

DEVELOP 10 TIMES FASTER



## In which order should these guides be read?

WINDEV is a powerful tool for developing Windows applications, that provides all the tools required to design and develop applications.

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

- 1** Reading the "Concepts".  
This guide presents the main concepts required to create a WINDEV application.  
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 WINDEV. It allows you to familiarize yourself with the main editors of WINDEV.
- 3** Test of examples  
Check the different examples supplied with WINDEV in the fields you are interested in (sales management, scheduling, ...).

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

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

This documentation is not contractually binding. Modifications may have been made to the software since this guide was published. Check the [online help](#).

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

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

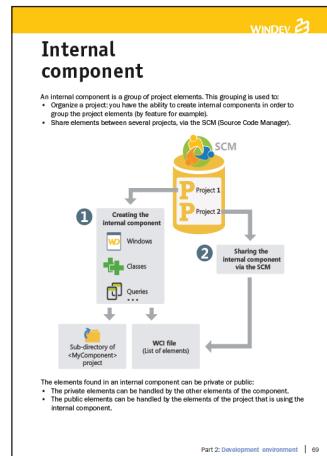
Note: If a difference exists between the guides and the online help, follow the instructions given in the online help.

We hope you enjoy getting started with WINDEV.

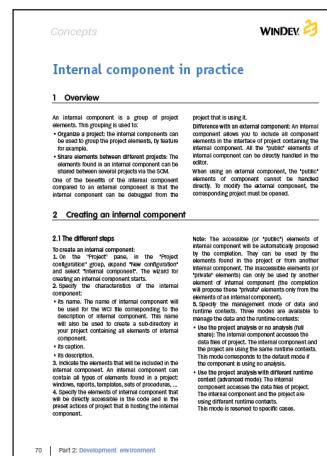
# Organization of the guide

This guide presents the main concepts required to create a WINDEV application. 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 - BASIC CONCEPTS

Project and Analysis.....	9
Development cycle of an application.....	14
GUI: The windows.....	16
Modal, non-modal or MDI windows .....	17
Internal window .....	19
Window templates.....	20
The different types of standard controls.....	34
Control templates.....	49
Edit: The reports.....	50
Report templates .....	51

## PART 2 - DEVELOPMENT ENVIRONMENT

The WINDEV editors.....	55
Project dashboard.....	59
WINDEV/WEBDEV/WINDEV Mobile: 100% compatible .....	60
Project configuration.....	61
Multiple generation.....	62
Custom-folders: Organize your project .....	63
Source Code Manager (SCM).....	64
Internal component .....	71
External component.....	74
Generation modes .....	82
Project Monitoring Center.....	87
Managing requirements .....	88
Managing tasks.....	89
Managing suggestions and incidents.....	90
Managing business rules .....	91

## PART 3 - DATABASES

Analysis: Database structure .....	95
The different types of accessible files .....	104
HFSQL Classic .....	106
HFSQL Client/Server.....	107
HFSQL Client/Server cluster .....	108
HFSQL: the physically created files.....	109
Associating the controls with data.....	110
The queries.....	112
The embedded queries.....	113
The Table/Looper controls .....	114
Universal replication .....	116
3-tier architecture .....	117

**PART 4 - ADVANCED CONCEPTS**

RAD RID .....	121
The UML model .....	122
User groupware .....	129
Multilingual applications .....	137
Optimizing an application .....	143
Distributing "Reports and Queries" .....	145
UMC: User Macro-Code .....	146
The available tests .....	148
The debugger .....	149
Unit tests/Automatic tests .....	151
Unit tests on the executable .....	152

**PART 5 - SETUP**

The WINDEV framework .....	161
The different modes for installing an application .....	163
Monitor your applications .....	172

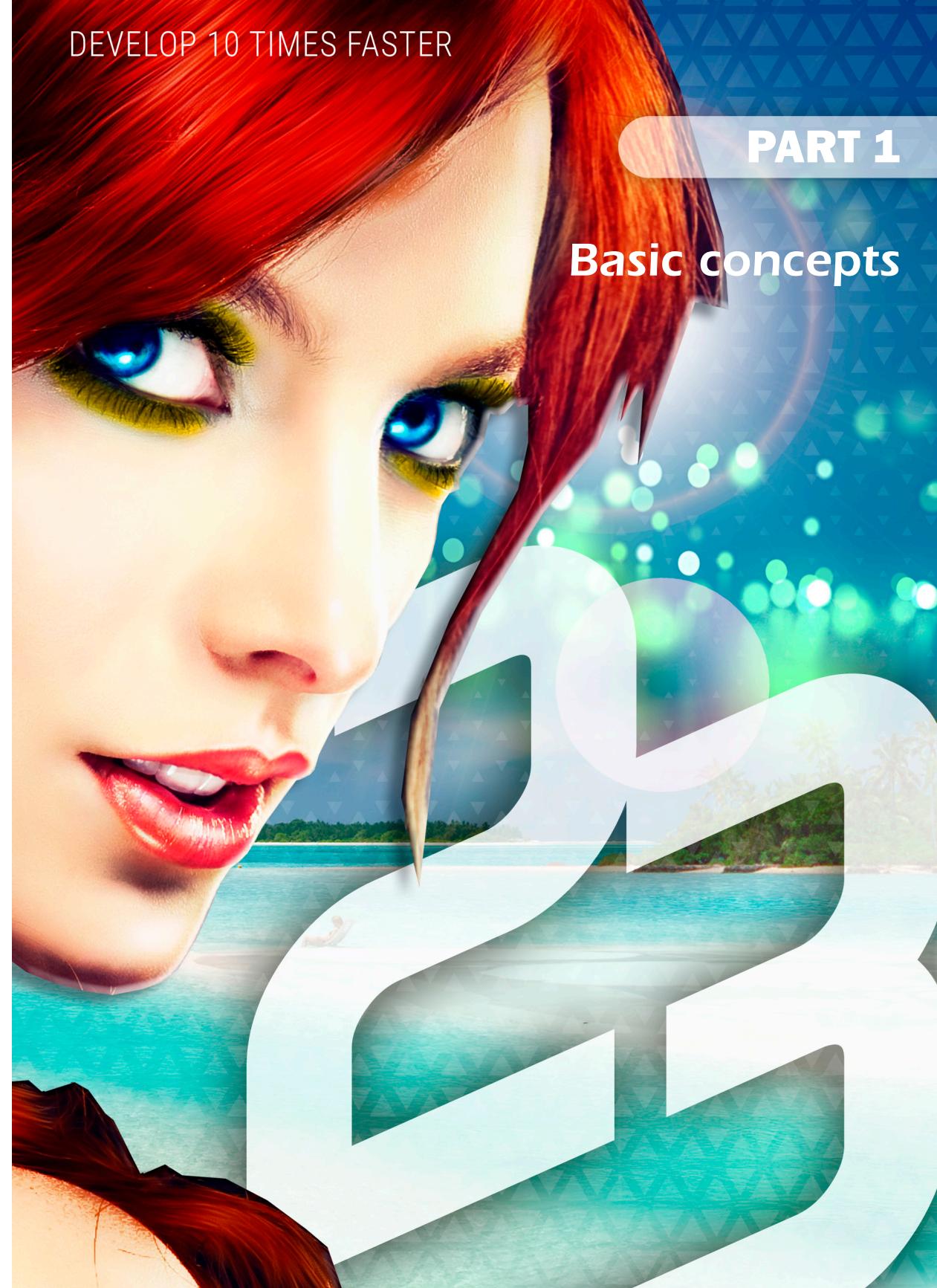
**PART 6 - COMMUNICATION**

Managing emails .....	175
Transferring files by FTP .....	177
Managing sockets .....	179
WINDEV and telephony .....	181
The Webservices .....	182

**PART 7 - APPENDICES**

Components supplied with WINDEV .....	185
Examples supplied with WINDEV .....	189

DEVELOP 10 TIMES FASTER

**PART 1****Basic concepts**

# Project and Analysis

Developing a **Windows application** with WINDEV is based on two main elements: the Project and the Analysis.

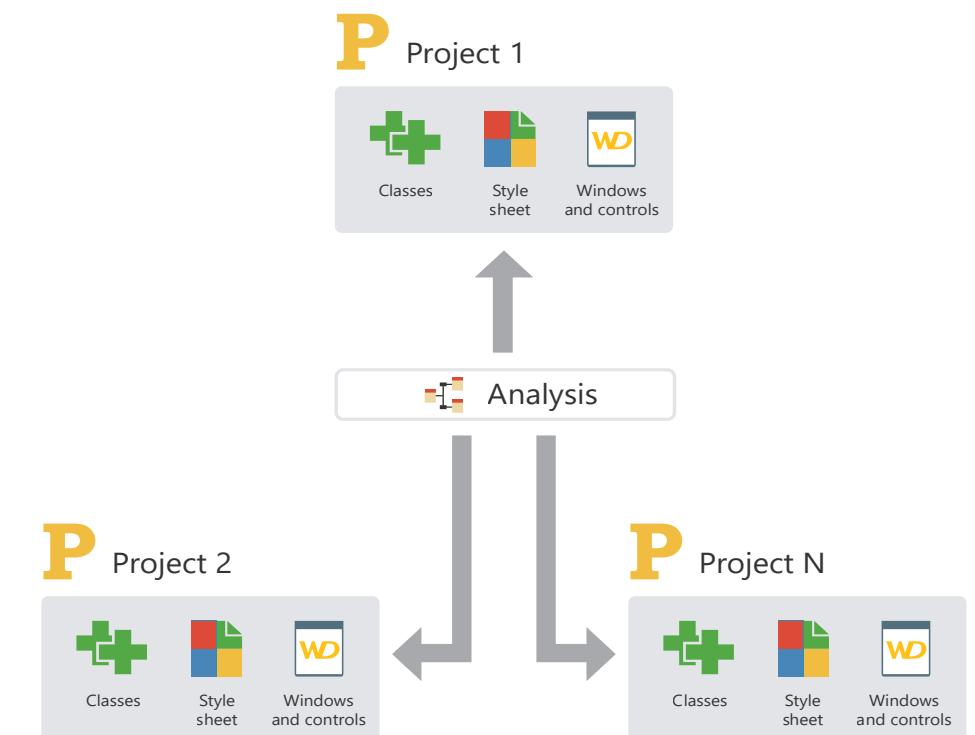
A **WINDEV project** is a set of elements: windows, reports, controls, classes, components, ... The combination of these elements is used to create a Windows application.

A **WINDEV analysis** corresponds to the diagram of data used by the application. The analysis groups the different tables (also called data files) used by the application.

An application 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 an application, the first step consists in creating a project.

When creating a project, several questions will be asked in order for your project to match your

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

**Note:** The analyses are presented in the "Analysis: Database structure", page 95 chapter.

## 2 Creating the project

### To create a project:

1. Click  among the quick access buttons of WINDEV. 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 ".WDP" file. All the elements associated with the project will be created in the specified directory.

- **the type of project.** Several choices are proposed: application, component, Java application, ... These options will be presented later. In most cases, a project is used to generate an application (executable).

- **the operating system** and the runtime mode (for the executables).

- **whether the project will be handled by several developers.** You have the ability to use 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 and windows, ...

- **the style book.**

- **the supported languages.** These languages will be proposed by default as soon as an option can be translated in a control, a window, a report, ...

3. Specify **whether the project must use a database or not.** If yes, it can be an existing database or a new one.

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

### Notes:

- If you have asked to create a database, the corresponding wizard is automatically started.

- The specified information can be modified in the project description. To display the description window of the project, on the "Project" pane, in the "Project" group, click "Description".

## 3 Project dashboard and project graph

### 3.1 Dashboard

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

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

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

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

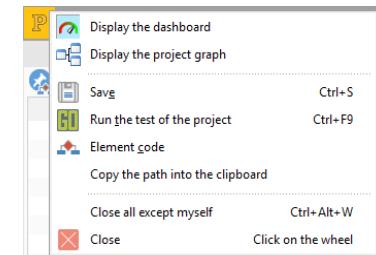
The dashboard gives an answer to all these wishes. The dashboard proposes different indicators called "Widgets". These Widgets can be configured by the user: you have the ability to add or delete some, and to resize them.

You can for example display the project statistics (number of windows, ...), the result of the 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 59) for more details.

### 3.2 Project graph

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

To display the graph of your project:

1. Display the popup menu of project in the bar of opened documents.

2. Select "Display the project graph".

### 3.3 The project elements

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

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

This option is used to:

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

- delete elements from your project. The corresponding files are not physically deleted.

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

## 4 Operations performed on a project

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

- Archive a project.
- Restore a project.

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

See the online help for more details.

## 5 Project documentation

WINDEV allows you to print several documentations that present all the elements (window, page, data file, item, ...) found in the project. To print these documentations:

- on the "Home" pane, in the "General" group, expand  and select "Print the project documentation". The documentation can contain all the characteristics of the project.
- on the "Home" pane, in the "General" group, click . In this case, the documentation contains the characteristics of the current element (window, report, query, ...).

### 5.1 Type of documentation

Several types of documentation are proposed:

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

### Edit mode of documentation

The documentation can be:

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

### 5.2 Print areas

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

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

1. On the "Display" pane, in the "Edit" group, check "Print areas". Borders representing the areas that can be printed in the documentation are displayed in the current editor.
2. Reduce the display of the graphic representation (select "Zoom" in the "Display" pane or press [Ctrl] + Mouse wheel) to view the entire graph.
3. With the mouse:
  - move these borders to the requested location by keeping the mouse button down (the mouse cursor is black).
  - choose the number of pages on which the graphic representation must be printed (the mouse cursor turns into a North-West/South-East double arrow).

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

To do so:

1. Define the print areas.
2. Display the layout options of the documentation: on the "Home" pane, in the "General" group, expand  and select "Documentation layout".
3. Check "Resize the print areas of opened documents".
4. Print the documentation.

# Development cycle of an application

WINDEV covers the entire development cycle of an application:



## Details of different steps:

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

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

**Test and generation step:** WINDEV proposes several automatic test tools to guarantee the reliability of applications and to make sure that no regression occurs during the development steps.

**Deployment step:** Deploying a WINDEV application can be done according to several methods: by download, via a local network (including in Push mode) or via Internet. In any case, the HFSQL data files will be automatically updated according to the evolutions made in the application.

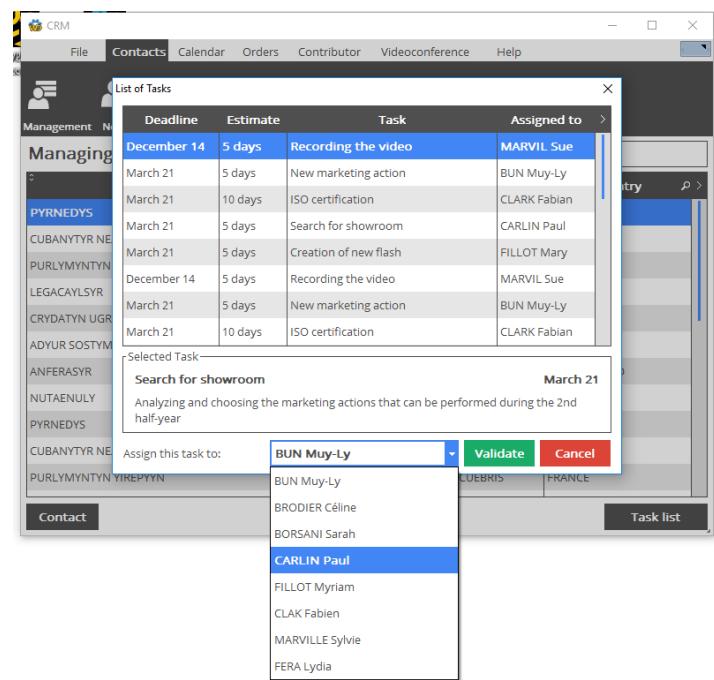
# GUI: The windows

The windows are the foundation of application GUI (Graphical User Interface).

WINDEV includes an advanced window editor allowing you to easily and quickly develop all types of GUI.

Several features enable you to easily obtain intuitive and user-friendly applications:

- powerful controls,
- an anchoring mechanism allowing the GUI to automatically adapt to the size of the display,
- a system for GUI compilation with detection of errors (empty titles, untranslated captions, overlap, ...),
- ...



# Modal, non-modal or MDI windows

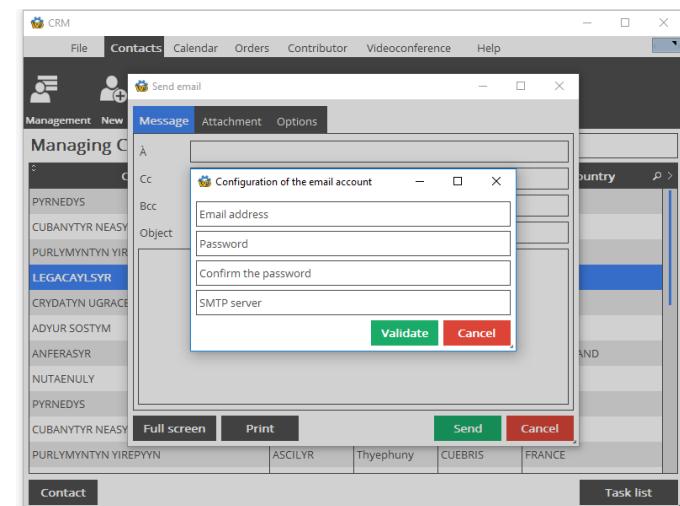
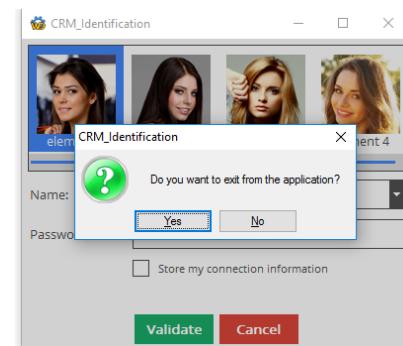
The applications created with WINDEV can use several types of windows (and mix each one of these types).

## Modal windows

A modal window is opened in the foreground of the application and it locks the access to the other windows that are already opened (if any). In most cases, the modal windows are used as:

- main windows of an application,
- dialog boxes whose validation triggers an action.

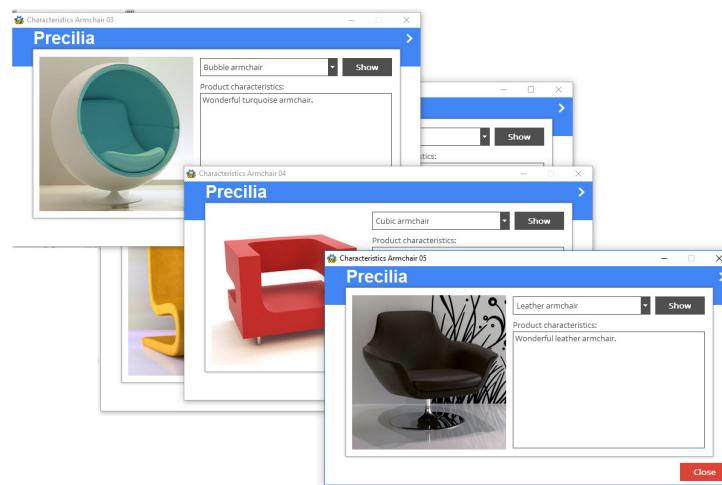
With WINDEV, when a modal window is opened, a gray shadow appears on the windows displayed in the background: it is the DDW feature or Dim Disabled Windows.



## Non-modal windows

A non-modal window is opened in parallel of other application windows. Each one of the windows remains accessible. In most cases, the non-modal windows are used:

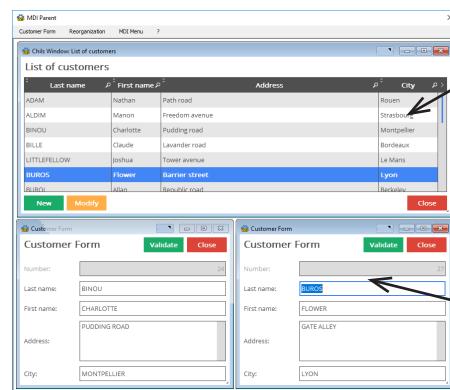
- as tool palettes,
- as dialog boxes whose modification directly affects the main window.



In this example, each description window of a polo shirt is used to open the description window of another polo shirt. All the opened windows remain accessible.

## MDI windows

In an MDI application, all the application windows are displayed in the same main window: the MDI parent window. Two types of windows can be found in the MDI applications:



**MDI parent window:** All the other application windows are displayed in this window. This window allows the user to handle the main application menu.

**MDI child window:** Application window displayed in the parent window. The user will be able to type information in these windows.

# Internal window

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

## 1. Creating an internal window.

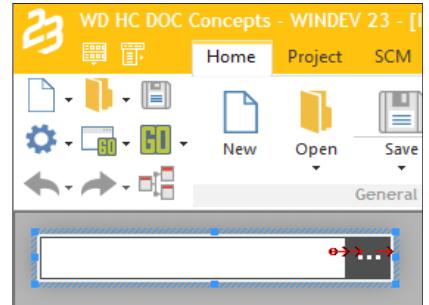
To create an internal window:

- Click among the quick access buttons.
- The window for creating a new element is displayed: click "Window" then "Internal window".

An internal window is a specific window that contains no title bar and no menu.

All types of controls can be used in this window.

In the example, an internal window is used to select files.



