td>Retrieves a value from the local storage.</td>
</tr>
<tr>
<td>LocalStorageOccurrence</td>
<td>Returns the number of values for the local storage.</td>
</tr>
<tr>
<td>LocalStorageValueName</td>
<td>Returns the name of a value for the local storage.</td>
</tr>
<tr>
<td>MenuAddPopup</td>
<td>Transforms a menu option of a page in order for this option to open a popup.</td>
</tr>
<tr>
<td>MenuAddURLOption</td>
<td>Adds a new menu option at the end of a page menu. This menu option is used to display the page corresponding to the specified URL.</td>
</tr>
<tr>
<td>PageActivateDDW</td>
<td>Enables or disables the DDW (Dim Disabled Window) when displaying a modal page (the pages used to communicate with the user for example).</td>
</tr>
<tr>
<td>PageAddress</td>
<td>Used to find out the Internet address of a page.</td>
</tr>
<tr>
<td>PageCloseDialog</td>
<td>Closes the current page. This page was opened by PageDisplayDialog. A return value can be returned to the calling page.</td>
</tr>
<tr>
<td>PageDisplay</td>
<td>Displays a site page in the browser of Web user.</td>
</tr>
<tr>
<td>PageDisplayDialog</td>
<td>Displays a page in modal mode. This function is used to establish a dialog with the user. The page is displayed in the foreground while the opening page is displayed in the background, grayed by the DDW mechanism.</td>
</tr>
<tr>
<td>PageExist</td>
<td>Checks whether the specified page is currently displayed in the browser of the Web user. Used to find out whether a page is displayed in a frame other than the current one.</td>
</tr>
<tr>
<td>PageInitialization</td>
<td>Resets to zero (or not) the controls found in the current page and starts the processes for initializing the controls and the page.</td>
</tr>
<tr>
<td>PageParameter</td>
<td>Returns the value of a parameter passed to the current page.</td>
</tr>
<tr>
<td>PagePosition</td>
<td>Scrolls a page up to position a control in the visible section of the page (top) in the browser.</td>
</tr>
<tr>
<td>PageRateDDW</td>
<td>Allows you to find out and modify the rate of gray for the pages that use DDW (Dim disabled pages).</td>
</tr>
<tr>
<td>PageReadHeaderHTTP</td>
<td>Reads and returns one or more HTTP headers received by the current page (these headers are sent by the client to the server in the current request).</td>
</tr>
<tr>
<td>PageRefresh</td>
<td>Refreshes a page displayed in the browser of Web user from the context found on the server.</td>
</tr>
<tr>
<td>PageSubmit</td>
<td>Validates the specified page and starts the execution of a button.</td>
</tr>
<tr>
<td>PageToASP</td>
<td>Sends the data found in a page currently displayed in the browser to an ASP server.</td>
</tr>
<tr>
<td>PageToFile</td>
<td>Automatically initializes the memory value of the items of a data file with the value of the controls found in the page.</td>
</tr>
<tr>
<td>PageToJSP</td>
<td>Sends the data found in a page currently displayed in the browser to a JSP server.</td>
</tr>
<tr>
<td>PageToPHP</td>
<td>Sends the data found in a page currently displayed in the browser to a PHP server.</td>
</tr>
<tr>
<td>PageToSource</td>
<td>Automatically initializes the memory value of the items of a data file and/or the value of the WLanguage variables with the value of the controls found in the page.</td>
</tr>
<tr>
<td>PageUse</td>
<td>Displays a WEBDEV page in the browser of the Web user and closes all the current page contexts.</td>
</tr>
<tr>
<td>PageVisible</td>
<td>Indicates whether the page is visible to the user.</td>
</tr>
<tr>
<td>PageWriteHeaderHTTP</td>
<td>Adds an additional HTTP header that will be returned to the browser.</td>
</tr>
<tr>
<td>PHPDisplay</td>
<td>Calls an external PHP script and returns the result page in the current browser window.</td>
</tr>
<tr>
<td>PHPExecute</td>
<td>Calls an external .php script and returns the result in a string.</td>
</tr>
</tbody>
</table>
### Concepts

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PopupClose</strong></td>
<td>Hides a popup displayed in the page via <strong>PopupDisplay</strong>.</td>
</tr>
<tr>
<td><strong>PopupDisplay</strong></td>
<td>Displays a popup in a page with a DDW effect (Dim Disabled Windows).</td>
</tr>
<tr>
<td><strong>PreviousPage</strong></td>
<td>Returns the name of the previous page.</td>
</tr>
<tr>
<td><strong>rssDisplay</strong></td>