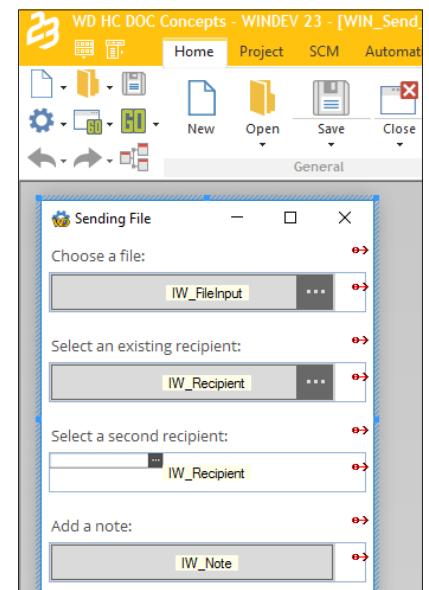
## 2. Using an internal window.

To use an internal window:

- Create an Internal Window control.
- In the description window of the control, select the internal window to use.
- Validate the description window of control.

### Notes:

- The internal window used in the "Internal window" control can be modified by programming.
- Limitations: The host area is rectangular and no overload is allowed. To perform overloads, we recommend that you use control templates.

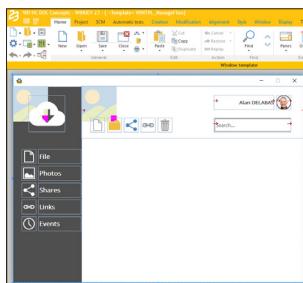


# Window templates

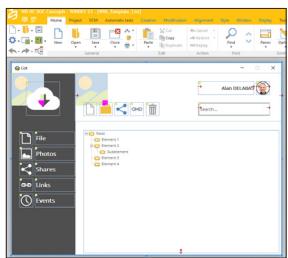
WINDEV allows you to create window templates. These templates contain all the graphic elements common to all the windows of your application.

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

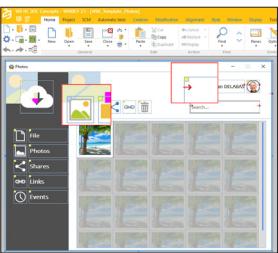
A window template allows you to comply with the style book defined for an application.



Defining a window template.  
The template is bordered by an orange line in the editor.



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



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

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

## The windows in practice

The purpose of a window is to display, present or type information. This information can come from data files found in an analysis, external files, queries, ...

WINDEV proposes several solutions for creating the project windows:

- Create a blank window with the wizard.
- Build all the application windows from the analysis description.
- Create a window from the analysis description (with or without its code).
- Create standard windows.
- Create windows based on a template, ...

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

### 1 Creating a window

WINDEV allows you to create several types of windows:

- Blank windows, without control.
- Windows created by RAD/RID.
- Internal windows. These windows can be used in the "Internal Window" controls.
- Windows based on a window template.
- Windows imported from a non-WINDEV application.

This paragraph presents the creation of blank windows, internal windows or windows based on a template. See the online help for more details.

#### 1.1 Creating blank windows

To create a blank window:

1. Click among the quick access buttons. The window for creating a new element is displayed: click "Window" then "Window". The wizard for window creation starts.
2. Choose the "Blank" type.
3. Specify the skin template of the window.
4. Validate the window creation. A blank window is created in the editor. The backup window is automatically displayed.
5. Specify the title and name of the window. The window corresponds to a ".WDW" file. By default,

See "The controls in practice", page 39 for more details.

This chapter presents the following topics:

- "Creating a window", page 21.
- "Simple operations on a window from the editor", page 22.
- "Main characteristics of a window", page 22.
- "Window and image", page 24.
- "Opening a "popup" window", page 26.
- "Dim disabled windows", page 27.
- "Internal window", page 27.
- "Window templates", page 28.
- "Processes associated with windows", page 28.
- "The menus", page 29.
- "The status bar", page 33.

this file will be created in the main project directory. This name will be used to handle the window.

6. Validate the window backup.
7. Create the controls in the window.

**Note:** The main characteristics of windows are presented in the online help. The different types of controls are presented in "The controls in practice", page 39.

#### 1.2 Internal window

An internal window is a window without title bar and without status bar that will be directly included in the other windows of your application via the "Internal window" control.

This allows you to include a window (and its code) in another window.

To create an internal window:

1. Click among the quick access buttons. The window for creating a new element is displayed: click "Window" then "Window". The wizard for window creation starts.
2. Select the "Internal window" tab.
3. Select the type of window and the associated skin template.
4. Validate. The backup window is automatically displayed.

### 1.3 Window based on a template

WINDEV also allows you to create a window based on a window template. In this case, all you have to do is select the requested window template.

The window templates contain all the graphic elements and the code common to all the windows found in your application.

#### To create a window based on a template:

1. Click  among the quick access buttons. The window for creating a new element is displayed:

click "Window" then "Window". The wizard for window creation starts.

2. Select the "Based on a template" tab.
3. Select the requested template and the associated skin template.
4. Validate. The backup window is automatically displayed.

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

A window template allows you to comply with the style book defined for an application.

## 2 Simple operations on a window from the editor

The window editor allows you to perform the following operations on the windows:

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

- Resize a window.
- Save and copy a window.
- Import and export a window.
- Modify the tab order.

See the online help for more details.

## 3 Main characteristics of a window

### 3.1 Characteristics

A window can include all types of controls. A window can:

- be **resizable**: the user can enlarge or reduce the window. Depending on the anchoring characteristics defined for each control, the position of controls in the window can change during this resize operation (see "Anchoring controls", page 45 for more details).
- include a **background image**: this image is displayed in window background and the controls are superimposed on this image. You also have the ability to define the characteristics of resize operation via the 9-image mode. See "Window and image", page 24 for more details.
- be **clipped**: the window will have custom shapes (window shaped like a circle, a computer, ...). See "Clipped window", page 26 for more details.
- include a **drop-down menu**: this menu allows the users to quickly access the main application features. To create a drop-down menu in a window, on the "Windows" pane, in the "Bars and menus" group, expand "Main menu" and select "Add the main menu". The first option is displayed in the window. The options can be handled via the popup menu of options. Then, the menu options can be handled by programming. See "The menus" (page 29) for more details.
- include a **status bar**: this status bar will display the help message associated with the different window controls. This status bar can also be used to display additional information. See "The status bar", page 33 for more details.
- be **multilingual**: all languages defined for the window will be available for all window controls. A window can be associated with more languages than the project (windows shared between several projects for example). See "Multilingual applications", page 137 for more details.
- be automatically **grayed** when it becomes inaccessible. This feature is called DDW, which stands for Dim Disabled Windows. See "Dim disabled windows", page 27 for more details.

- allow **Drag and Drop**. See the online help for more details.
- allow the **interface to be modified** by the end user. See the online help for more details.

**Note:** To force an action on a window, all you have to do is define a button of this window in "Automatic execution" mode. See "Automatic button execution", page 47 for more details.

### 3.2 Description window

This window includes several tabs that group all configurable window characteristics.

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

### 3.3 Displaying characteristics

To display the description window:

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

### 3.4 Characteristics by tab

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

See the online help for more details about the characteristics found in each tab.

#### General tab

The "General" tab is used to specify the window name as well as its title.

#### GUI tab

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

- visibility of window,
- position of window,
- management of right click and mouse cursor,
- association with a popup menu, ...

#### Details tab

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

- interface modification by the end user,
- animation,
- automatic closing.

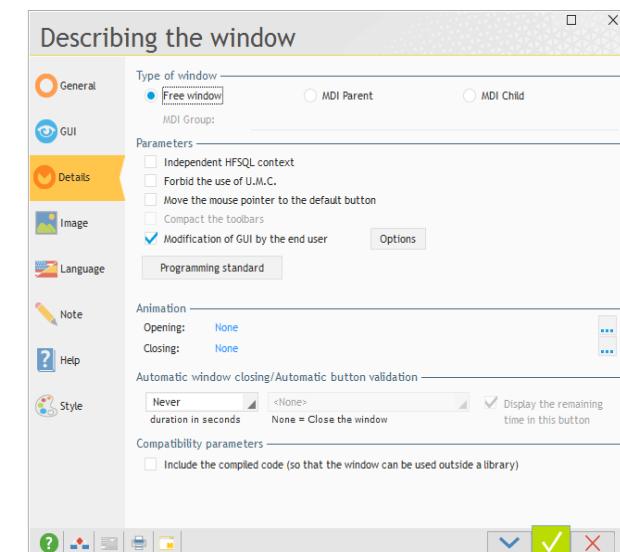
#### Image tab

This tab is used to configure:

- the background image of window,
- the image of sizing handle,
- the window icon.

#### Languages tab

This tab is used to find out the languages supported by the window as well as the language displayed in the editor. You have the ability to add, modify or delete languages.



**Notes tab**

The "Notes" tab is used to describe the operating mode of window. This information will be printed in the program documentation (project documentation, window documentation, ...).

**Help tab**

The "Help" tab is used to associate a context-sensitive file with the window.

## 4 Window and image

### 4.1 Overview

To optimize the style of your screens, WINDEV allows you to use:

- background colors. Several standard colors can be used. The background color is configured in the "Style" tab of the description window of the window.
- images in your windows. These images are used as background image of the window. Your controls are displayed above this image. The configuration of the background image is performed in the "Image" tab of the window.

#### Notes:

- If necessary, WINDEV also gives you the ability to use the XP/Vista style and theme in your windows. See the online help for more details.
- To customize your windows, you also have the ability to use a skin template or a window template.

### 4.2 Configuring the background image

The configuration is performed in the "Image" tab of window description.

You have the ability to select the file corresponding to the background image or a catalog image. The main image formats that can be displayed are: BMP, JPEG, TIFF or GIF.

You can also Drag and Drop the image file found in the explorer toward the WINDEV window.

This help file contains the context-sensitive help pages displayed by each control.

**Style tab**

The "Style" tab is used to define the window style:

- Presence of title bar.
- Presence of different icons.
- Presence of status bar, ...

The following options can be applied to the image:

- Management of multilingual images.
- Display mode.
- Position.
- Transparency.
- Symmetry.
- 9-image mode.
- Clipping the window according to the background image.

**Management of multilingual images**

A different background image can be displayed according to the runtime language of program. The display characteristics can be specific to each language.

**To define a multilingual background image:**

1. Click the icon found on the right of the name of background image.
2. Select the "Multilingual" option.
3. Choose the appropriate image for each language defined in your window.
4. Validate. The "Multilingual value" caption appears in the control for defining background images.

**Display mode**

An image can be displayed in a window according to several display modes. These display modes are used to produce different effects:

- **100%:** the image is displayed at 100% from the top left corner of Image control.
- **Centered:** the image is displayed at 100% and it is centered in the Image control.
- **Stretched:** the image is stretched to occupy the entire surface of Image control.

- **Tiled:** the image is displayed at 100% and it is tiled in the Image control to occupy the entire surface.

- **Homothetic:** the image is proportionally enlarged to be entirely displayed in the Image control.

- **Homothetic extended:** the image is proportionally enlarged in order for the smallest image side to be entirely displayed in the Image control.

- **Homothetic centered:** the image is homothetic and it is centered in the Image control.

- **Homothetic centered extended:** the image is extended and it is centered in the Image control.

- **Homothetic without enlargement:** if the image can be entirely displayed in the window, it is displayed at 100%. If the image is bigger than the window, it is automatically reduced proportionally in order to be entirely displayed.

- **Homothetic centered without enlargement:** the image is centered and homothetic without enlargement.

The "High Quality Display" option is used to improve the quality of the image in homothetic mode when this one is reduced in relation to its initial size.

**Position**

This parameter is used to define the position of top left corner of image in relation to the window surface.

These options will be stored even if the image is modified by programming (..Image).

**Transparency**

The transparency options for the background image are as follows:

- **Transparent magenta:** The magenta color found in the image will be transparent when the image is displayed in the editor or at run time.

Transparency: none

Transparent magenta



- **Top left pixel:** all pixels with the same color as the top left pixel of image will be transparent.

These options will be stored even if the image is modified by programming (..Image).

**Symmetry**

The symmetry options are used to change the orientation of image found in the image file. These options will be stored even if the image is modified by programming (..Image).

Initial image:



The options are as follows:

Vertical symmetry	Horizontal symmetry	Horizontal and vertical symmetry
-------------------	---------------------	----------------------------------

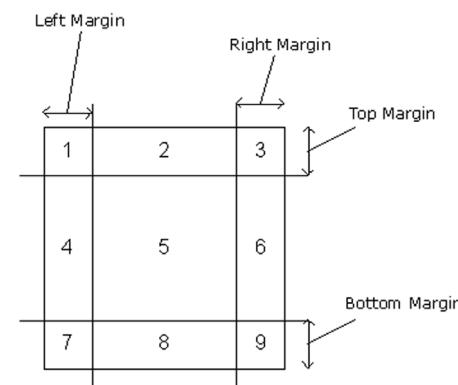
**9-image mode**

If this option is selected, the 9-image mode will be used on the background image of the window.

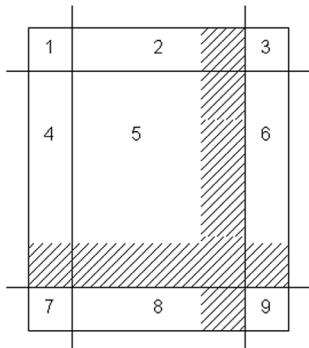
The 9-image mode consists in dividing the background image into 9 areas.

For each area, you can define whether the image section corresponding to the area must be displayed or not (all you have to do is check the name of the area in order to display it). If the image section is not displayed, it will be replaced by the background color of the window.

The margins are used to define the size of each area.



When resizing the window, only the areas 2, 4, 6, 8 and 5 are resized to occupy the entire space:



The 9-image mode is used to manage the operating mode of each area when resizing the window.

For each area:

- by repeating the image that constitutes the area,
- by stretching the image that constitutes the area.

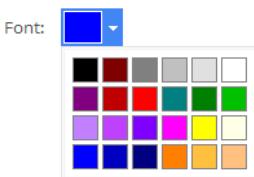
## 5 Opening a "popup" window

### 5.1 Overview

WINDEV gives you the ability to open "popup" windows.

A "popup" window corresponds to a small window that is opened above the current window. In most cases, this window allows the user to perform a selection. This window returns the value selected by the user to the main window. This window is automatically closed as soon as a left mouse click is performed (inside the popup window or not).

For example:



**Note:** The characteristics of a popup window are as follows:

- opening below the control that opens the popup window (or above the control or on the left of control if there is not enough space below the control). The opening position can be modified by programming.
- automatic closing as soon as the window loses focus or during the call to **Close**.
- only the closing icon is displayed in the title bar of window (if the title bar was defined in the style).

## 4.3 Clipped window

WINDEV allows you to clip windows. The windows will have custom shapes (window shaped like a circle, a computer, ...).

The window shape is defined by an image. In this image, define the color that will be used to define the transparent areas ("Light magenta" for example). All these areas will be invisible when displaying the window at run time: the window will be clipped.

**Caution:** The image borders must be as sharp as possible (no gradient, ...) to avoid any unexpected display effects when previewing or opening the clipped window.

The window clipping is configured in the "Image" tab of the description window of window.

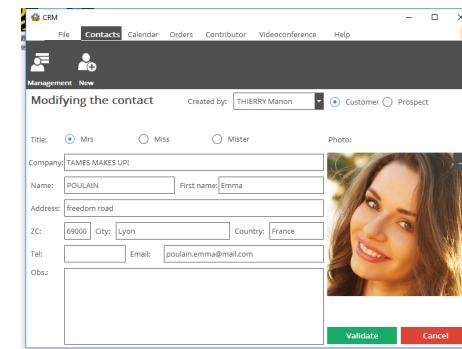
## 6 Dim disabled windows

Your application opens several windows at the same time and you don't know which one to use?

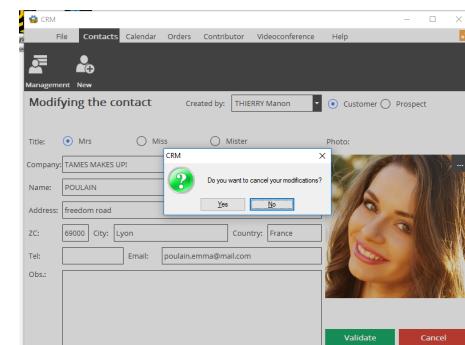
The inactive windows can be automatically dimmed. This feature allows you to always know which window to use.

This feature is called DDW, which stands for Dim Disabled Windows.

For example:



Step 1: You modify the information regarding a customer then you click the "Validate" button.



Step 2: A window asks you to confirm or cancel these modifications. The main window (in the background) has become unavailable. It is automatically dimmed. You immediately know which window must be used.

The DDW features are available for the entire project or they can be customized for each window. You can also act on the DDW feature by programming.

See the online help for more details.

## 7 Internal window

The internal windows are used to dynamically share the same section of interface in one or more applications.

The interface that will be used several times is created in an "Internal Window" window.

This interface is used in the different windows of your application via the "Internal window" control.

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

See the online help for more details.

## 8 Window templates

WINDEV allows you to create window templates. These templates contain the graphic elements and the code common to all the windows found in your application.

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

A window template allows you to comply with the style book defined for an application.

Using window templates in your applications allows you to:

- simplify the creation of application windows.

- simplify the layout of application windows.
- simplify the update of style book defined for the application.

A window template is as easy to create as a window. To create a window from a template, all you have to do is choose "Based on a template".

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

See the online help for more details.

## 9 Processes associated with windows

### 9.1 Processes managed by default

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

#### • Global declarations:

Declaring the variables global to the window

#### • End of initialization:

Run when opening the window, before the code for taking focus. The initialization process of the window controls is run after this code.

#### • Closing:

Run when closing the window. Note: resuming the input in one of the window controls in this code prevents from closing the window.

#### • Request for refreshing the display:

Used to group the code for updating the window controls: content of controls, status (visible, grayed, ...), static to update, ...

This process is started upon request by the following functions:

- **RequestUpdateUI:** Runs the process at the end of current process (asynchronous execution).

- **ExecuteUpdateUI:** Runs the process immediately.

- Assigning the ..Value property (internal window only)

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

- Retrieving the ..Value property (internal window only)

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

### 9.2 Optional processes

Several optional processes can be managed. To do so, you must:

1. Click "..." found in the "Other processes" process of code editor.

2. The window of optional processes is displayed.

3. Select the optional processes to add and validate.

4. The selected processes are automatically added after the processes proposed by default.

You have the ability to manage:

- the gain and loss of focus,
- the modification of one of window controls,
- the modification of window size,
- the window rollover by the mouse,
- the left mouse button down, up, with double click,
- the right mouse button down, up, with double click,
- the mouse wheel, ...

## 10 The popup menus

A popup menu can be opened by the user via a right mouse click. This menu allows the users to easily access the specific features of a window or window control.

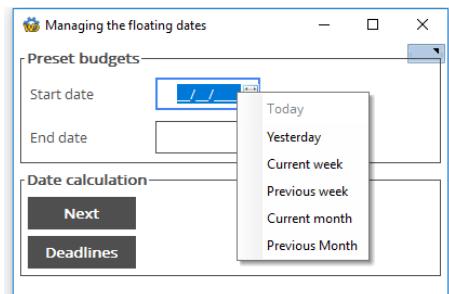
WINDEV allows you to manage two types of popup menus:

- **the system popup menus** managed by Windows. These popup menus are displayed during a right click on some types of controls (edit control, table, ...). See "Setup media", page 168 for more details.
- **the custom popup menus** managed by WINDEV. These menus are created in the window editor and they can be handled by programming.

WINDEV allows you to create custom popup menus and to associate these popup menus:

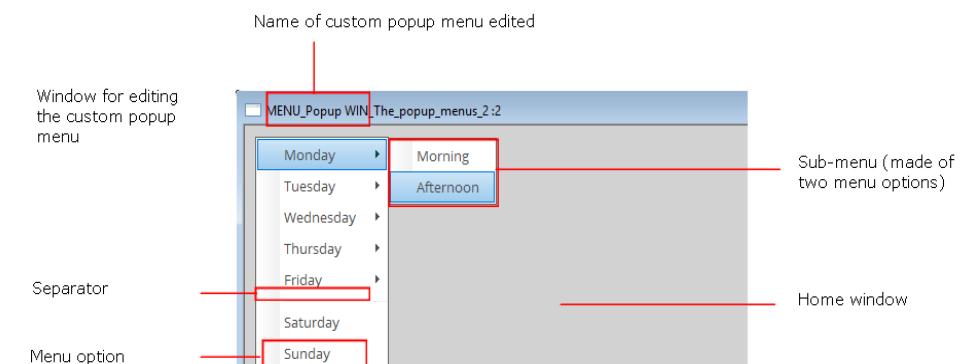
- with one or more window controls.
- with a window.

For example:



### 10.1 Window for editing popup menus

WINDEV allows you to create and modify custom popup menus in a specific edit window. To better understand the management of the custom popup menus, let's take a look at the vocabulary linked to this edit window:



## 10.2 Available features

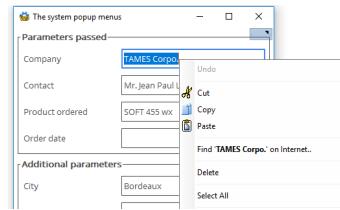
- WINDEV allows you to:
- Create a custom popup menu.
  - Associate a custom popup menu with one or more elements (window or controls).
  - Edit a custom popup menu.
  - Display the description window of a custom popup menu.
  - Delete a custom popup menu.

See the online help for more details.

## 10.3 Default system popup menu

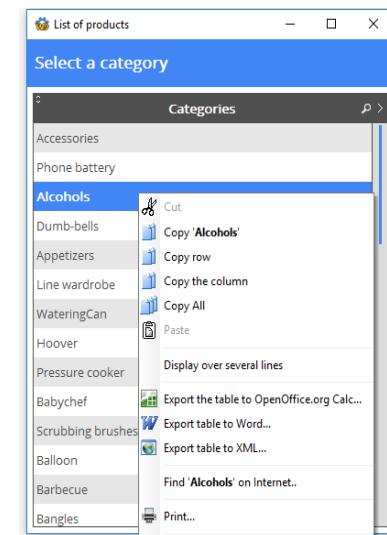
The following types of controls have a default system popup menu (displayed if no custom popup menu is associated with the control):

- an edit control:



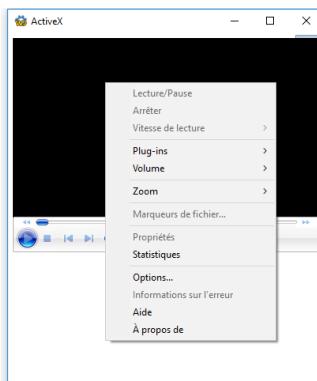
See the online help for more details.

- a table column:



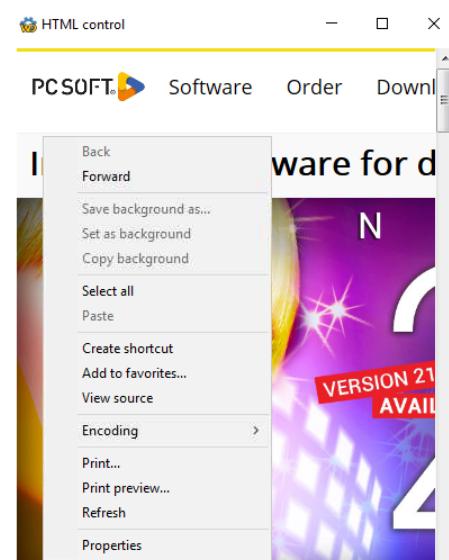
See the online help for more details.

- an ActiveX control:



**Note:** The system popup menu displayed depends on the ActiveX associated with the control

- an HTML control:



**Note:** The system popup menu displayed is the one of Internet Explorer. It depends on the object that is selected in the page (image, text, link, ...).

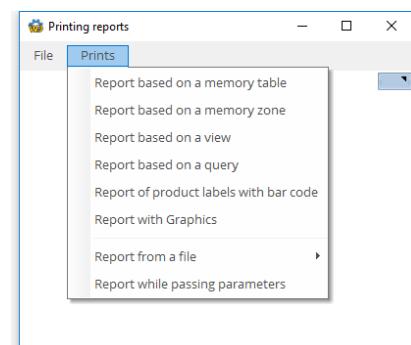
This default popup menu can be disabled from the editor or by programming.

## 11 The main menus

### 11.1 Overview

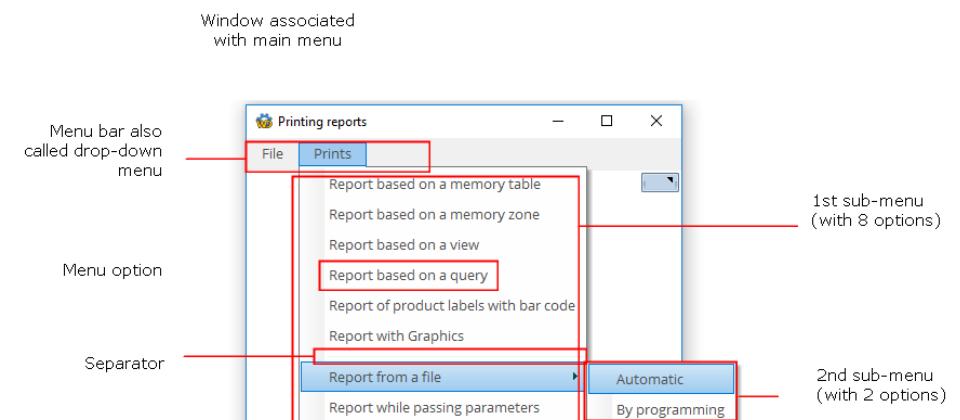
A main menu is necessarily associated with a window. This menu is displayed below the title bar of the window in WINDEV. In most cases, the applications contain a menu window. This window contains a main menu and it is used to access the other application windows.

For example:



### 11.2 Vocabulary associated with the main menus

To easily handle a main menu, you must be familiar with the following vocabulary:



### 11.3 Available features

WINDEV allows you to:

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

See the online help for more details.

## 12 The menu options

### 12.1 Overview

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

In the editor, the operations performed on the menu options are the same for a main menu and for a custom popup menu.

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 window:
  - open the relevant window in WINDEV.
  - select one of the menu options. The options and the sub-options are automatically displayed.
- a custom popup menu of a window: on the "Window" pane, in the "Bars and menus" group, expand "Popup menus" and select "<NamePopupMenu>".

### 12.2 Handling the menu options in the editor

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

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

### Special case: About menu

WINDEV allows you to automatically insert a "tool" menu ("?" type) in your application. On the "Window" pane, in the "Bars and menus" group, expand "Main menu" and select "Add the '?' menu".

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

The description window of menu options allows you to:

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

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

See the online help for more details.

### 12.3 Handling the menu options by programming

The method for handling a menu option is identical for the popup menus and for the main menus.

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

- "<OptionName>" if the menu option is handled from the window.
- "<WindowName>,<MenuName>,<OptionName>" if the option is handled from another window.

## 13 The status bar

In the status bar of a window, you have the ability to display:

- an **information message related to the current control**. This message is entered in the "Help" tab of the description window of the control. The message will be displayed in the status bar when the control is selected by the user.
- a **specific information message**. This message is displayed by programming with **Message**.
- a **progress bar**. This progress bar is displayed by programming with **Gauge**.

To display various information at the same time, you have the ability to divide the status bar into several boxes ("Insert a cell" from the popup menu of the status bar).

### 13.1 Description window of status bar

The description window of a status bar is used to:

- add and/or delete cells.
- modify the order of cells.
- specify the information that will be displayed for each cell.

See the online help for more details.

### 13.2 Type of cell in the status bar

The type of cell is used to display:

- The progress bar. This progress bar will be run when using **Gauge**.
- The help message associated with the control with focus. This help message is defined in the "Help" tab of control or by programming with **..Message**.
- The date of current computer.

- The time of current computer.
- The cursor position in the current edit control (Column and Row of Caret).
- The status of [Caps Lock] key indicating whether the Caps Lock key is enabled or not.
- The status of [INS] key indicating whether the [Insert] key is enabled or not.
- The name of current user in an application that is using the user groupware.

Two specific options are also available:

- By programming: The information displayed in the status bar is specified by programming (**Message** for example).
- Update process: The "Whenever updated" process is automatically associated with the cell of status bar. This process is run on a regular basis. This process can contain the elements that will be displayed in the cell for example.

### 13.3 Characteristics of a cell in the status bar

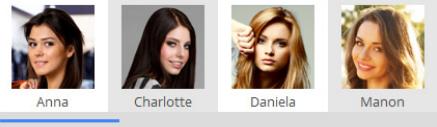
The description window of status bar also allows you to define the display characteristics of cells found in the status bar.

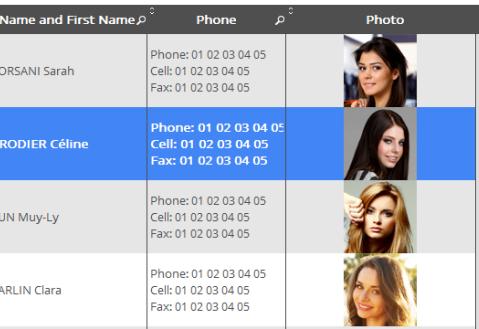
- The cell position in the status bar.
- The background image of cell.
- The background color of cell.
- The cell width.
- Whether the cell is editable.
- The cell anchor (its operating mode when resizing the window).

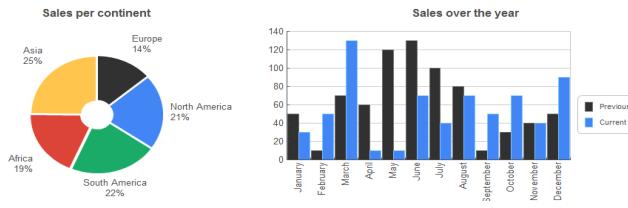
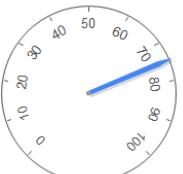
All these characteristics can be handled by programming via the properties available for the cells in the status bars.

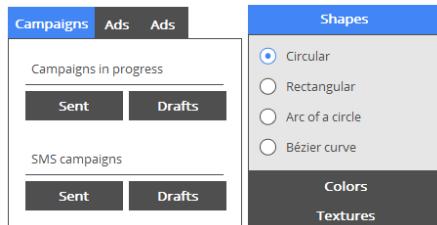
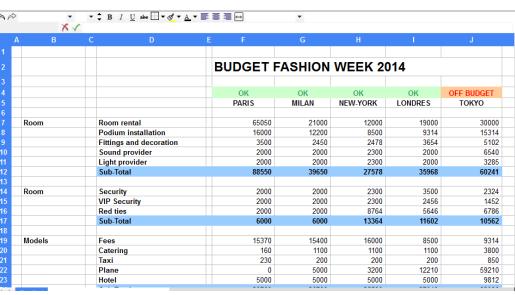
# The different types of standard controls

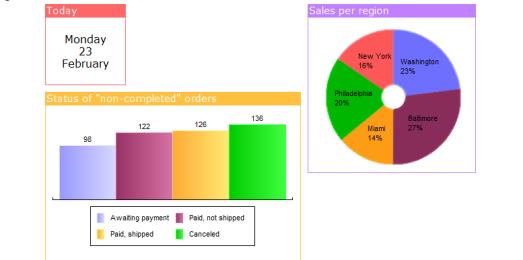
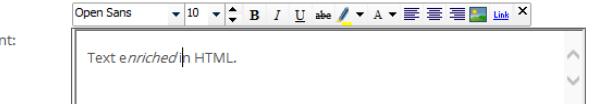
WINDEV includes several controls supplied with the product. Let's see how to use them according to the operation to perform:

You want to...	Use a control...
Display a text, a title, ...  A simple static.  An <u>RTF</u> static with enrichment.	Static
Read a PDF file	PDF reader
Display and modify a Word document.	Word processing
Select a value from a list (Country, City, ...)  Title: Mr <input checked="" type="radio"/> Mr <input type="radio"/> Mrs <input type="radio"/> Miss	Radio button, Combo box, List box
Select several values from a list (recipients of a message, ...)  Bodywork: Berline <b>Coupé</b> Monospace Décapotable	Check box, List box
Select one or more values from a listview (picture directory, ...)  	ListView
Select a value in a preset range  Volume: 	Slider, Round slider, Spin, Range Slider

Select a date		Calendar
Display a graphic image (photo, ...), an animation or use the drawing functions		Image
Display the file content in a table (list of customers, order details, ...).		Table
Repeat controls in a window (product catalog with photo, ...)		Looper

Display a video, an animation		Web camera Multimedia Conference
Type information	Email: <input type="text" value="mike.durand@mail.com"/>	Edit control
Program an action in a window (display another window, start a print, ...)		Button
Display information via a hierarchical system (directory)		TreeView TreeView table
Display a column chart, a line chart, a pie chart, ...		Chart
Display a progress		Progress bar

Group the controls by theme and display the themes one by one		Tab Sidebar HideShow
Display a bar code		Bar code
Display events over a period of time		Organizer
Create statistical tables		Pivot table Spreadsheet

Represent weighted data		TreeMap
Display the important elements in dashboard format		Dashboard
Type or display HTML data		HTML edit control HTML control
Allow the selection of interval or zoom		Range slider
Display a map, an itinerary		Map
Edit an image		Image editor
Display an ActiveX control		ActiveX

## The controls in practice

WINDEV proposes several types of controls that can easily be included in your windows. These controls can be created via the "Creation" pane of WINDEV ribbon.

All the controls can be handled by programming.

### 1 Creating a control

#### 1.1 The different types of window controls

The following controls are available in the window editor of WINDEV:

- Edit control,
- PDF reader,
- Static,
- Gantt chart,
- Button,
- TreeView table,
- TreeView,
- Image,
- Organization chart,
- Radio button,
- Progress bar,
- Check box,
- Looper,
- List box,
- Multiline zone,
- Combo box,
- Scrollbar,
- ListView,
- Sidebar,
- HideShow,
- Web camera,
- Conference control,
- OLE control,
- ActiveX,
- Shape,
- Bar code,
- HTML control,
- Supercontrol,
- Organizer,
- Scheduler,
- Rating,
- Pivot table,
- Map,
- Carousel,
- Cube,
- Widget,
- Xaml,
- Calendar,
- Dashboard,
- Internal window,
- Control template,
- Layout control.

All the controls can be handled by programming.

See the online help for more details.

#### 1.2 Creating a new control

To create a control:

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

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

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

All the window controls can be associated with a file item EXCEPT FOR:

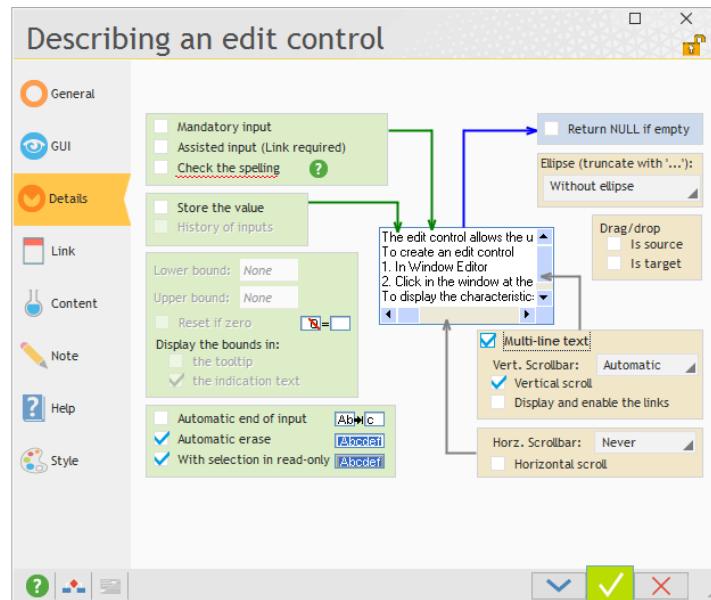
- |            |                  |
|------------|------------------|
| • Button   | • TimeLine       |
| • Shape    | • Dashboard      |
| • Tab      | • Xaml           |
| • Toolbar  | • Carousel       |
| • Splitter | • Cube           |
| • Chart    | • HTML control   |
| • Map      | • Layout control |

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 the control.

## 2 Characteristics of a control

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



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

### 2.1 Displaying characteristics

To display the description window of control:

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

#### Notes:

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

### 2.2 Characteristics by tab

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

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

#### General tab

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

#### GUI tab

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

- Initial control status when opening the window
- Visibility of control
- Size of control
- Alignment, ...

#### 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

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

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

#### Content tab

This tab is available for the window controls only.

The "Content" tab is used to define:

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

#### Notes tab

The "Notes" tab is used to describe the operating mode of the control. This information will be printed in the program documentation (project documentation, window documentation, ...).

#### Help tab

The "Help" tab is used to enter and configure the types of help systems 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.
- A help window with a specific title, displayed when pressing the F1 key.

A control can also be associated with a specific help page, found in a help file in CHM format. This help page will be displayed:

- in the format of a context-sensitive help (tooltip displayed upon request), if a help file was defined for the window.
- in the format of a help page. **WHelp** will have to be used to display the help page.

#### Style tab

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

- Modify the aspect of control: all you have to do is select the control element to modify, then its style characteristics. Only the aspect of current control is modified.
- Apply an existing style to the control: all you have to do is select the style to apply.
- Create or modify a style: the created or modified style is updated for all project controls that use this 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 control,
- modify the size of control.

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

- The **black handles** are used to view and resize a control.
- The **white handles** are used to view a control. This type of handles is also displayed when a window is in read-only mode for the developer: the controls cannot be modified.
- The **gray handles** indicate the first selected control during a multiple selection. This control will be the reference control.

### 3.2 Available operations

The window editor allows you to:

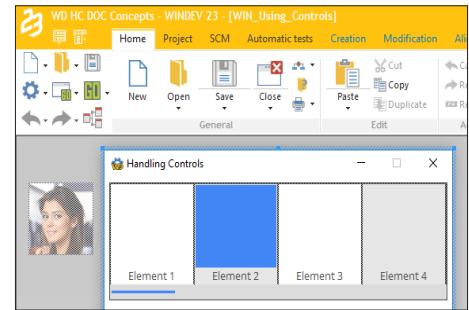
- Select one or more controls (with the lasso for example)
- Group the selected controls
- Move a control

## 4 Aligning controls

The alignment of controls allows you to create "professional" and outstanding interfaces. To help you build standard interfaces, WINDEV proposes the following tools:

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

- Modify the control caption
- Display an advanced tooltip when a control is hovered by the mouse cursor, containing: the control name, its position, its size, its anchoring, its initial status.
- Use an off-window control. This control will be visible in edit, it will not be visible at run time.



Note: An "off-window" control allows you to:

- keep a control in a window without displaying it at run time. This allows you to use the control content in programming (like a variable).
- display new controls in a window when resizing this window.

### 4.1 The rulers

In the window editor, you have the ability to display alignment rulers.

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 window.

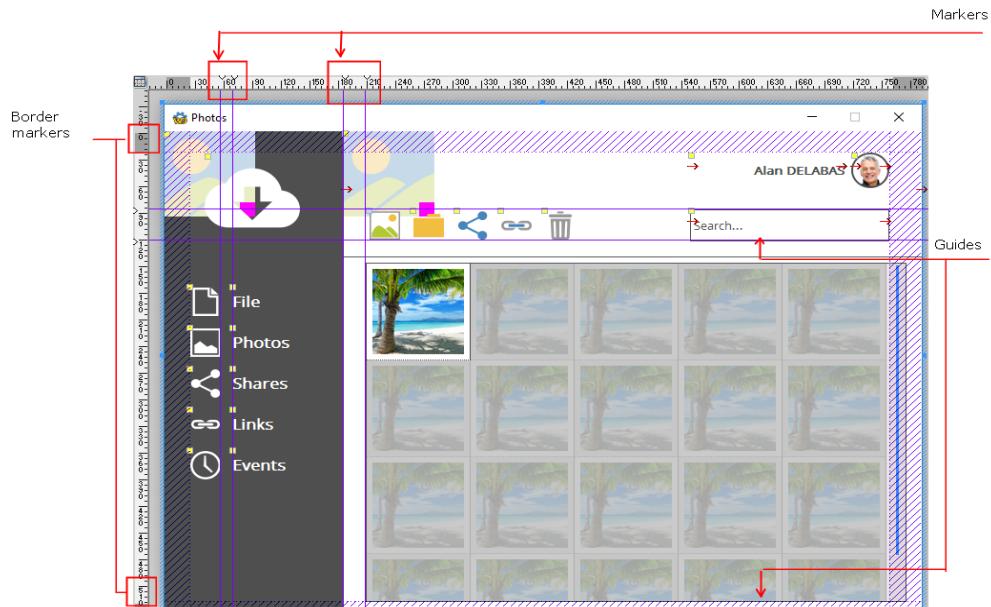
### Displaying the rulers

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 controls in the windows.
- the border guides, used to define a border of same size on each window side.

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

For example, for a window:



### Handling markers

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

### 4.2 Snap-on grid

The snap-on grid is used to position vertical and horizontal markers in the window. 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 window, 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 window.

### 4.4 Advanced interface checker

The advanced interface checker is a tool used to harmonize the positioning of controls in the different application windows.

The advanced interface checker proposes to apply, to the current window or to the selected controls, the presentation rules issued from the Windows standard: alignment of controls, normalization of button size, ...

### 4.5 The alignment options

To align several controls, WINDEV 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).

## 5 Edit options of controls in a window

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.
- the position of window when it is opened.

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.

### 5.1 Making a control visible/invisible

To make a control visible/invisible in the window editor:

1. Select the control in the editor.
2. On the "Modification" pane, in the "Help for edit" group, check/uncheck "Visible (editor)".

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 one with the white handles).
  - the last selected control.
  - the largest selected control.
  - the control found in the most top left position of selected controls.

### 5.2 Locking/Unlocking a control

To lock/unlock a control in the window editor:

1. Select the control in the editor.
2. On the "Modification" pane, check/uncheck "Locked".

A locked control is a control that cannot be selected with the lasso. For example, you may want to lock a tab. When several controls displayed in a tab are selected with the lasso, the tab is not automatically selected.

All the other operations can be performed on a locked control.

### 5.3 Framing (or not) a control

To frame (or not) a control in the window editor:

1. Select the control in the editor.
2. On the "Modification" pane, check/uncheck "Framed".

Framing a control allows you to easily view all the controls found in a window (especially for the image controls).

## 6 Anchoring controls

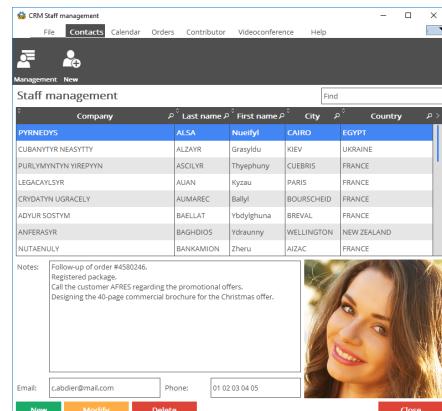
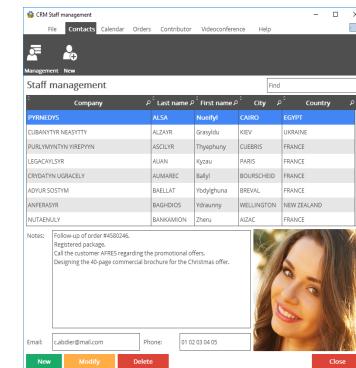
The windows of a WINDEV application can be resized at run time. The anchoring mechanism is used to automatically adapt the size and position of controls when resizing the window.