<td>Builds a RSS stream and returns the content of the RSS stream to the client.</td>
</tr>
<tr>
<td><strong>SaaSCheckService</strong></td>
<td>Checks the access rights of the user to a service of the SaaS site.</td>
</tr>
<tr>
<td><strong>SaaSClientConnexion</strong></td>
<td>Returns the characteristics of the connection to the database of the client.</td>
</tr>
<tr>
<td><strong>SaaSConnect</strong></td>
<td>Authenticates a user of a SaaS site beside the SaaS webservice that manages the site.</td>
</tr>
<tr>
<td><strong>SaaSConnectedUser</strong></td>
<td>Returns the user currently connected via <strong>SaaSConnect</strong>.</td>
</tr>
<tr>
<td><strong>SaaSDisconnect</strong></td>
<td>Disconnect the user of a SaaS site beside the SaaS webservice that manages the site.</td>
</tr>
<tr>
<td><strong>SaaSIsConnected</strong></td>
<td>Defines whether the connection to the SaaS webservice is still established.</td>
</tr>
<tr>
<td><strong>SaaSReadSiteParameter</strong></td>
<td>Reads an information specific to the client for the current SaaS site.</td>
</tr>
<tr>
<td><strong>SaaSWriteSiteParameter</strong></td>
<td>Saves a specific information for a SaaS site in the configuration of a client account.</td>
</tr>
<tr>
<td><strong>ScriptDisplay</strong></td>
<td>Calls an external script or page (.php,.asp,.mhtml or .mht) and returns the result page in the current browser window.</td>
</tr>
<tr>
<td><strong>ScriptExecute</strong></td>
<td>Calls an external script (.asp or .php) and returns the result in a string.</td>
</tr>
<tr>
<td><strong>SiteAddress</strong></td>
<td>Returns the Internet address for connecting to a dynamic WEBDEV site found on the same server.</td>
</tr>
<tr>
<td><strong>SourceToPage</strong></td>
<td>Automatically initializes the controls of a page with:</td>
</tr>
<tr>
<td></td>
<td>• the values of the associated items in the current record (loaded in memory) of the HSQL data file described in the data model editor.</td>
</tr>
<tr>
<td></td>
<td>• the values of the associated WLanguage variables.</td>
</tr>
<tr>
<td><strong>SSLActive</strong></td>
<td>Used to enable or disable the secure SSL mode.</td>
</tr>
<tr>
<td><strong>StringDisplay</strong></td>
<td>Returns a specific string (or a buffer) to the client browser in response to a request.</td>
</tr>
<tr>
<td><strong>SysVersion</strong></td>
<td>Returns information about the PHP version used on the current server.</td>
</tr>
<tr>
<td><strong>ToastDisplayPopup</strong></td>
<td>Displays a popup page during a given duration in order to display a &quot;Toast&quot; message.</td>
</tr>
<tr>
<td><strong>UploadCopyFile</strong></td>
<td>Saves on the server a file &quot;uploaded&quot; by the Web user.</td>
</tr>
<tr>
<td><strong>UploadCurrentFile</strong></td>
<td>Indicates the file currently uploaded via the Upload control.</td>
</tr>
<tr>
<td><strong>UploadCurrentFileSize</strong></td>
<td>Returns the total size (in bytes) of the file currently uploaded via an Upload control.</td>
</tr>
<tr>
<td><strong>UploadCurrentFileSizeSent</strong></td>
<td>Returns the size (in bytes) already sent for the file currently uploaded via an Upload control.</td>
</tr>
<tr>
<td><strong>UploadDelete</strong></td>
<td>Deletes a file from the list of files to upload: the file will not be uploaded on the server.</td>
</tr>
<tr>
<td><strong>UploadDeleteAll</strong></td>
<td>Clears the list of files to upload: no file will be uploaded on the server.</td>
</tr>
<tr>
<td><strong>UploadFileName</strong></td>
<td>Returns the name of a file &quot;uploaded&quot; by the Web user. This upload may have been performed:</td>
</tr>
<tr>
<td></td>
<td>• via an &quot;Upload&quot; edit control.</td>
</tr>
<tr>
<td></td>
<td>• via a single-file or multi-file Upload control.</td>
</tr>
<tr>
<td><strong>UploadFileSize</strong></td>
<td>Returns the total size (in bytes) of a file found in an Upload control.</td>
</tr>
<tr>
<td><strong>UploadSize</strong></td>
<td>Returns the total size (in bytes) of the file currently uploaded via an Upload control.</td>
</tr>
<tr>
<td><strong>UploadSizeSent</strong></td>
<td>Returns the total size (in bytes) of files already sent by the current upload via an Upload control.</td>
</tr>
<tr>
<td><strong>UploadStart</strong></td>
<td>Starts sending the selected files into an upload control.</td>
</tr>
</tbody>
</table>
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 WEBDEV.