The management of control anchoring includes two parameters:

- **the anchor of control:** used to define the modification that will be applied to the control according to the change of window size. The control can move to left or to bottom, it can be enlarged in width and/or in height, ...
- **the anchor rate:** used to manage the percentage by which the control is moved or stretched. This parameter allows you to:
  - keep a control centered in width in the window regardless of its size.
  - proportionally enlarge some window controls.

These parameters can be configured:

- in the window editor.
- by programming with the WLanguage properties.



## 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 description window of control ("Description" from the popup menu of control).
  - via "Anchor" from the popup menu (right mouse click) of the control.
  - 3. In this window, choose the type of anchoring that will be used. The most common options are represented by icons:

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

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

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

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

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

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

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



**Vertical Distribution:** This option is available when several controls are selected. The anchoring options are used to keep a vertical distribution of selected controls when the window is enlarged in height.



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



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



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



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

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



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



**Horizontal Distribution:** This option is available when several controls are selected. The anchoring options are used to keep an horizontal distribution of selected controls when the window is enlarged.



**Vertically centered to right:** The control remains centered in height in the window regardless of the window height.

However, the control is anchored to right and it moves to right when the window is enlarged.

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



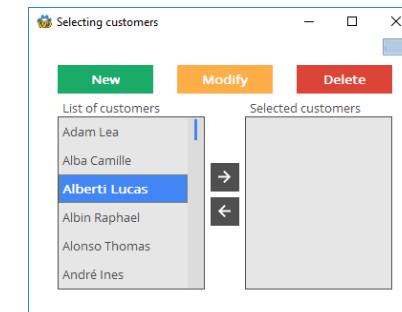
**Centered:** The control remains centered in height and in width in the window regardless of the window size.

4. Define (if necessary) the different anchor rates.
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).

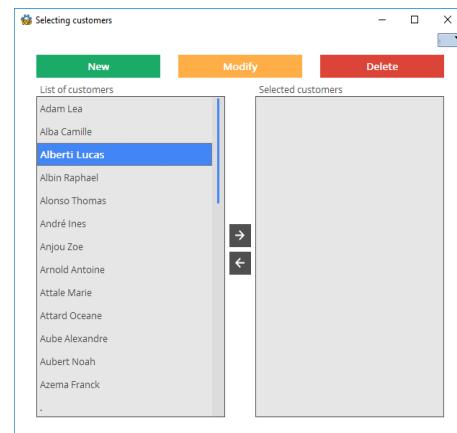
## 6.2 Example for configuring the anchoring of controls

Let's see a detailed example that illustrates the possibilities of anchoring controls in the windows.

Initial window:



Enlarged window:



To obtain this result:

1. Select the [<] and [>] buttons: these controls must be centered in height and in width (.
2. Select the two lists: these controls must be anchored to right, in width and in height ("to the right", "in width" and "in height" options checked).
3. Select the 3 buttons: these controls must stretch proportionally to right (.

Note: Several WLanguage properties allow you to configure the anchor of controls by programming.

## 7 Automatic button execution

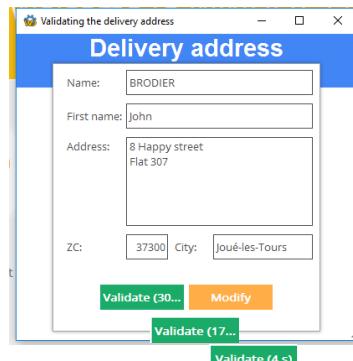
To force an action on a window, all you have to do is define a button of this window in "Automatic execution" mode. The principle is straightforward, if no window button was used after a specific duration, the process of the "Automatic execution" button is automatically run. At run time, the duration is automatically decremented. When it reaches zero, the code of the button is automatically run.

### 7.1 Implementation

To automatically run a window button:

1. Define the window button that will be run automatically.
2. In the window description ("Details" tab of description window), specify:
  - the time before the automatically execution of button.
  - the button to run.

3. At run time, the remaining time is automatically displayed.



## 7.2 Programming

You also have the ability to define an automatic validation button with *DelayBeforeClosing*.

## 7.3 AAF: Automatic Application Features

This feature is also directly accessible to the end user. It can ask for an application button to be automatically run after a specific duration.

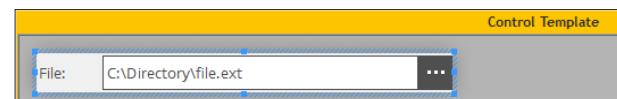
# Control templates

WINDEV gives you the ability to create control templates. A control template is a set of controls that can be re-used in several windows.

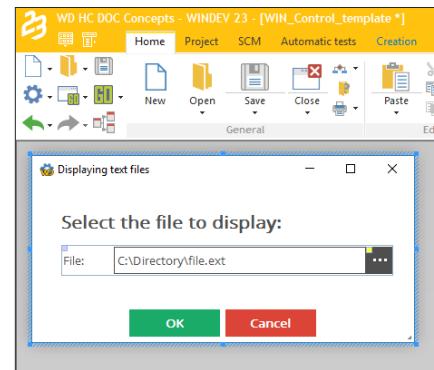
The modifications performed in a control template are automatically applied to all the windows that use this template.

A control template is used to:

- group a set of controls for a specific purpose.
- make the controls independent of the window that is hosting them.



Defining a control template.  
The template is enclosed by a green line in the editor.



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

To create a window based on a control template, 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.

# Edit: 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, ...

A report can be displayed on the screen, saved in an HTML file, printed, ...

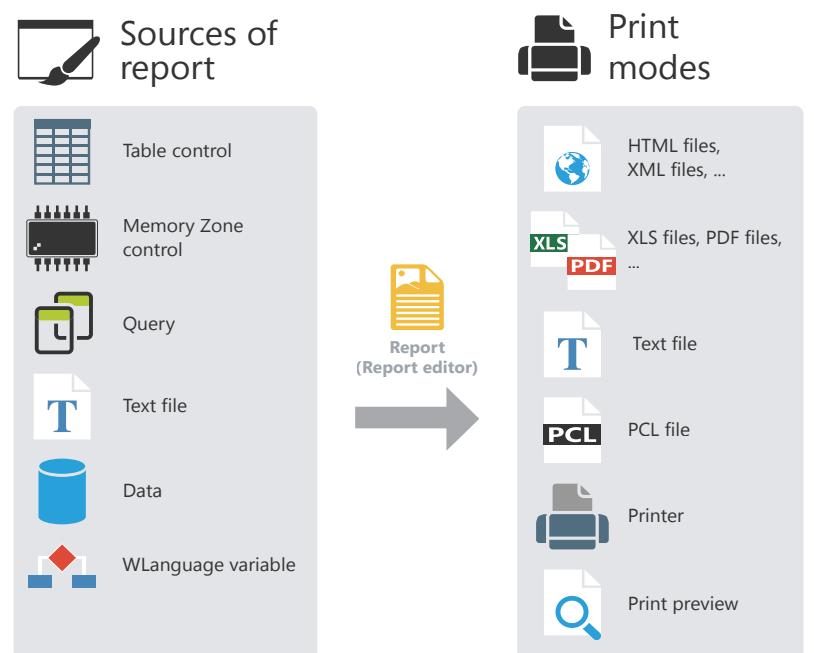
A report can be used to summarize and synthesize data.

You can:

- group the data.
- sort the data according to any criterion.
- perform calculations (means, statistics) or create charts.

The diagram below presents a simplified definition of a report:

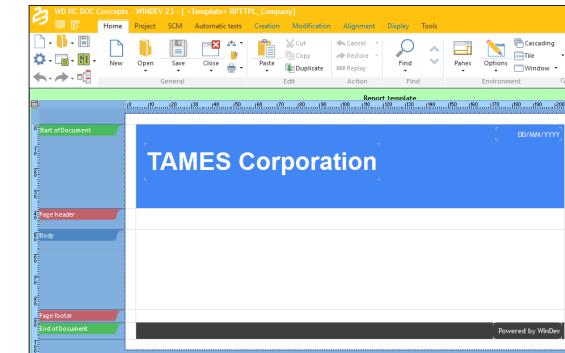
- the data to print comes from a data source (file described in an analysis, HFSQL view, query, memory zone or text file).
- the report groups, sorts and formats the data.
- the report execution can be directed to the screen, in an HTML file, in a RTF file, ... or to a printer.



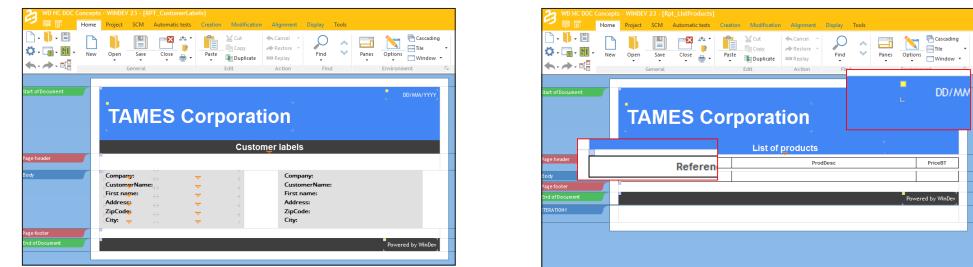
# Report templates

Most of the time, the prints use a standardized appearance and layout: date in the top right corner in a specific format, page footer with print time and file name, logo in the top left corner, ...

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.

**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.

DEVELOP 10 TIMES FASTER

PART 2

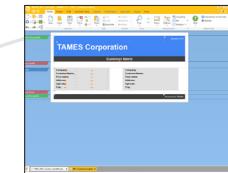
Development  
environment

# The WINDEV editors

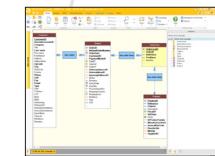
To handle a project, WINDEV proposes several editors:

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

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

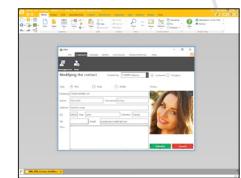


Report editor

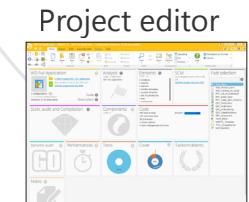


Data model editor

P



Window editor



Project editor



Query editor

# User-friendly editors in practice

## 1 Introduction

WINDEV is built around a set of editors operating in a single environment and adapted to the developer requirements:

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

- code editor, debugger and compiler.
- help editor.
- modeling editor.
- document editor.
- image editor.

These editors allow you to create the different elements (windows, reports, databases, programs, ...) used by the application.

## 2 The different editors

### 2.1 Project editor

The project editor includes two main features:

- viewing all the project elements (windows, reports and queries).
- viewing and creating the sequences between windows and reports.

Let's see some characteristics of the project editor:

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

### 2.2 Data model editor

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

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

The data model editor allows two types of database descriptions:

- direct description of the analysis linked to the

project (also called Logical Data Model (LDM))

- description of the Conceptual Data Model (CDM) then automatic generation of the analysis.

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

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

### 2.3 UML editor

The UML language is a graphical language allowing you to:

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

The UML editor is mainly used to:

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

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

Let's see some characteristics of UML editor:

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

### 2.4 Query editor

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

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

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

Let's see some characteristics of 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 Window editor

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

Several skin templates (style sheet associated with the window) and types of windows are proposed. They help you make your windows more ergonomic and make your applications more user friendly.

See "The windows in practice", page 21 for more details.

Let's see some characteristics of window editor:

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

### 2.6 Report editor

The report editor is used to easily create printed reports.

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

Let's see some characteristics of report editor:

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

## 2.7 Code editor

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

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

Let's see some characteristics of code editor:

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

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

## 2.8 Help editor

The help editor is used to create a set of help pages ("CHM" file) that documents your project.

A help page is created for each window found in your project. These pages are used to describe the different operations that can be performed by the users.

Let's see some characteristics of help editor:

- WYSIWYG editor ("What You See Is What You Get"). Ability to directly see the help pages of your project.
- Compilation of help pages and CHM creation.
- Creation of index and summary.
- Inserting images.
- Inserting links between the help pages.
- Ability to display the page titles in a bar.
- Zoom in the help editor.

## 2.9 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:

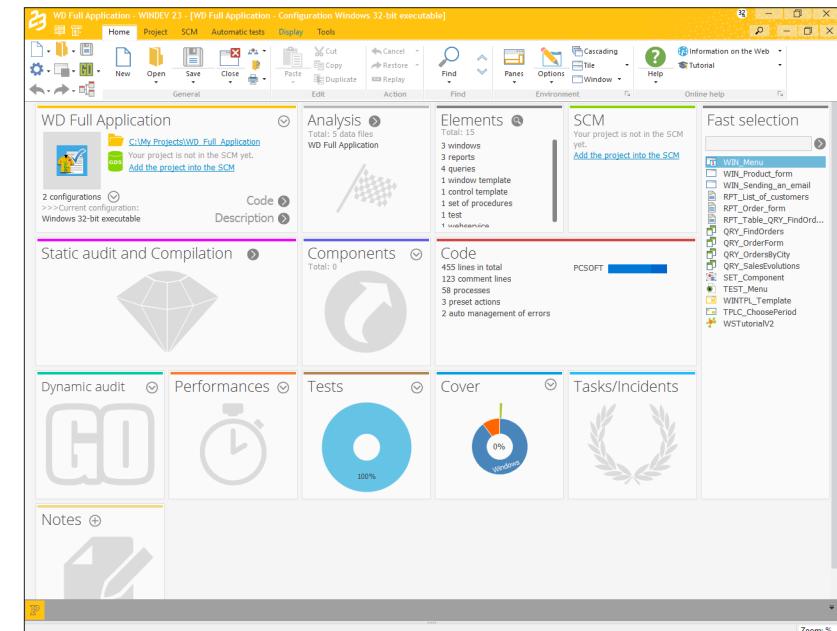
- Centering.
- Resizing.
- Symmetry, rotation.
- Management of texts.
- Management of layers.
- Access to images of image catalog.
- Preserving transparency.

# Project dashboard

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

The project dashboard includes several indicators about the project content:

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



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

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

# WINDEV/WEBDEV/WINDEV Mobile: 100% compatible

The created projects are often multi-target projects.

For example, for an ERP intended to operate in Windows, in addition to the application that will be the base of the applicative solution, the salesmen will be equipped with iOS or Android mobile devices, the storekeepers will use mobile devices for the management of stocks while Intranet/Internet sites will be implemented.

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

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

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



When a project is opened in a product other than the one that was used to create it, a wizard allows 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 WINDEV displays the thumbnails of the WEBDEV pages and the WINDEV Mobile windows for example. Clicking a WEBDEV page from the WINDEV project editor opens the WEBDEV page (WEBDEV must be installed on the computer).

**Note:** If you own WINDEV and WINDEV Mobile, you have the ability to handle the WINDEV Mobile configurations in WINDEV directly. You can open the mobile windows, modify them, run their test, ...

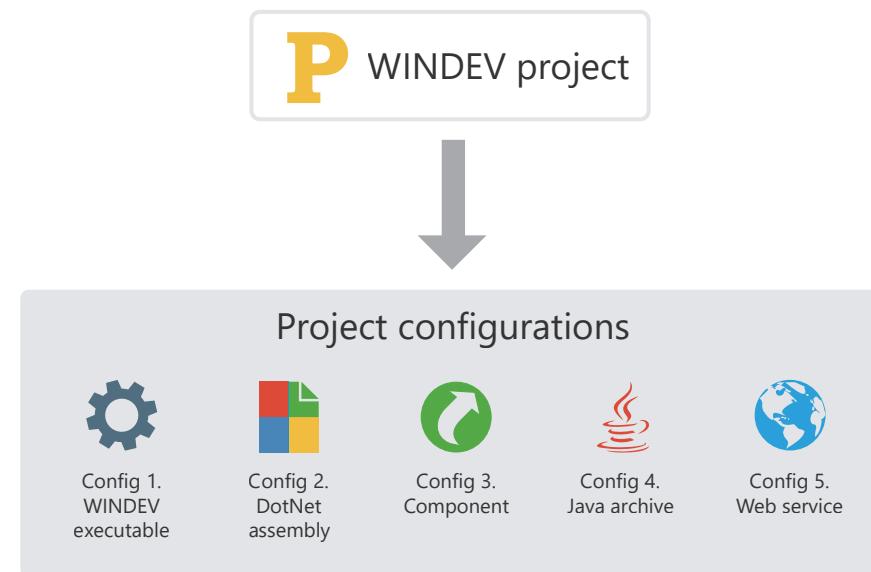
# Project configuration

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

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

- executables that do not contain the same elements, with different names, ...
- different components
- Webservices
- .Net assemblies

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



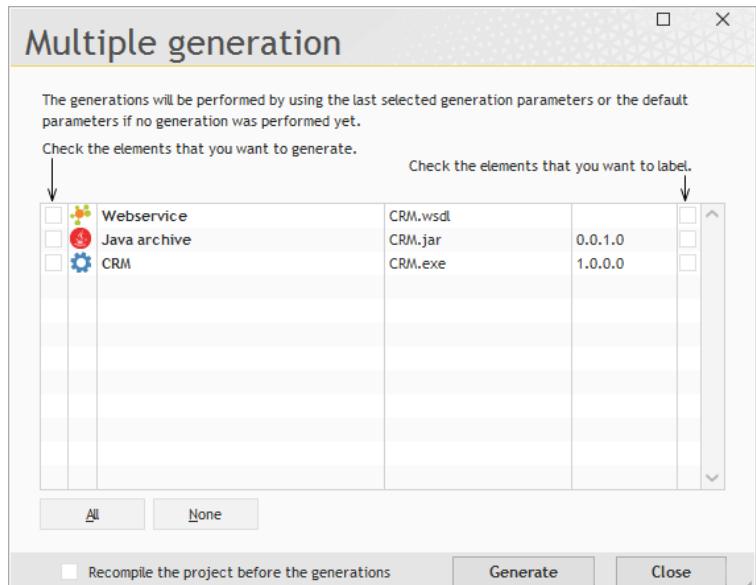
The multiple generation allows you to generate all project configurations (or some of them) in a single operation.

# Multiple generation

The project configurations are used to easily define the different "targets" of your project. For the same project, you have the ability to define the executables, libraries, Webservices, .Net assemblies, ...

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

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, click "Multiple generation".

# Custom-folders: Organize your project

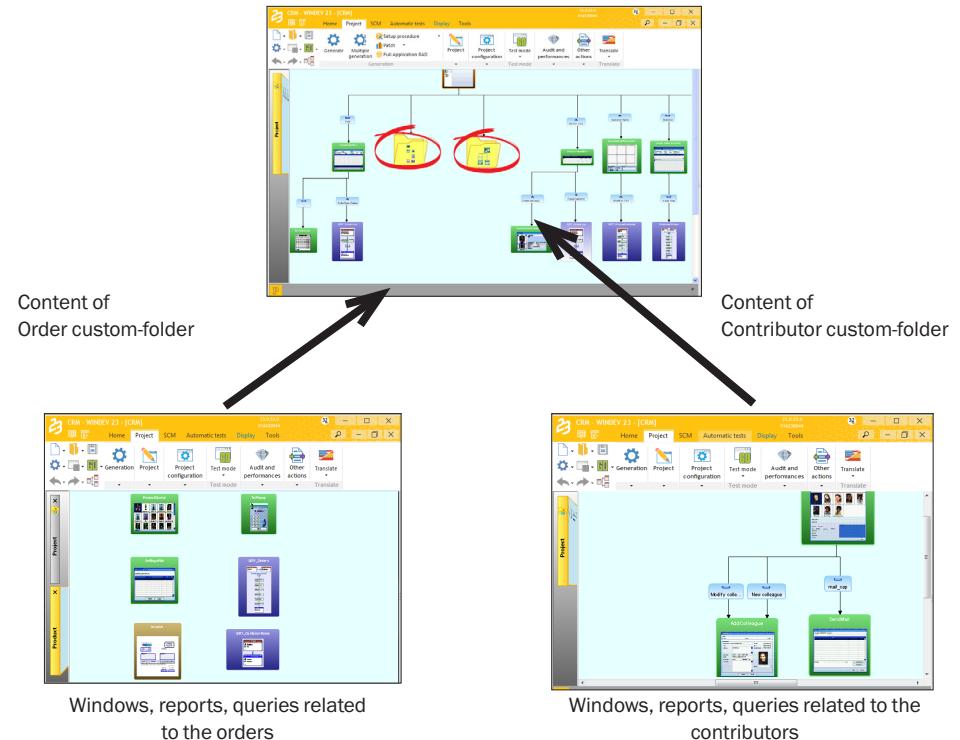
All elements included in a project are listed in the "Project explorer" pane. By default, the elements are sorted according to their type: windows, reports, classes...

In large projects, it is often more relevant to group the elements related to the same feature: stock management or order management for example.

To do so, create the "custom-folders" in the tree structure of project explorer and drag the different elements into these folders.

Some elements can be common to several "custom-folders".

It makes it easier to work on part of the application.