Additional examples can be downloaded from our site (www.windevcom).

### Complete examples

<table>
<thead>
<tr>
<th>Example name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eBusiness</td>
<td>Full eCommerce site that can be customized via a Web management interface.</td>
</tr>
<tr>
<td></td>
<td>• The showroom/payment section of the site is developed in AWP in order to get the best possible referencing of products.</td>
</tr>
<tr>
<td></td>
<td>• The administration section of the site is developed in standard WEBDEV session to guarantee the maximum security.</td>
</tr>
<tr>
<td>Honolulu</td>
<td>Free Intranet portal: messaging, forum, blogs, ...</td>
</tr>
<tr>
<td>Photo_Gallery</td>
<td>This example is a site of photo gallery.</td>
</tr>
<tr>
<td>WAds</td>
<td>Management of classified ads. Multi-criteria search, addition, modification and deletion of classifieds, as well user management.</td>
</tr>
<tr>
<td>WebFleet</td>
<td>Simulates the management of a computer fleet.</td>
</tr>
<tr>
<td>webmillion</td>
<td>Uses a data file containing more than 1 million records.</td>
</tr>
<tr>
<td>WW_Association_AWP</td>
<td>Association site made of 2 main sections:</td>
</tr>
<tr>
<td></td>
<td>• the “visitor” section, built in AWP mode therefore can be referenced.</td>
</tr>
<tr>
<td></td>
<td>• the “member space” section, built with WEBDEV in standard mode and therefore secured.</td>
</tr>
<tr>
<td></td>
<td>This site presents news, a photo gallery, the association overview, ...</td>
</tr>
<tr>
<td>WW_Association_PHP</td>
<td>Association site in PHP WEBDEV made of 2 main sections:</td>
</tr>
<tr>
<td></td>
<td>• the visitor section, accessible to all.</td>
</tr>
<tr>
<td></td>
<td>• the management section, reserved to the members of association.</td>
</tr>
<tr>
<td></td>
<td>This site presents news, a photo gallery, the association overview, ...</td>
</tr>
<tr>
<td>WW_Blog_AWP</td>
<td>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.</td>
</tr>
<tr>
<td>WW_Blogs_php</td>
<td>This example is a PHP site for blog management.</td>
</tr>
<tr>
<td>WW_CMS</td>
<td>This example is a CMS (Content Management System), typically a site for displaying articles.</td>
</tr>
<tr>
<td>WW_FAQ</td>
<td>This Responsive Web Design example proposes the main features of a FAQ site.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WW_Forum_AWP</td>
<td>This example proposes the main features of a user forum (creation of forums, topics and messages, moderation by an administrator, ...).</td>
</tr>
<tr>
<td>WW_Forum_PHP</td>
<td>This example proposes the main features of a user forum (creation of forums, topics and messages, moderation by an administrator, ...).</td>
</tr>
<tr>
<td>WW_Loan</td>
<td>This example is used to simulate loans and to calculate:</td>
</tr>
<tr>
<td></td>
<td>• the amount of monthly payments for a given loan.</td>
</tr>
<tr>
<td></td>
<td>• the amount that can be borrowed for a given monthly payment.</td>
</tr>
<tr>
<td></td>
<td>• the income of an investment for a given monthly payment.</td>
</tr>
<tr>
<td></td>
<td>The amortization table is displayed for each case.</td>
</tr>
<tr>
<td>WW_MotorEquipment</td>
<td>This example represents an online store.</td>
</tr>
<tr>
<td></td>
<td>It contains the “front-office” section (overview of products), the basket section, the secure payment.</td>
</tr>
<tr>
<td>WW_Newsletter</td>
<td>Used to create a site for managing newsletters. It allows you to create, write, and manage the sending of mass newsletters to subscribers.</td>
</tr>
<tr>
<td>WW_Overview_Mobile_Application</td>
<td>Responsive Web site presenting a Mobile application that includes a static page that is entirely responsive.</td>
</tr>
<tr>
<td>WW_Precilia_Sport</td>
<td>This example is a Web site with a store section.</td>
</tr>
<tr>
<td></td>
<td>The site occupies the full browser width with a background image.</td>
</tr>
<tr>
<td>WW_Precilia_Winter</td>
<td>This example is an online site for skiing equipment.</td>
</tr>
<tr>