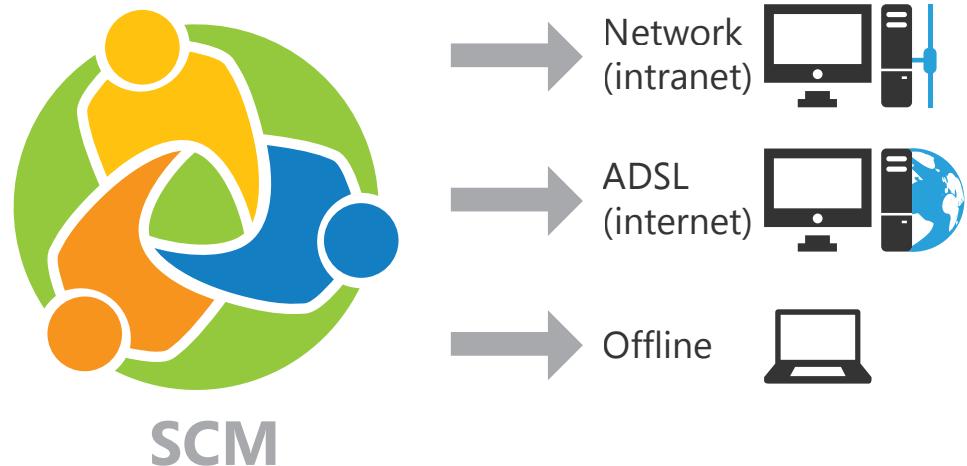
## Notes:

- To organize your projects while sharing a set of elements between different projects, WINDEV also proposes the internal components.
- The "custom-folders" are displayed in the project graph and in the "Project explorer" pane.

# Source Code Manager (SCM)

## Overview

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

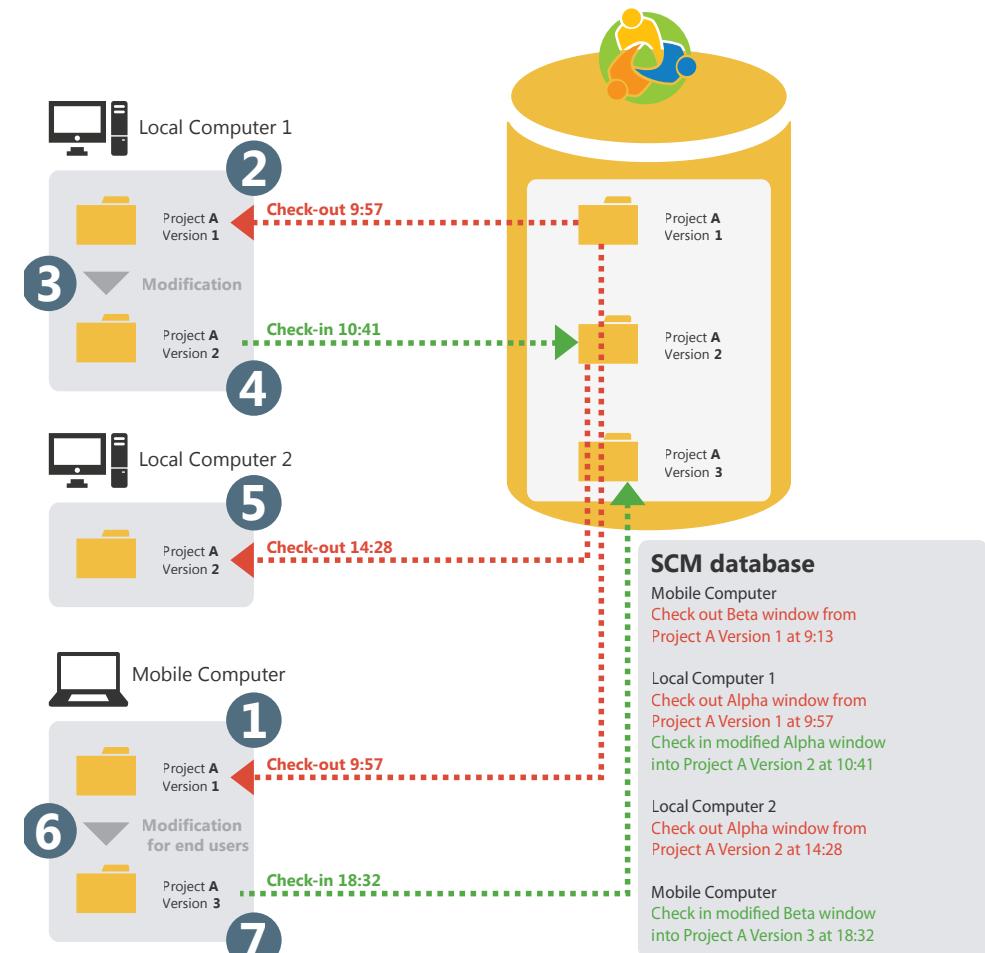


The elements found in the SCM can be shared:

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

## Operating mode of Source Code Manager

The following example presents the Source Code Manager:



If a project element (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 SCM database stores a history of all the project elements since their creation.

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

# Sharing a project in practice

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

## 1 The Source Code Manager

### 1.1 Overview

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

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

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

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

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

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

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

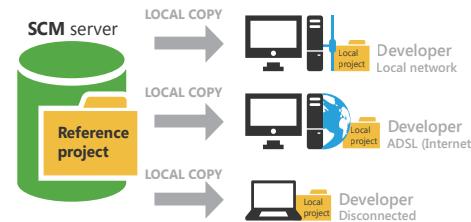
The SCM can also be used in offline mode (in a train, on a plane, ...).

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

#### Setup



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

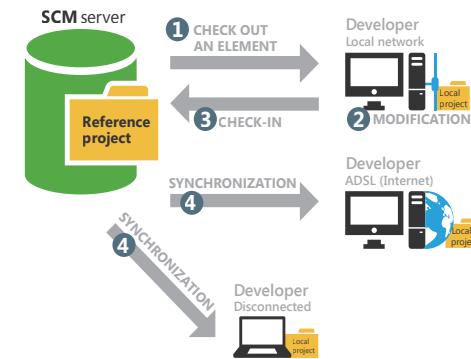
Each developer who is using the Source Code Manager retrieves a local copy of project.

#### Use

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

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

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



## 2 Creating a database for the SCM

### 2.1 Overview

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

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

The SCM database can be created:

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

### 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 1 system on your server (several disks that store the same information).
- Use a UPS to protect the power supply of your server.
- Perform backups of SCM database on a regular basis (at least once a week).
- Place the server in a "secure" area and use a firewall.

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

**The SCM database must be created once only.** WINDEV allows you to create this SCM database at different times:

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

Once the SCM database is created, all the WINDEV projects to share can be imported into this SCM database.

### 2.3 Backups

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

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

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

#### 3.1 Adding a project into SCM

To add a local project into the SCM:

1. Open your project in WINDEV.
2. On the "SCM" pane, in the "Project" group, click "Add the project".

3. In the wizard, select the SCM database to use.

4. In the wizard, select the SCM database to use.

Specify whether you want to use:

- a source database found on a network share (SCM database in HFSQL Classic format). Specify the directory of the SCM database (network computer or shared directory).
- an SCM database found on a HFSQL Client/Server server. The SCM database will be in HFSQL Client/Server format. In this case, you must specify:
  - the server
  - the database
  - the port used

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

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

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

#### 3.2 Sharing resources

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

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

### 4 Working with SCM

#### 4.1 Project options affecting the SCM

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

##### • Propose to get the latest version of elements when opening the project:

When opening a project found in the SCM database, this option proposes to retrieve the latest version of project elements.

By default, the latest version of elements is automatically retrieved.

##### • Propose to check the elements back in when the closing project:

When the project is closed, this option is used to display the list of elements that are currently

checked out in order for some of them (or all of them) to be checked back in.

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

##### • Check out/Check in the project automatically:

This option is used to automatically manage the "project file". If this option is checked, the project file is checked out only if the action performed requires it. Once the action was performed on the project, the project file is automatically checked back in.

This option is used to disable the "Master/Guest" management on the project. This option can also be enabled on the "SCM" pane, in the "Project" group, by expanding "Master/Guest" and by selecting "Manage the project check-out automatically".

#### 4.2 Checking out an element

##### The different check-out modes

The SCM proposes two modes for checking out project elements:

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

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

##### To change the check-out mode:

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

##### Opening a project element to modify its characteristics

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

1. Check out the element from the Source Code Manager.
2. Select the check-out mode of element. The check-out mode can be:
  - **exclusive**: no one else will be able to check out this element until it is checked back in. The element can be checked out for test only.
  - **for test**: the element can be modified but the modifications will not be checked back in.
  - **multiple**: the element can be checked out by other users. In this case, the differences between the different element versions can be viewed when the element is checked back in.
3. Validate. The element is opened. The title bar indicates that the element is checked out.

#### 4.3 Checking an element back in

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

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

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

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

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

#### 4.4 Management modes of project

Two management modes are available with the SCM:

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

##### Master and guest

The Source Code Manager distinguishes between 2 types of users:

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

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

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

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

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

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

#### Caution: Modifying the options of the project:

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

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

#### Automatic mode

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

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

## 5 Working in offline mode with the SCM

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

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

The principle is straightforward:

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

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

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

See the online help for more details.

## 6 SCM administrator

The SCM administrator is used to directly handle the different projects included in the source code manager.

It allows you to:

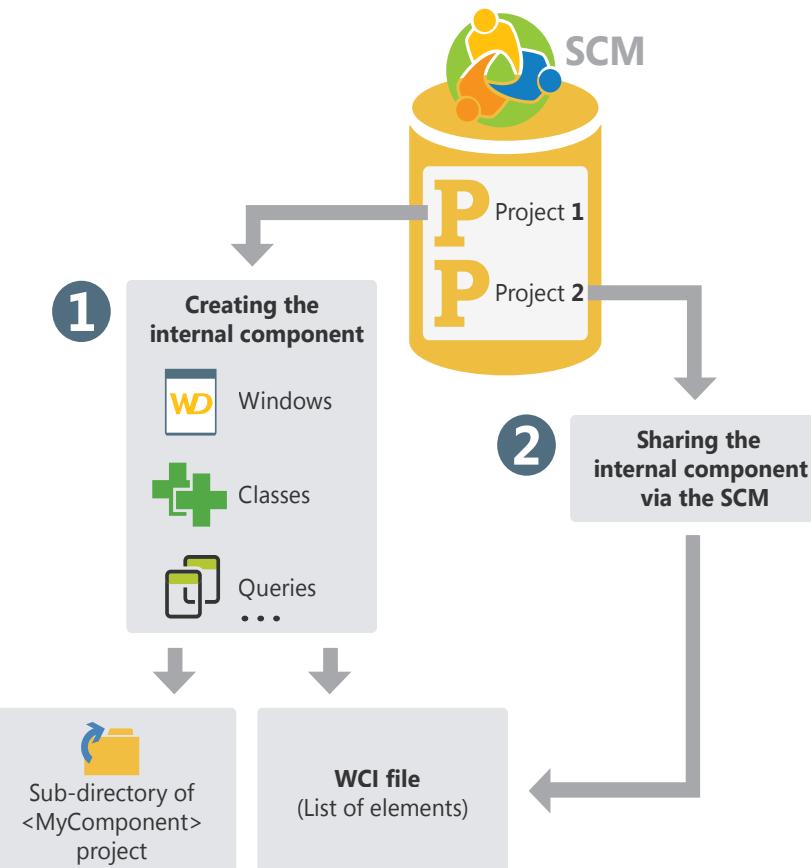
- manage the SCM databases (creation, connection to a SCM database).
- manage the branches.
- manage the files and directories found in a project of SCM database (add, delete, rename, ... files and directories).

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

# Internal component

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

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



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

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

# Internal component in practice

## 1 Overview

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

- **Organize a project:** the internal components can be used to group the project elements, by feature for example.
- **Share elements between different projects:** The elements found in an internal component can be shared between several projects via the SCM.

One of the benefits of the internal component compared to an external component is that the internal component can be debugged from the project that is using it.

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

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

- **Use a specific analysis:** The internal component accesses its own data files. The internal component and the project are using different runtime contexts.

This mode corresponds to the default mode if the component is using an analysis.

In this case, you must specify the analysis used by the internal component. You also have the ability to create a new analysis. This analysis will be associated with the internal component.

- **Validate the creation of internal component:** You will have the ability to:

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

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

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

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

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

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

## 2 Creating an internal component

### 2.1 The different steps

To create an internal component:

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

2. Specify the characteristics of the internal component:

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

- its caption.

- its description.

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

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

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

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

- **Use the project analysis or no analysis (full share):** The internal component accesses the data files of project. The internal component and the project are using the same runtime contexts. This mode corresponds to the default mode if the component is using no analysis.

- **Use the project analysis with different runtime context (advanced mode):** The internal component accesses the data files of project. The internal component and the project are using different runtime contexts. This mode is reserved to specific cases.

## 3 Sharing the internal components (via the 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 the SCM. The internal component will be

automatically found in the SCM.

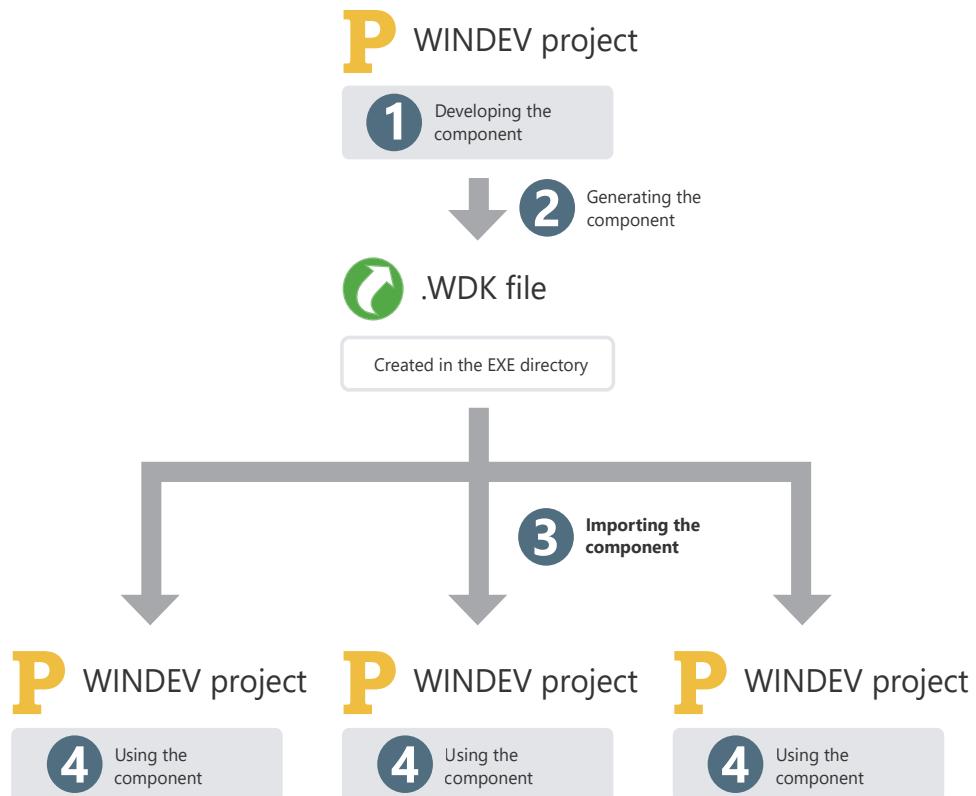
- import a project containing one or more internal components into the SCM. The internal components will be automatically included in the SCM.

# External component

An external component is a set of WINDEV elements: windows, 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,
- ...

An external component can be distributed to other WINDEV developers (freely or not). These developers will be able to easily include the feature proposed by the external component in their application without having to access the corresponding source code (if it was not distributed as well). Then, the external component will be included in the application and distributed along with it.



## External component in practice

### 1 Overview

#### 1.1 Definition

An external component is a set of WINDEV elements: windows, 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, ...

#### Notes:

- To optimize the organization of your projects, you have the ability to use internal components. See "Internal component in practice", page 72 for more details.
- In the rest of this chapter, "component" means "external component".

An external component can be distributed to other WINDEV developers (freely or not). These developers will be able to easily include the feature proposed by the external component in their application. The component will be included in the application and distributed with the application.

When creating the component, the author specifies:

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

**Note:** Examples of components are supplied with WINDEV. These components are accessible from the home window ("Open an example" option).

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

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

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

The component includes three files:

<Component Name>.WDK	<b>Component file.</b> 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 application that is using the component.
<Component Name>.WDI	<b>Description of component</b> (for developing applications only). This file is required to include the component in a project. This file must not be supplied in the setup procedure of the application that is using the component.
<Component Name>.WDO (optional file)	<b>Optional file</b> File in text format containing the list of additional elements supplied with the component (data files, .INI files, ...). See "The WDO file", page 80.

### 1.3 What is included in an external component?

A component contains:

- the different elements to distribute (windows, reports, ...). These elements can be accessible (or not) when importing the component into a WINDEV project.
- a short description of component.

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

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

## 2 Creating and generating an external component

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

The creation of a component is performed in several steps:

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

Then, the component can be distributed and reused.

## 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 supplying the necessary files directly.

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 component

The component is created and used on the same computer.

When importing the component into a WINDEV project, select the WDI file corresponding to the component. This file is found in the EXE directory of component project.

**Caution:** If the component is using specific files (data files, text files, ...), a <ComponentName>.WDO file must be created once the component is generated. This file contains the list of external files (data files, text files, ...) used by the component.

These files referenced in <ComponentName>.WDO will be automatically copied into the EXE directory of the project that is using the WINDEV component.

### 3.3 Simple distribution of 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 WINDEV 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 WINDEV (<ComponentName>.WDK file and <ComponentName>.WDI file).
- If necessary, the specific files handled by the component (data files, initialization files, ...) as well as <ComponentName>.WDO.

The <ComponentName>.WDO file contains the list of files that must be supplied with the component. See "Setup media", page 168 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 WINDEV component. This program installs all the files required for using the

## 4 Using an external component in an application

A component can be re-used at any time in any WINDEV application: you just have to know the directory of component files and import it into the project. The component can also be imported via the SCM.

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

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

To propose a setup procedure for a component:

1. Create the setup procedure of component: on the "Project" pane, click "Setup procedure".

2. In the wizard planes, specify:

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

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

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

"external component", page 76 for more details.

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

The method for creating the executable that is using the component is the same as the method for creating the executable for a standard WINDEV application. The DLLs required for the component to operate are automatically detected.

## 5 Deploying an application that is using an external component

### 5.1 Overview

The method for deploying an application that is using one or more components is the same as the method for deploying a standard application ("Setup procedure" on the "Project" pane).

When deploying an application that contains a component, the following files are automatically installed on the user computer:

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

### 5.2 Updating components and deployed executables

When updating a component and/or a deployed executable, you have the ability to install on the end-user computer:

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

See the online help for more details.

## 6 Modifying an external component

### 6.1 Overview

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

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

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

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

#### 1. The recompilation of the host project

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

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

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

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 into the component but they are not required to run the application.

### 6.2 The different types of compatibility

The management of compatibility is linked to the versions of a component. Indeed, if modifications have been made to the component, the applications 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 applications that use this component must always be greater than or equal to the version currently deployed.

This compatibility is automatically supported. If the version of the component installed with the application is less than the version used by the application, the application will not operate.

- the **forward compatibility**: using a new version of a component with the projects compiled with an earlier version can be forbidden. Therefore, the projects must be recompiled in order to use the new version of the component.

## 7 Advanced characteristics of component

### 7.1 Automatic documentation

A component must be supplied with a documentation in order to be re-used.

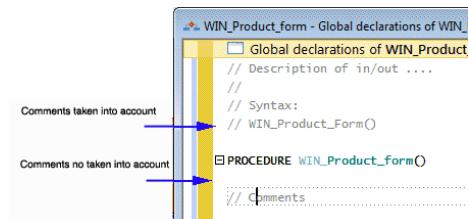
WINDEV simplifies the creation of this documentation by proposing:

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

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

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

- The comments found at the beginning of WLanguage procedures



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

#### When is the documentation generated?

The component documentation is generated during the first generation of component ("Generate" on the "Project" pane).

When generating the documentation:

- the comments found in the code are used to create the documentation.
- if there is no comment, WINDEV 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 re-generate 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 overview of component and the technical documentation of component are available:

- when a component is included in a project: on the "Project" pane, in the "Project" group, expand "Import" and select "Import an external component".
- at any time, when double-clicking the component icon found in the "Project explorer" pane or via "Description" from the popup menu of component.

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

## 7.2 Visibility of a 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 component, two files are automatically created in the EXE directory of current project:

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

These two files must be distributed along with the component.

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

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

### What is this WDO file?

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

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

This ".WDO" file can contain:

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

This file will be used when 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 an application", page 77 for more details.

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

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

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

## Distributing a component with WDO

To distribute a component that is using a WDO file:

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

In the wizard, you must:

1. Ask to modify the files to install ("Modify the list of files to install" 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 Windows applications, WINDEV allows you to generate other types of projects.



## Applications

The applications are the most common generation mode. The applications developed with WINDEV can target the following runtime platforms: Windows (32 bits or 64 bits) and Linux.



## Services

WINDEV can be used to generate services for Windows (32 bits or 64 bits) and for Linux (in Linux, the services are usually called daemons). A service is a specific type of application, started at the same time as the operating system and containing no GUI.



## Libraries and patches

A library is a unique file that groups several elements of a WINDEV project: windows, reports, etc. You have the ability to generate stand-alone libraries that can be used by other applications as well as corrective patches for an application that is already deployed ; this allows you not to have to reinstall the full application for a minor correction.



## External components

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



## SOAP or REST Webservices

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

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



## Applications and .Net assembly DLLs

WINDEV allows you to generate applications and assemblies for the .Net platform of Microsoft. A .Net assembly generated by WINDEV can be imported into a .Net application developed in another language in a way similar to an external component between two WINDEV applications.



## Java applications and applets

WINDEV allows you to generate Java applications as well as Applets. The Java applications can be run on any computer equipped with a Java virtual machine, regardless of its operating system. This allows you to build applications that operate in Windows, Mac OS or even Solaris for example.