<td>WW_PreciliaTShirts</td>
<td>This example is an online site for selling tee-shirts.</td>
</tr>
<tr>
<td></td>
<td>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.).</td>
</tr>
<tr>
<td>WW_Rewali</td>
<td>Online travel booking.</td>
</tr>
<tr>
<td>WW_Schedule</td>
<td>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.</td>
</tr>
<tr>
<td>WW_SchedulingActivities_Club</td>
<td>This example presents a site for a structure proposing activities according to a specific schedule; a sport club for example.</td>
</tr>
<tr>
<td>WW_Static_Company</td>
<td>This example is a static site for presenting companies, a “showroom” site.</td>
</tr>
<tr>
<td>WW_Static_Crafts</td>
<td>This example is a static site for presenting a crafts workshop or a company working in the crafts field</td>
</tr>
<tr>
<td>WW_Static_Restaurant</td>
<td>This example is a static site for presenting a restaurant, a “showroom” site.</td>
</tr>
<tr>
<td>WW_TamesShop</td>
<td>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.</td>
</tr>
<tr>
<td>WW_TicketTracker</td>
<td>This example can be used in all areas where it is necessary to follow interventions between several people.</td>
</tr>
<tr>
<td>WW_WebDoc</td>
<td>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.</td>
</tr>
<tr>
<td>WW_WebEstate</td>
<td>This site is Responsive. Its layout fits any browser size automatically.</td>
</tr>
<tr>
<td>WWDLoad</td>
<td>This site is used to propose files that can be downloaded by the users according to the group to which they belong.</td>
</tr>
</tbody>
</table>
Training examples

<table>
<thead>
<tr>
<th>Example name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebApp</td>
<td>This example shows how to use planes in WEBDEV.</td>
</tr>
<tr>
<td>WW_Auto_Error_Management</td>
<td>This example explains how to use the automatic management of WEBDEV errors.</td>
</tr>
<tr>
<td>WW_Auto_Zoom</td>
<td>This example displays an image miniature and when it is clicked, the image is enlarged without disrupting the page layout.</td>
</tr>
<tr>
<td>WW_Dashboard</td>
<td>This example presents the use of the Dashboard control.</td>
</tr>
<tr>
<td>WW_Drawing_HTML5</td>
<td>Drawing functions in browser code for the browsers that support the HTML5 standard.</td>
</tr>
<tr>
<td>WW_Edit_Period</td>
<td>This example shows the integration of javascript code in a WEBDEV site.</td>
</tr>
<tr>
<td>WW_Organizer</td>
<td>Example for using the Organizer control in WEBDEV.</td>
</tr>
<tr>
<td>WW_PayPal_PHP</td>
<td>Explains how to propose a secure payment via the PayPal solution in your WEBDEV PHP sites.</td>
</tr>
<tr>
<td>WW_RSS_Stream</td>
<td>This example explains how to read and display a RSS stream in a WEBDEV page via the RSS type.</td>
</tr>
<tr>
<td>WW_SAASClientSite</td>
<td>This example presents the functions for SaaS management found in WEBDEV.</td>
</tr>
<tr>
<td>WW_Signature</td>
<td>WW_Signature is an example explaining how to draw in HTML5 and how to save the image in browser code.</td>
</tr>
<tr>
<td>WW_Wizard</td>
<td>This example includes a class and a page template that can be re-used to create and manage a Web wizard.</td>
</tr>
<tr>
<td>WWBrowserDialog</td>
<td>This example presents a dialog between two distinct browsers.</td>
</tr>
<tr>
<td>WWPocket</td>
<td>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.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Component name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WW_CloudTags</td>
<td>Used display a cloud of tags in a dynamic WEBDEV application.</td>
</tr>
<tr>
<td>WW_Extraction</td>
<td>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.</td>
</tr>
<tr>
<td>WW_PayBox</td>
<td>Used to install a system for secure payment via PayBox.</td>
</tr>
<tr>
<td>WW_PayPal</td>
<td>Used to include PayPal payment buttons in your WEBDEV pages.</td>
</tr>
<tr>
<td>WW_SecurePayment</td>
<td>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, SPPPlus.</td>
</tr>
</tbody>
</table>

Additional components are available on our site (www.windev.com).