A Java Applet is an application intended to operate inside a Web browser.



## Windows Store Apps applications

WINDEV allows you to generate Windows Store Apps applications. These applications can be used in tile mode or they can be distributed via Windows Store.

# Creating the executable in practice

When your application is entirely described (windows, reports, ...), you must create the corresponding executable program.  
The executable program (".EXE" file) can be run directly: there is no need to start the application

## 1 Creating the executable program of a WINDEV application

### To create an executable program:

1. Select (if necessary) the project configuration corresponding to the executable to create:
  - Project configuration for a Windows 32-bit executable
  - Project configuration for a Windows 64-bit executable
  - Project configuration for a Linux executable
2. On the "Project" pane, click "Generate".
3. The wizard for executable creation starts.
4. Specify:
  - the **general options** of the executable (name of executable, icon of executable program, splash screen, ...).
  - the **operating mode** of executable:
    - *Multi-instance*: This program can be started several times from the same computer: several instances will be opened at the same time.
    - *Single-instance*: This program can be started once only from the same computer: a single instance will be opened at a time. If the user tries to run another program instance, the current instance will be enabled.
  - the **use mode of library**. The library is optional. It can be created in a separate file (".WDL" extension) or included in the executable. Including the library in the executable program presents the benefit of having a single file to install or update but it increases the size of created ".EXE" file.
  - the **error message** that will be displayed if a

from WINDEV.

When the executable is created, the application can be deployed on the end-user computers by creating the setup program.

programming error occurs.

- the use of **patches for updates**. In this case, all you have to do is copy the file corresponding to the patch into the executable directory.
  - the languages supported by the executable.
  - the **files found in the library** (if there is one).
  - whether the components used in the project are included in the executable.
  - the **location of the data files of user groupware** (only if your application supports the user groupware).
  - the **location of the data files** that will be used by the executable.
  - the **version information**. This information will be displayed in the "Version" tab of the window for executable properties (accessible from the Windows explorer). The **version number** can be automatically **incremented** whenever creating the executable.
  - the mode for using the WINDEV framework (using a framework specific to the application or a framework common to all the WINDEV applications installed on the computer).
  - the manifest to use (for an execution on Windows Vista and later only).
  - whether a **project backup** must be performed.
5. Validate the executable description. The executable is automatically generated according to this information. The executable can be directly run after its creation.

# Java application in practice

## 1 Overview

WINDEV allows you to generate Java applications without even knowing the Java language. As for all WINDEV applications, the application is developed in WLanguage.

### 1.1 What is a Java application?

A Java application is entirely compiled in JAVA. If this application handles a database, this database must have a JDBC driver (MySQL, Oracle, ...). This driver is specific to each database (it can be downloaded from the site of database publisher).

The access to a HFSQL database is performed natively (in Windows and Linux only) via the SQL functions of WLanguage. No additional driver is required.

A Java application corresponds to a ".jar" file and it requires the Java runtime in order to be run.

**To compile (in Java) a Java project created with WINDEV**, Oracle J2SDK (version 1.4.2 or later) must be installed on the development computer. A version is available on the setup CD of WINDEV. This program can also be downloaded from the Oracle site.

### 1.2 Why develop a Java application with WINDEV?

You've developed a WINDEV application for Windows and you want to make this application accessible independently of the operating system? Generate a Java application so that the users working with Mac OS X for example will also be able to use your application.

This allows you to share your network applications.

### 1.3 Principle for generating a Java application with WINDEV

As mentioned above, there is no need to know the Java language to develop a Java application with WINDEV. You develop your application in WLanguage like any other WINDEV application and, when compiling the project, the windows and the WLanguage functions are automatically converted into Java.

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

## 2 Features for a Java project

### 2.1 Creating a Java project

A Java project is created like any other WINDEV project:

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

By default, a WINDEV project for Java corresponds to a specific project configuration. The type of project generation can be modified at any time. All you have to do is create a new project configuration.

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

- The window RAD or the full application RAD is specific to a database handled by a Java application.

- The compiler indicates whether the project contains functions or controls that cannot be used in Java. A purple message is displayed in the "Compilation errors" pane.
- Creating the archive is used to generate a ".jar" file in the EXE directory of the project.
- The setup module only handles the files required to deploy the Java application.

**Note:** we advise you create a new WINDEV project to develop a Java application rather than transform a WINDEV application into Java.

## 2.2 JAVA RAD

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

The RAD takes into account all the specific features of Java generation (controls, processes, functions that can be used in a WINDEV application generated in Java).

Regardless of the type of window to generate, the first screen of the wizard is used to specify the type of connection to the database:

- HFSQL (in Windows or Linux): the standard wizard for window creation is proposed.
- JDBC: you have the ability to specify the parameters for connecting to the database using the JDBC driver.

## 2.3 Test of Java project in the editor

During the test of Java project () , the project is compiled and started in Windows. The project is not compiled in Java.

To run the Java application, you must start the ".Jar" file found in the EXE directory of your project.

## 2.4 Compiling a Java project in WINDEV

The compilation of a Java project is used to:

- identify the programming errors found in the WLanguage code
- view the controls, processes, WLanguage functions not available in Java. These errors appear in purple in the "Compilation errors" pane. See the online help for more details.

**Note:** as indicated in "What is a Java application?", page 85, the Sun J2SDK must be installed on the development computer to compile your project in Java.

## 2.5 Creating a Java archive

WINDEV allows you to:

- create and compile in a Java compiler the different Java files required by the application
- create a Java archive that can be run directly:
  - a standard Java application,
  - a Java applet,
  - a Java Web Start application.
- use the project windows from your own Java source files. In this case, you have the ability to generate the corresponding Eclipse project.

To create the Java archive, all you have to do is create the executable: on the "Project" pane, in the "Generation" group, click "Generate". A wizard helps you create the Java archive.

## 2.6 Deploying a Java application

Like for any application, WINDEV allows you to create a setup program for a Java application. This setup procedure is available for Windows only. The setup program will be an executable and it will be used to install the necessary files as well as the Java runtime (JRE). For the other platforms, your files will have to be deployed manually.

To start the wizard for setup creation, on the "Project" pane, in the "Generation" group, click "Setup procedure".

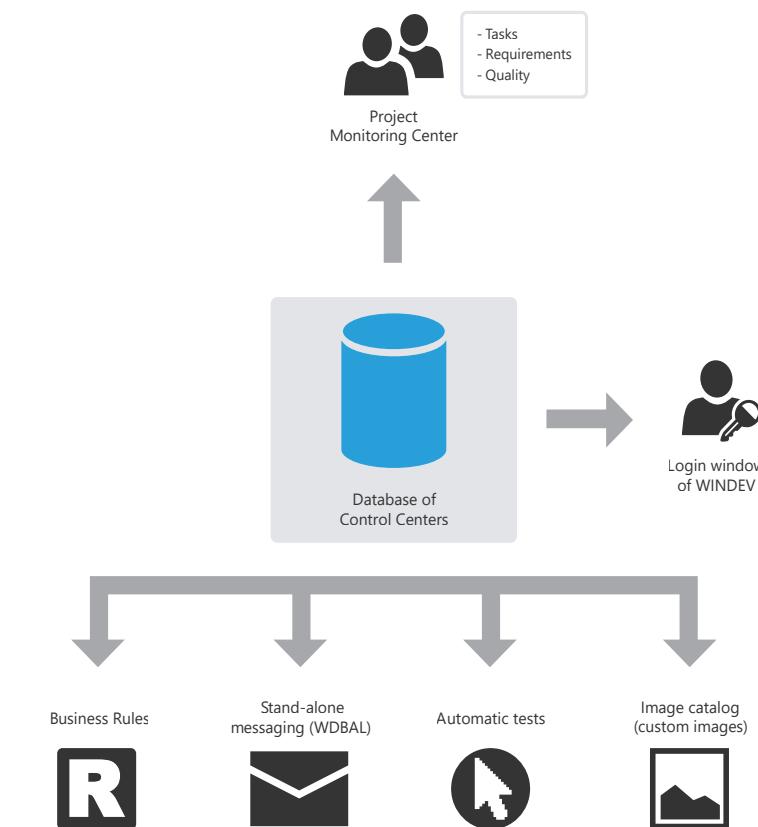
See the online help for more details (keyword: "Java").

# Project Monitoring Center

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

- Manage the requirements of a project,
- Manage the monitoring of a project (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 WINDEV:



**Note:** When installing WINDEV, the setup program proposes:

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

# Managing requirements

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

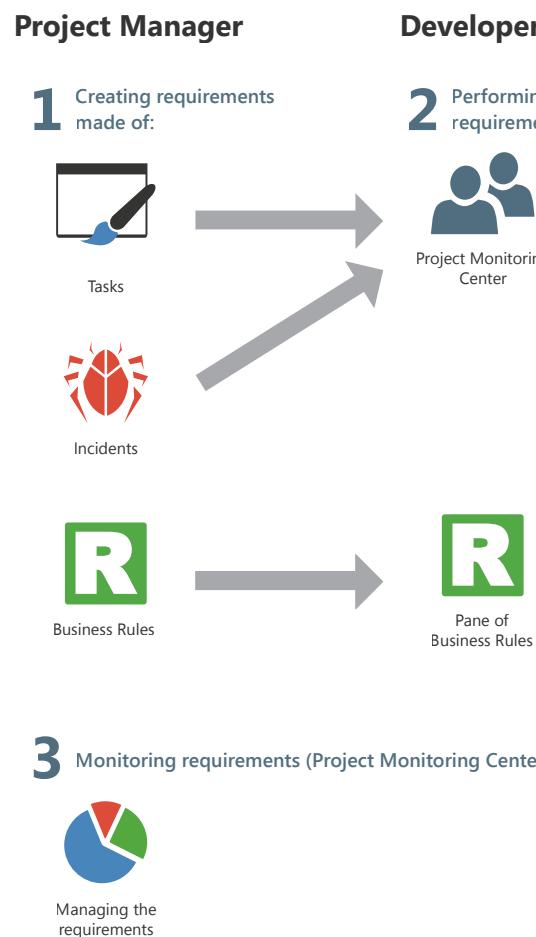
The Project Monitoring Center allows the project manager to manage the requirements of a development project.

To do so, you must:

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

Each developer performs the different tasks assigned to him.

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



# Managing tasks

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

Assigned to	Project	Title	Estimated time	Time passed	Avail date
ANNE	CRM	Completion of requirement: New window	5 hours	578 days	1/30/2018
ANNE	CRM	Completion of requirement: Connection	5 hours	0 day	1/31/2018
ANNE	CRM	Completion of requirement: data files	1 day	0 day	2/1/2018
ANNE	MyCarlsGreen	Completion of requirement: Collaborator schedules	1 day	0 day	2/2/2018
ANNE	MyCarlsGreen	Completion of requirement: site redesign	4 hours	0 day	2/2/2018
ANNE	MyCarlsGreen	Completion of requirement : customer contact form	3 hours	0 day	2/5/2018
ANNE	CRM	Servers management	1 day	0 day	2/6/2018

Details of task #77: Completion of requirement: New window  
 You must add a new window to manage the new customers.  
 This window must contain:  
 - a spreadsheet  
 - a dashboard

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.

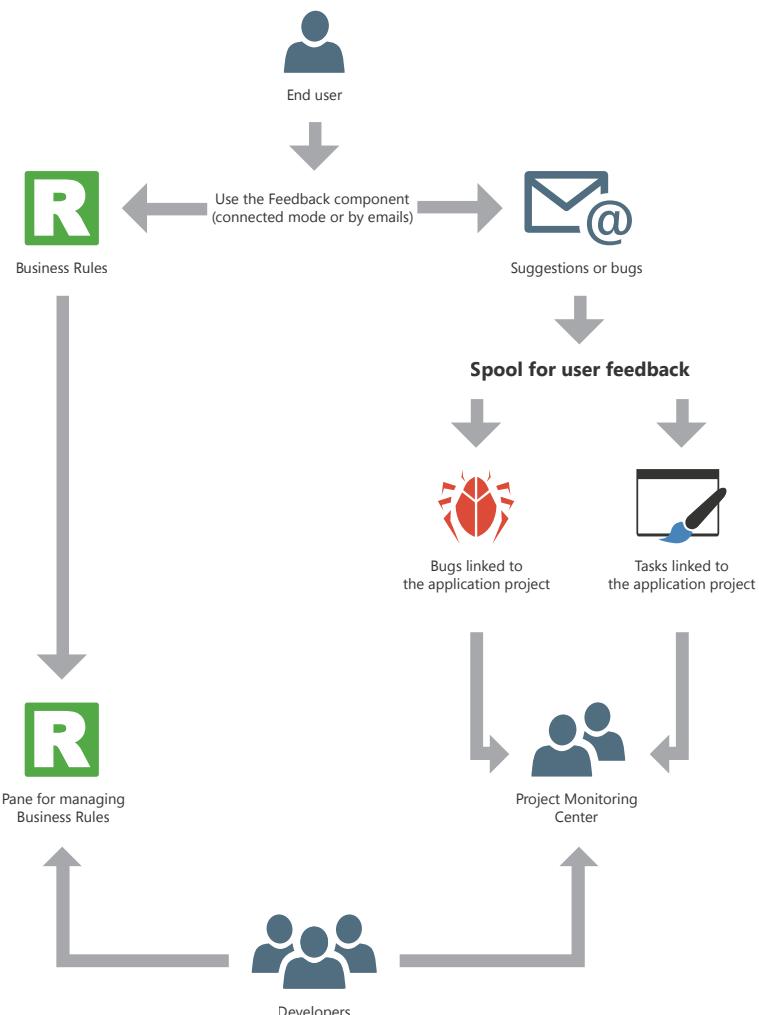
A task can be linked to a project element (window, report, ...). Whenever the relevant element is opened, the time spent on this element is counted and stored in the Project Monitoring Center. Conversely, the element corresponding to the task that you want to perform can be automatically opened from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

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

# Managing suggestions and incidents

The Control Centers also allows you to manage the user feedback regarding your applications.

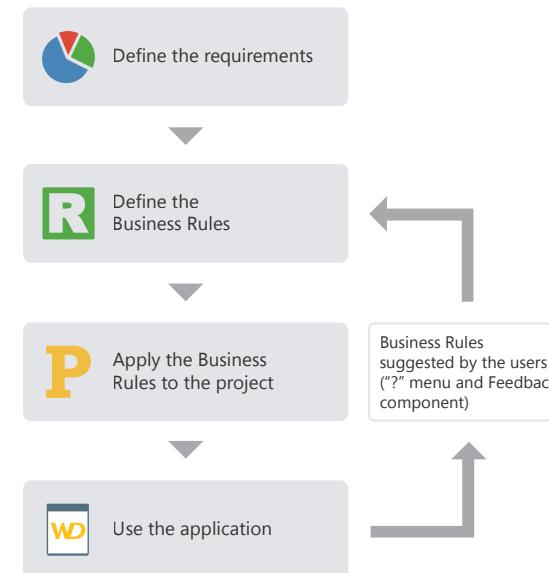
If your application contains an automatic help menu ("?" menu with "Send suggestions..."), the users of your application will be able to easily send suggestions, incidents or business rules.



**Note:** if a fatal error occurs when running the application, an error report can be automatically generated and sent to the developer via the "Feedback" component supplied with WINDEV.

# Managing business rules

WINDEV allows you to manage the business rules. A business rule is used to define a specific operating mode or to specify a particular process. For example: calculating a specific VAT rate, the rules for changing customer status, 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 associated with the project are displayed in the "Business rules" pane of the development environment. This pane displays the number of project elements to which the business rules apply and the percentage of rule currently implemented.

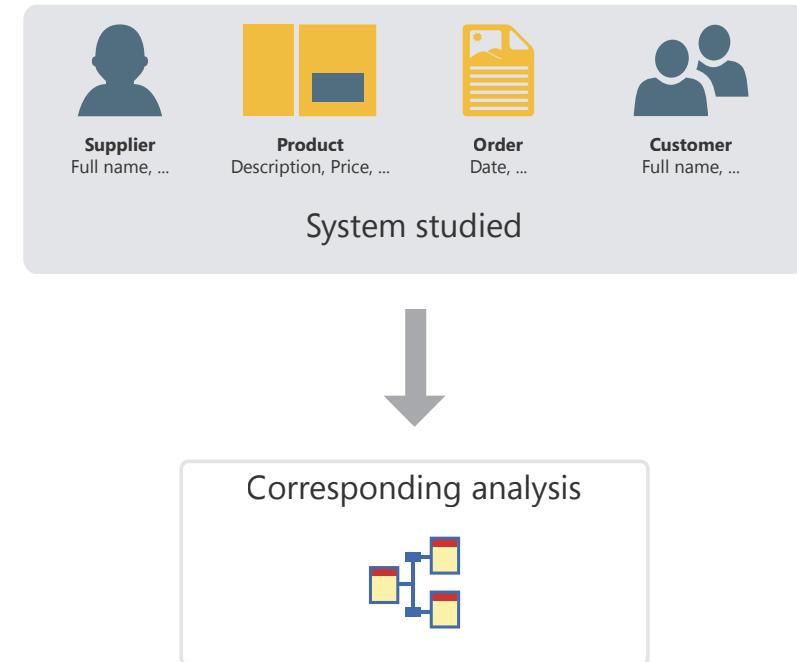
DEVELOP 10 TIMES FASTER

PART 3

Databases

# Analysis: Database structure

When a WINDEV, WEBDEV or WINDEV Mobile project is using data files, this project must be associated with an analysis. An analysis allows you to describe the structures of data (files, items, ...) used in your project.



The data model editor is used to easily create an analysis. The analysis of a WINDEV project corresponds to the LDM (Logical Data Model). The entire structure and data organization are described: the data is grouped by 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 data 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 the data (files, items, ...) used in your project.

Two methods can be used to describe an analysis:

- **1st method:** Direct creation of analysis (which

means Logical Data Model (LDM)).

- **2nd method:** <strong>Creating the Conceptual Data Model (CDM) then creating the analysis from CDM. See the online help for more details.

This chapter presents the first method.

## 2 Creating an LDM (Logical Data Model)

In WINDEV, 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 project analysis (<Project Name>.ANA directory). This directory must be accessible in read/write.

- **the analysis caption** that briefly describes the purpose of analysis.

- **whether the analysis must be associated with the current project.**

- the type of database used by the project.

3. The creation of the first data file is automatically proposed.

4. Create all the elements (data files, items and links) of your analysis.

### 2.2 Adding a data file 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 WINDEV.
- A data file imported from an existing database (using a specific format for example).

The following paragraphs explain how to create a data file.

To create a new data file:

1. On the "Analysis" pane, in the "Creation" group, click "New file". The wizard for creating data files starts.

2. Select "Create a new description of a data file".

3. Specify:

- **the name of data file.** This name is the logical name of data file. It will be used to handle the data file.

- **a short description of data file** that summarizes the subject of data file.

- **the representation of a record** in the data file. This representation improves the legibility of questions asked when describing links. This option must be preceded by an indefinite article (A or AN).

- whether the data file includes an "**Automatic identifier**" item. The value of this item is unique for each record and it is automatically calculated by WINDEV.

- **the type of database associated with the file.** Depending on the selected type, this data file will be handled by the HSQL engine, by an OLE DB driver or by one of the native accesses of WINDEV (SQL Server, Oracle, ...).

4. The created data file becomes the current file. The window for describing the file items is automatically opened. It allows you to describe the file items.

**To import a preset description of a data file:**

1. On the "Analysis" pane, in the "Creation" group, click "New file". The wizard for creating data files starts.

2. Select the option "Select a description among the preset data files".

3. Choose the preset data file to create (you have the ability to select several ones). This data file will be imported into the current analysis. This data file can be modified later.

4. Select the file items to store. These items can be modified later.

5. 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.

**Two solutions can be used to import an existing file description:**

**Solution 1: From the data model editor**

1. On the "Analysis" pane, in the "Creation" group, click "New file". The wizard for creating data files starts.

2. Select "Use the data files from an existing database" and select the type of database.

3. Specify the source database containing the descriptions to import and the type of this source database. Depending on the selected type, specify the requested information.

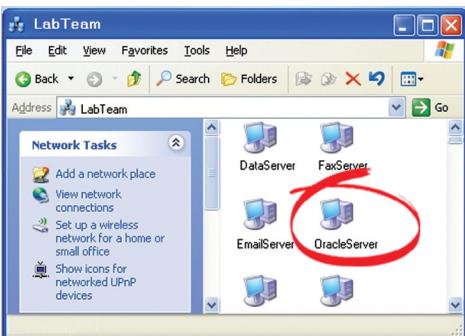
4. Select the tables or the data files whose description must be imported and validate.

5. The imported data file is automatically inserted into the current analysis.

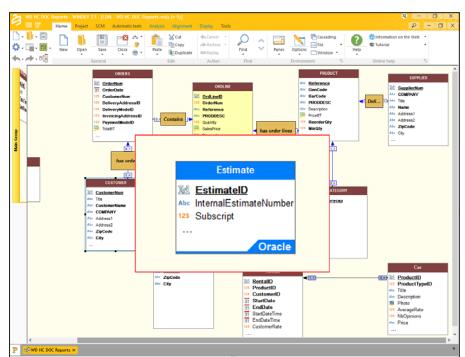
### Solution 2: From the Windows explorer

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

For example:



Here, an Oracle database, on the server. All you have to do is drag the database name ...



... to transfer its description into the data model editor.

## 2.3 Creating an item

### To create an item:

- Double-click the data file where the item will be created. The description window of file items is opened.
- Click the first empty row in the table of items.
- Specify the name, the caption and the type of item in the table.
- In the right section of the screen, specify the details about the new item (type, size, default value, sort direction, ...).
- In the bottom section of the screen, specify the details about the shared information.
- Validate the item description.

**Note:** you also have the ability to create an item from the meta-types proposed by WINDEV. To do so, click the "+ M" button on the right of table. The list of available meta-types is displayed.

## 2.4 Creating a link

Different types of links can be created between files. See "Setup: Summary table", page 168 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, an **owner** file and a **member** file are found:

- the owner file is the owner of the key.
- the member file is a member of analysis files containing a copy of the key.

To manage the link between two data files, the key of the owner file is copied into the member file.

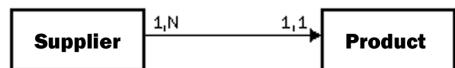
### To create a link:

- On the "Analysis" pane, in the "Creation" group, click "New link". The mouse cursor turns into a pen.
- Select the two data files to link. The window for link description is automatically opened.
- To define the cardinality between the two data files:
  - 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.
- To describe the advanced cardinalities, check "Display the advanced cardinalities" and answer the questions asked.
- Type the link caption by briefly describing the purpose of link.
- Specify the keys to link.
- Define the integrity rules. These rules are used to ensure the data integrity when one of the relation keys is modified or deleted.
- Validate. The link is automatically created.

For example, the key of Supplier file is copied into each record of Product file. Several records of the Product file can have the same key in the Supplier file:

- the owner file is Supplier file,
- the member file is Product file.

This type of link is represented as follows:



## 3.2 The cardinalities

The cardinalities are used to count the links between data files.

The cardinality is defined according to the answers to the two following questions:

- For each file record, what is the **minimum** number of records in the other data file to which this record is linked to?

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.

- For each file record, what is the **maximum** number of records in the other data file to which this record is linked to?

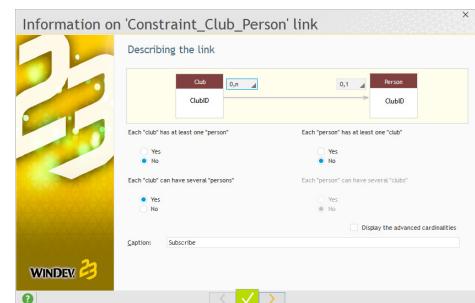
This answer provides the second part of cardinality (**maximum cardinality**):

- if the answer is "a single one", the cardinality is X,1.
- if the answer is "several", the cardinality is X,N.

The answer to these two questions defines the cardinality that can be: 0,1 ; 0,N ; 1,1 ; 1,N

The description of cardinalities is fundamental: it is the basis for respecting the referential integrity of the database.

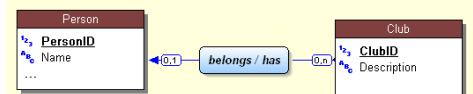
To be clearer, these two questions can be broken down into four questions. For example, to describe a link between Club file and Member file:



- each "Club" has at least one "Member": Yes/No?
- each "Club" can have several "Member": Yes/No?
- each "Member" has at least one "Club": Yes/No?
- each "Member" can have several "Club": Yes/No?

### Example of cardinalities:

This example presents two different cardinalities:



- Cardinality 0,1:** A person can be a member of a single sport club. This person does not have to belong to a sport club.
- Cardinality 0,N:** A club can have no member or several members.

## 3.3 The advanced cardinalities

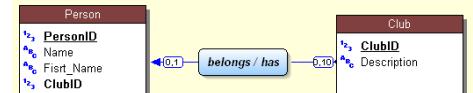
The cardinalities can be defined more precisely. We talk of **advanced cardinalities**.

These cardinalities are used to exactly specify the minimum cardinality and the maximum cardinality.

To define the advanced cardinalities, check "Display the advanced cardinalities" in the link description.

### Example of advanced cardinalities:

This example presents two different cardinalities:



- Cardinality 0,1:** A person can be a member of a single sport club. This person does not have to belong to a sport club.
- Cardinality 0,10:** A club can have nobody or up to 10 people as members.

### 3.4 The referential integrity

The referential integrity of a database corresponds to the respect of constraints implied by the links between data files.

The referential integrity consists in checking that:

- if a record is deleted from the owner file, the corresponding records are also deleted from the member files,
- if a record is added into a member file, a corresponding record exists in the owner file,
- if a record is modified in the owner file, the unique key is not modified, ...

The check of referential integrity depends on the nature of the link between the data files.

See the online help for more details.

### 3.5 The different types of links

Several types of links can exist between the data files:

- parallel
- optional
- complement
- shared
- complex

#### Parallel link

For a parallel link, *each record* found in a data file (Product) is linked to *a record* found in another data file (Information), and *conversely*.



To manage a parallel link, the identifier of Product file is copied into the Information file. This identifier is also a unique key in the Information file.

The owner file is Product and the member file is Information.

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

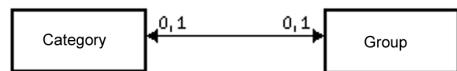
The parallelism of records in the file is respected if the following operations are simultaneously run on the two files:

- creating a record
- deleting a record
- reindexing with compression

#### Optional link

For an optional link:

- *Each record of a data file (Category) is associated with no record or with a single record* in another data file (Group).
- *Each record of the other data file (Group) is associated with no record or with a single record* in the first data file (Category).

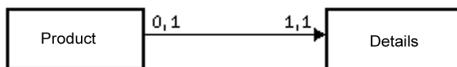


An optional link is performed by copying the identifier of each file into the other file.

#### Complement link

For a complement link:

- *Each record found of a data file (Product) is associated with no record or with a single record* found in another data file (Details).
- *Each record found in the other file (Details) has at least one record* associated with the first data file (Product).



The complement links are quite common. They are used when a record can have additional optional information.

To manage a complement link, the key of Product file is copied into Details file. To ensure the maximum cardinality of 1, it remains a unique key. The uniqueness of this key prevents from inserting more than one record in Details for a record of Product.

The owner file is Product, the member file is Details.

#### Shared link

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

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

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

Depending on the cardinality, we can distinguish between four types of shared links:

- Shared link with a 0,n - 0,1 cardinality
- Shared link with a 0,n - 1,1 cardinality
- Shared link with a 1,n - 0,1 cardinality
- Shared link with a 1,n - 1,1 cardinality

#### Shared link with a 0,N - 0,1 cardinality

With this type of link:

- 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 0,N - 1,1 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 1,N - 0,1 cardinality



With this type of link:

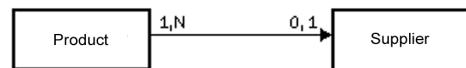
- an owner may have no member (a supplier may have no product).
- each member has a single owner (each product has a single supplier).

**Note:** This type of link is quite common. WINDEV allows you to automatically create the windows used to manage the data files linked by a 0,N - 1,1 link.

#### Shared link with a 1,N - 0,1 cardinality

With this type of link:

- each owner has at least one member (a supplier has at least one product).
- a member may have no owner (a product may have no supplier).



#### Shared link with a 1,N - 1,1 cardinality

With this type of link:

- each owner has at least one member (each supplier has at least one product).
- each member has a single owner (each product has a single supplier).



#### Complex link

For a "complex" link, you must manage a link file, called *relation file*.

The relation file will be automatically created. It will contain a unique key 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 (Order file) can contain one or more products. A product (Product file) can be used in several orders.

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

The OrderLine file contains:

- a unique key containing the keys of Product and Order.
- the number of products ordered.

## 4 Operations that can be performed on an analysis

WINDEV allows you to perform the following operations on an analysis:

- Duplicate/Copy an LDM: Duplicating an LDM allows you to have two identical LDMs with different names.
- Delete an LDM.
- Rename an LDM.
- Associate an LDM with a project.
- Enlarge or reduce the display of LDM in the editor.
- Move the display of LDM in the editor.

WINDEV 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.

WINDEV allows you to perform the following operations on an item of a data file found in the data model editor:

- Duplicate/Copy an item.
- Delete an item.
- Rename an item.

See the online help for more details.

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

The options available in the version manager are:

- Restoring an analysis.
- Canceling the last generation.
- Resetting the version number to 1.

### 6.1 Restoring an analysis

To restore an analysis from the version manager:

1. Select the version to restore.
2. You can:

- restore the version in the specified directory. An independent analysis corresponding to the selected version is created. This option is selected by default.
- overwrite the current analysis.

3. Validate. The restore operation is performed.

**Caution:** The data files 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 data files corresponding to the restored analysis version or delete the existing data files in order to re-create them.

### 6.2 Canceling the last generation

Canceling the last generation of the analysis is used to restore the analysis to its status before the last generation. The modifications performed since 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.

## 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:

- Checking the modifications made and generating the files of analysis description.
- Automatic modification of accessible data files (files found in the "EXE" directory of project).
- Synchronizing the project.

## 6 Managing the analysis versions

WINDEV allows you to manage the different analysis versions:

1. On the "Analysis" pane, in the "Analysis" group, expand "Generation".
2. Select "Manage the versions".
3. 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

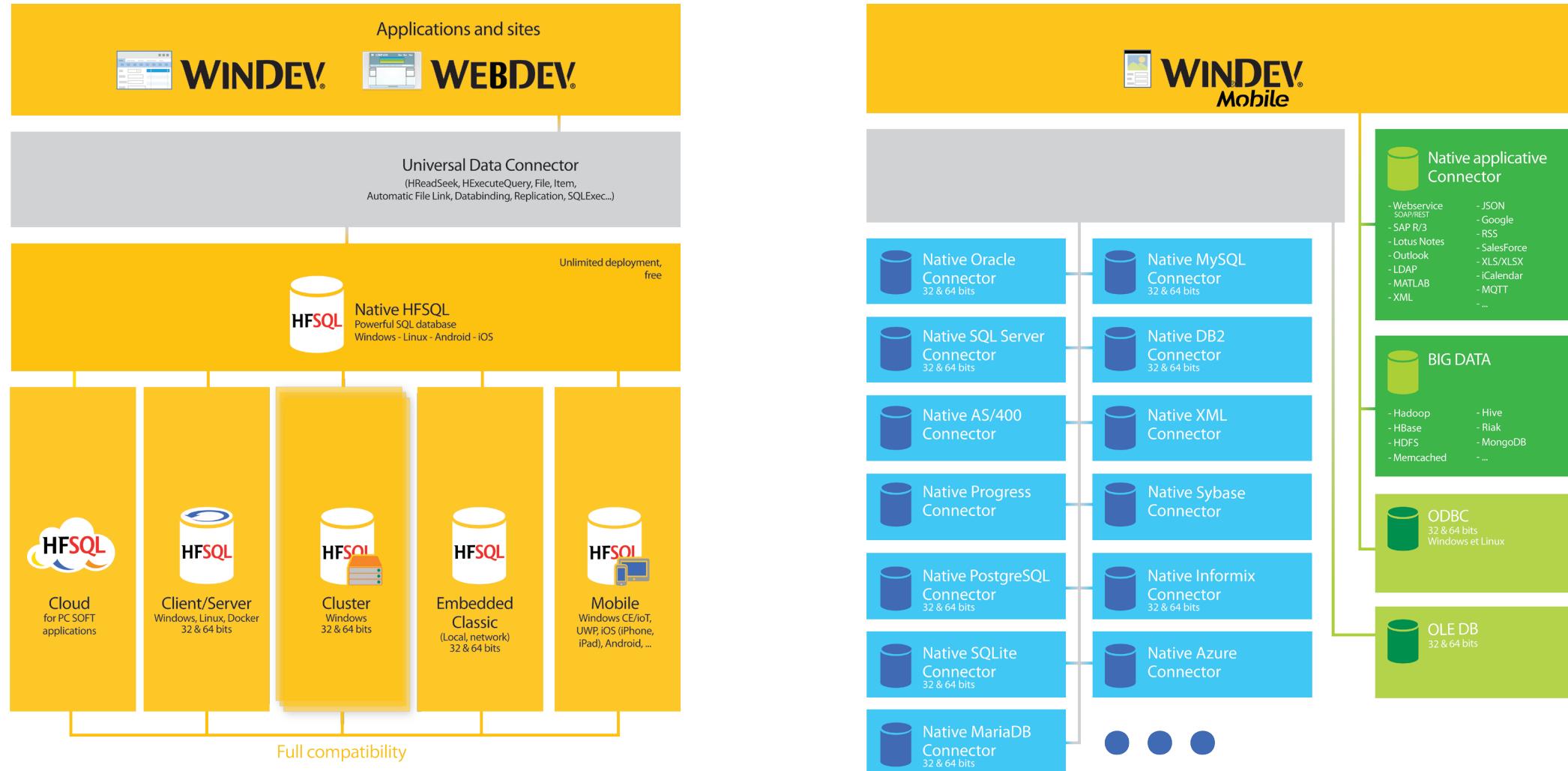
of analysis, the caption on the right indicates the generation date.

The "small rectangles" displayed between two versions have the following meaning:

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

# The different types of accessible files

WEBDEV, WINDEV and WINDEV Mobile propose a simple access to most of the databases available on the market.

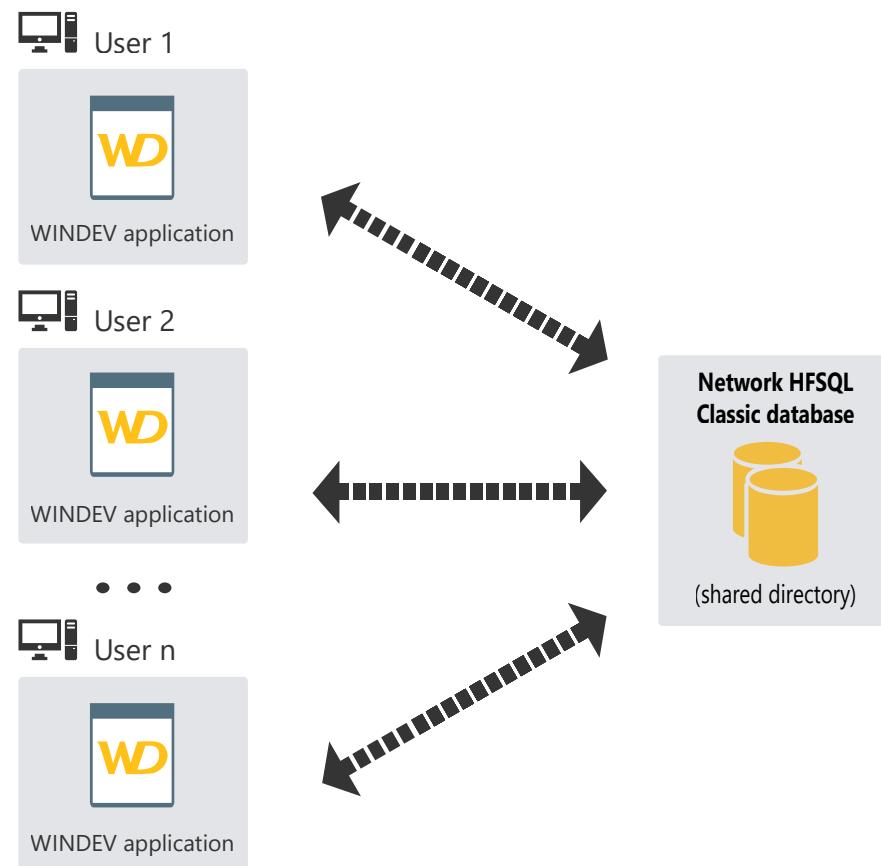


# HFSQL Classic

HFSQL Classic is a powerful system for managing relational databases (RDBMS) that can be used on a single computer or in local network. HFSQL Classic does not require the setup of a database server.

The characteristics of HFSQL Classic are as follows:

- A HFSQL Classic application can be run on different user computers distributed in local network.
- The data files are found in a directory that can be accessed by the user computers (a shared directory on the network for example). Each user computer physically accesses the data files.
- The processes (query, read/add in a data file, ...) are performed on each user computer.

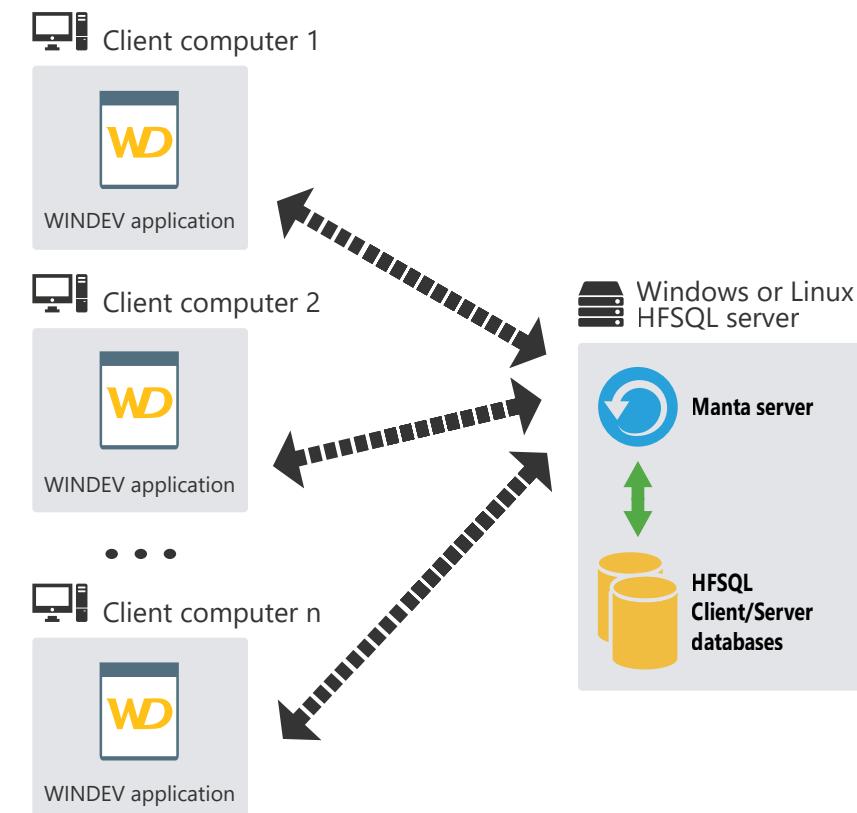


# HFSQL Client/Server

HFSQL Client/Server is a powerful system for managing relational databases (RDBMS) in Client/Server mode.

The characteristics of HFSQL Client/Server are as follows:

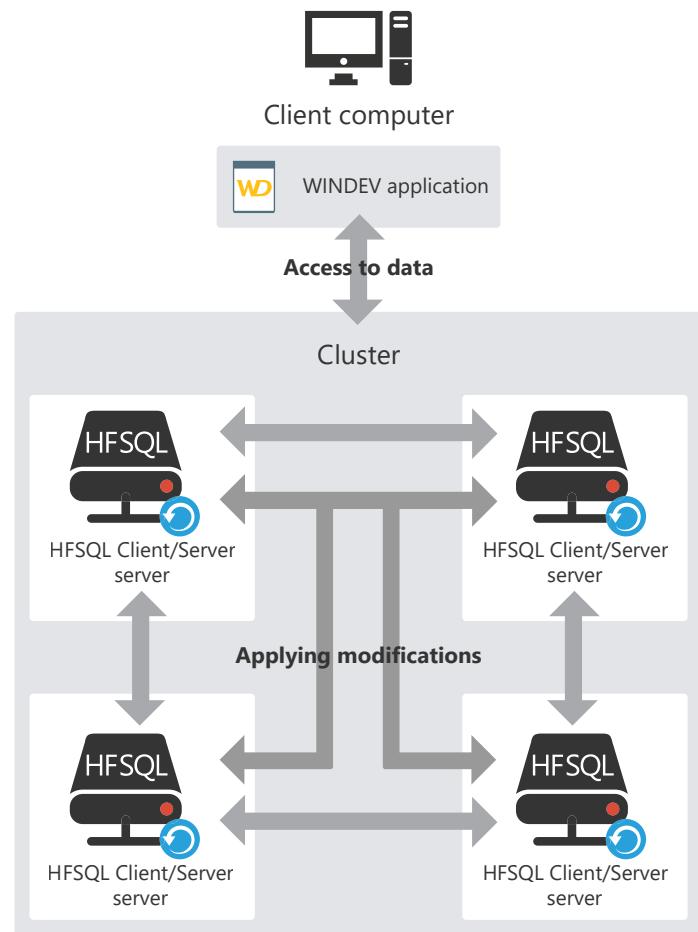
- A HFSQL Client/Server application can be run on different user computers (called client computers) that communicate with the database server via the network.
- The data files are found on a server. Only the server physically accesses the data files.
- All the processes (query, read/write operation in a data file, ...) are performed on the server.



# HFSQL Client/Server cluster

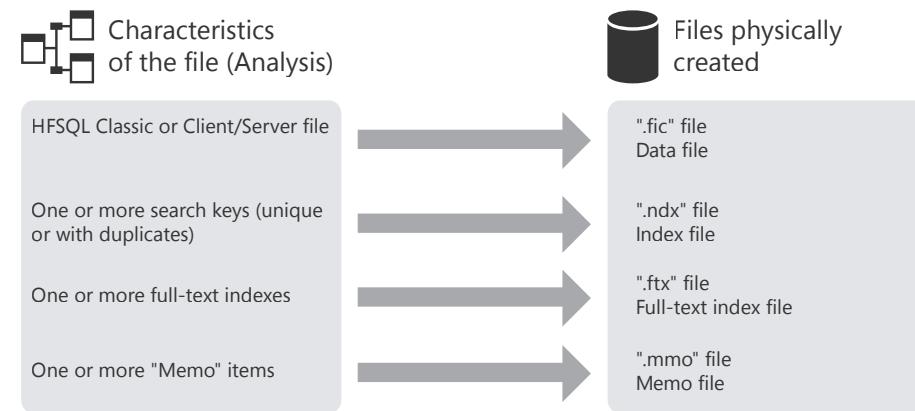
HFSQL Cluster is an extension of the database model of HFSQL Client/Server. In a database cluster, all the HFSQL servers contain a copy of the databases and they are synchronized in real time.

- The read load can be balanced between the different servers.
- The physical configuration can evolve without any interruption for the client computers.
- If one of the servers crashes, the client is automatically redirected to an operating server.



# HFSQL: the physically created files

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



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

# Associating the controls with data

A window can display information coming from:

- a database: the controls are directly linked to the items of data files or queries available in the database.
- variables found in the application code (variables global to the window or to the project or parameters passed to the window).

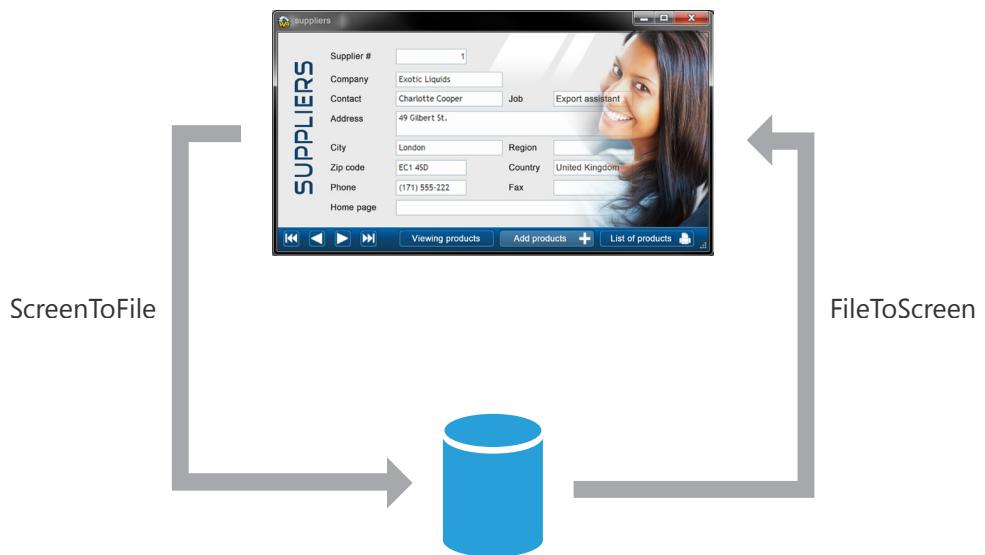
To display this information in a window, the controls of this window must be linked to:

- the different items of the database.
- the available WLanguage variables.

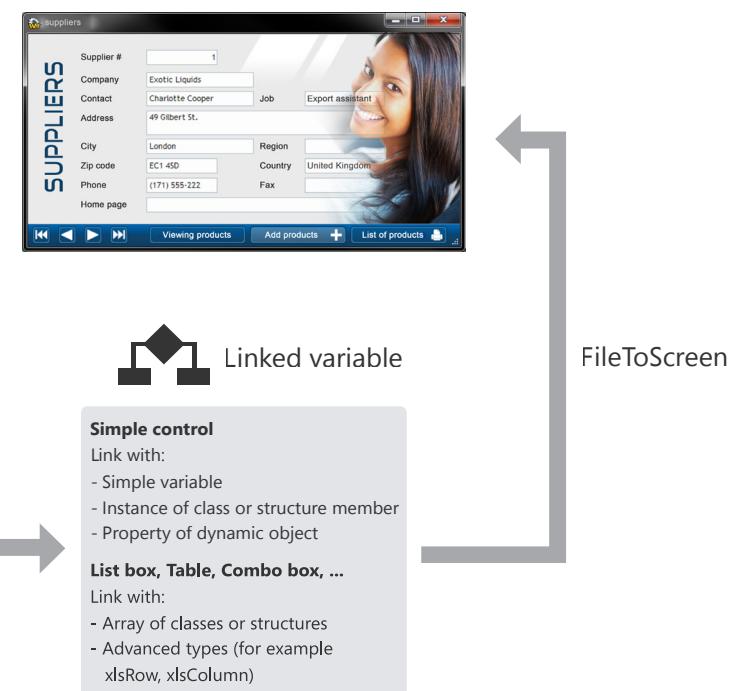
The method for displaying and retrieving the information is straightforward:

- The link between a control and an item or between a control and a variable is defined in the window editor when describing the control ("Link" tab).
- **ScreenToFile** is used to update the record or variable with the data displayed on the screen.
- **FileToScreen** is used to update the data displayed on the screen with the information saved in the data file or with the information saved in the variable.

## Link between control and item



## Link between control and variable

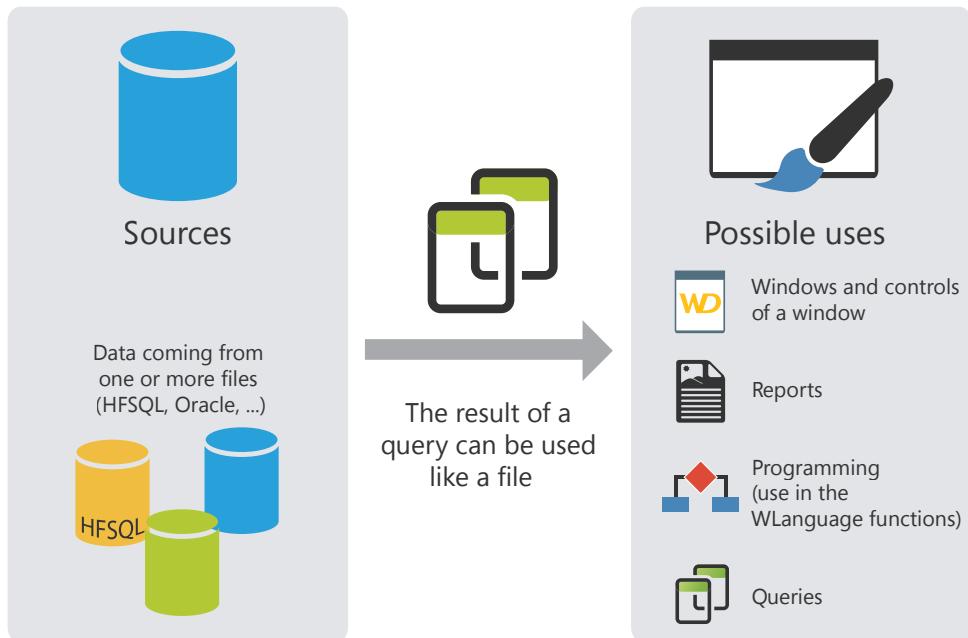


# The queries

A query is used to interrogate a database 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.



Data coming from  
one or more files  
(HFSQL, Oracle, ...)



The result of a  
query can be used  
like a file

Windows and controls  
of a window

Reports

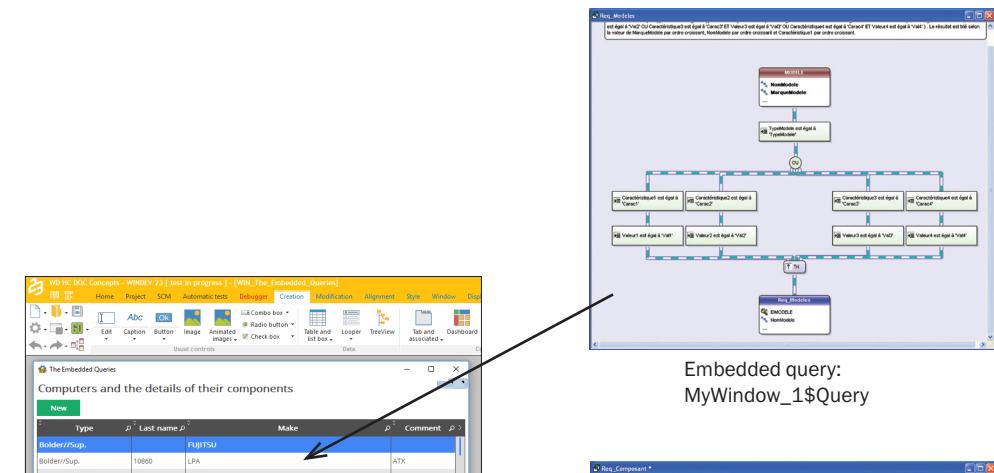
Programming  
(use in the  
WLanguage functions)

Queries

**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 window 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.

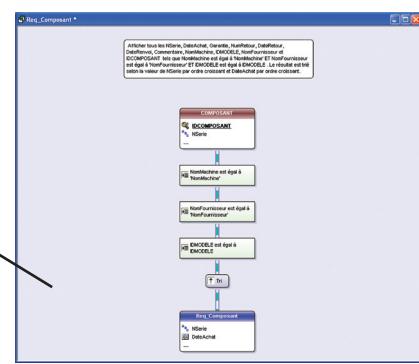


Embedded query:  
MyWindow\_1\$Query

Type	Last name	Make	Comment
Folder//Sup.	10800	LPA	ATX
Network card	1211TX	SMC	
Screen	123DE	DELL	
Video card	258 BUSINESS	MATROX	

Type	Make	Model	Supplier Name	Purchase Date	Guarantee
##	##	##	HP	OCTOBER 12	
##	##	##	DELL	NOVEMBER 12	
##	##	##	IBM	NOVEMBER 20	
##	##	##	COMPAQ	DECEMBER 01	
##	##	##	HEWLETT	DECEMBER 05	



Embedded query:  
MyWindow\_2\$Query

In this case, the query is integrated in the window. It is found in the WDW file corresponding to the window. If the WDW file is copied (into another project for example), the embedded queries used by this window will also be copied.

# The Table/Looper controls

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 structure of data file was described in the data model editor and the data was typed in the application for example. Browsing the data file allows you to display the 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 displayed data that is not linked to the data file is not kept when refreshing the display of Table control (when scrolling for example).

The number of records displayed in the control can be limited by a filter (**HFilter** used in the initialization code of control).

Several WLanguage functions can be used to handle the browsing Table controls. These functions start with "Table".

**To add a record into the Table control**, add the record into the corresponding HFSQL data file (**HAdd**) and redisplay the Table control with **TableDisplay**.

**To delete a record from the Table control**, delete the record from the corresponding HFSQL data file (**HDelete**) and redisplay the Table control with **TableDisplay**.

## Memory Table control

A memory Table control is used to directly display the data loaded in memory. The data is added into the Table control by programming (by **TableAddLine** for example).



The data being found in memory, the Table control allows you to perform all the operations on the data (sort on any column, search performed in the columns, ...).

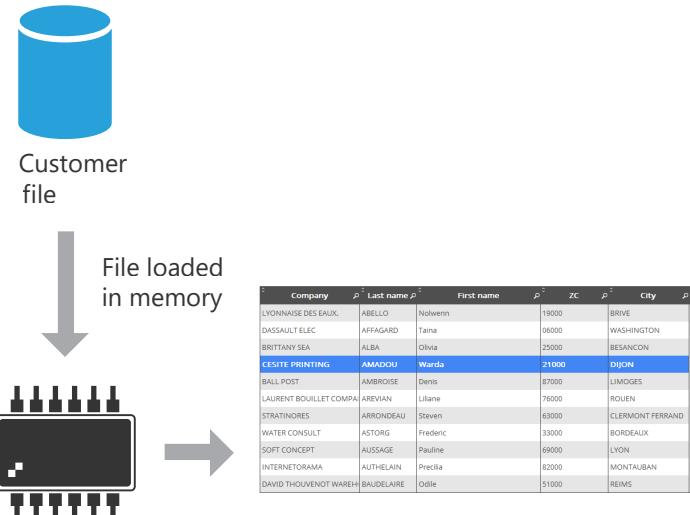
## Browsing Table control loaded in memory

The browsing Table controls loaded in memory combine the benefits of browsing Table controls with direct access with the benefits of memory Table controls.

The Table control is linked to the data file but the content of data file is entirely loaded in memory. The sort and the search are available for all the columns.

The data not linked to the data file is kept when handling the scrollbar (Check Box column for example).

The records found in the data file being loaded in memory, this type of control is recommended for the data files containing less than 100 000 records (to avoid memory overflow).



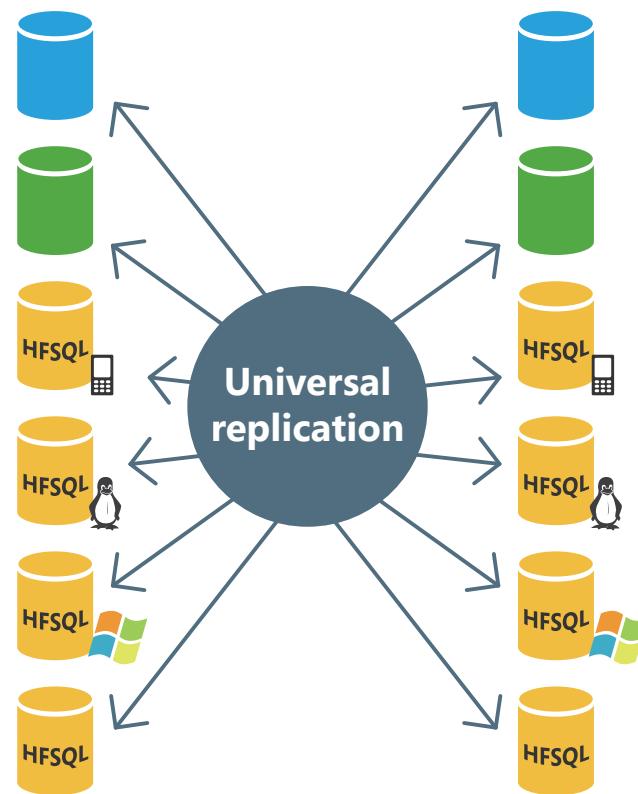
**Note:** The different fill modes (memory, browsing, browsing loaded in memory) are available for the List Box controls, Combo Box controls, Table controls, Looper controls,

...

# Universal replication

The universal replication is used to keep databases of identical or different formats updated (HFSQL, Oracle, SQL Server, ...). You can for example perform a synchronization between a HFSQL Classic database and an Oracle database.

The universal replication is using a centralized model: all the databases are synchronized with a master database. Then, the master database applies the modifications to the other databases.



The synchronization can be adapted to special cases. For example, you have the ability to retrieve the records regarding a specified product or the records created at a given date, manage the conflicts, display a configuration window, ...

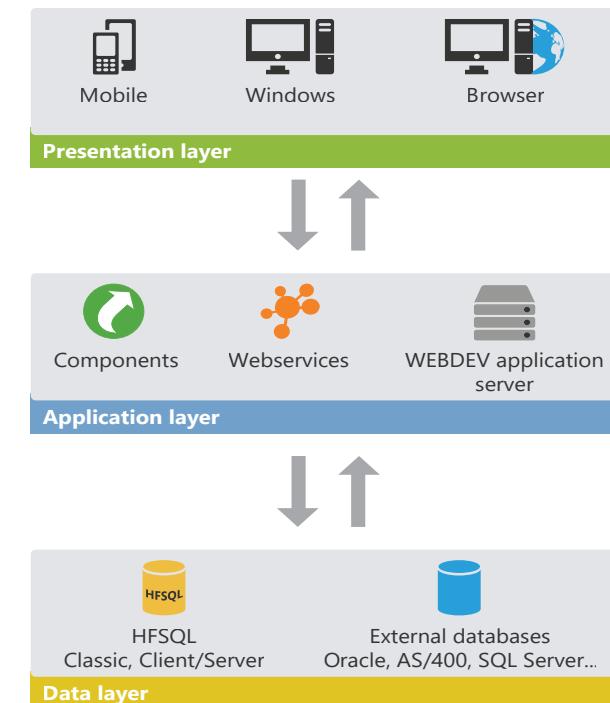
These changes must be done by programming via [HRplFilterProcedure](#).

# 3-tier architecture

The 3-tier architecture is a model for application architecture. Its basic principle consists in splitting an application into 3 layers:

- the presenter layer: corresponding to the display,
- the process or application layer: corresponding to the business processes of application,
- the layer for accessing persistent data.

The reason for separating the layers is to facilitate the maintenance and the future evolutions of the 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.



The 3-tier architecture is fully compatible with the development of applications with WINDEV.

DEVELOP 10 TIMES FASTER

PART 4

Advanced  
concepts

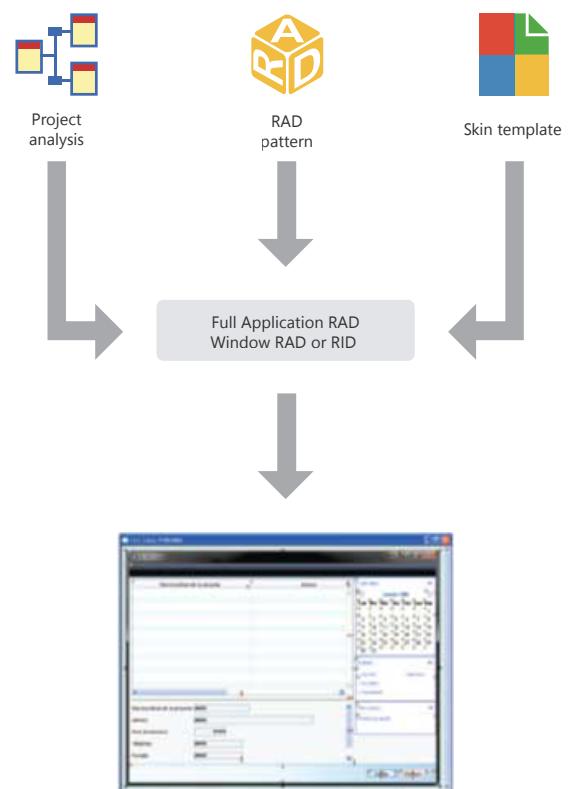
# RAD RID

The RAD (Rapid Application Development) and the RID (Rapid graphical Interface Design) are used to create windows from:

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

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

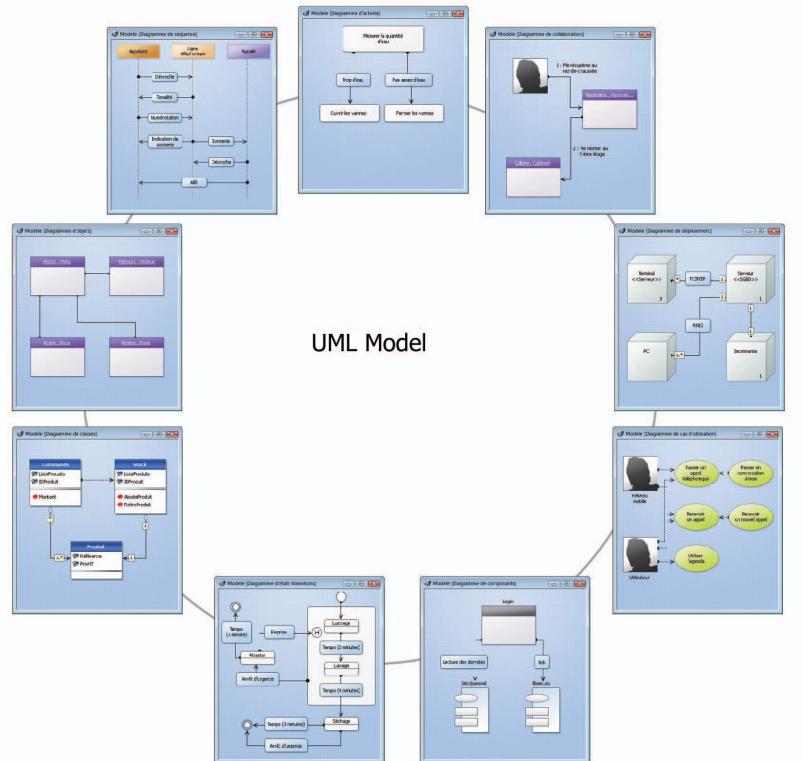
**In RID generation**, the generated windows only contain the controls linked to the analysis items. The code required for these windows to operate must be implemented by the developer. Only the code required for the additional pattern elements to operate is added. Your custom code can be typed directly.



# The UML model

WINDEV allows you to create nine types of UML models:

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

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

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

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

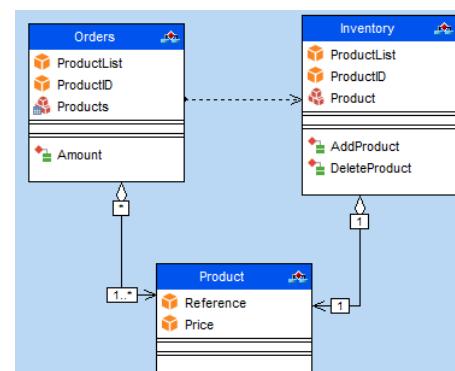
### 2 The different UML diagrams

#### 2.1 Class diagram

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

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

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



A class diagram contains the following elements:

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

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

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

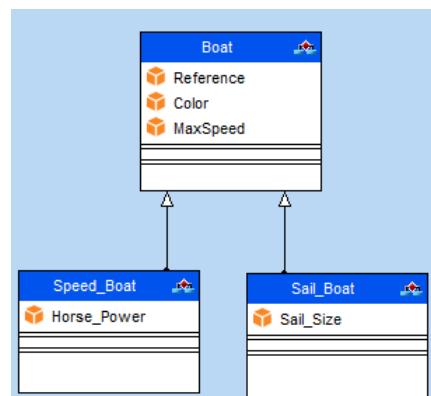
- **Public:** the element is visible to all the other classes.
- **Protected:** the element is visible to the class itself and to its sub-classes.
- **Private:** the element is visible to the class only.

**Relationship:** describes the behavior of classes between themselves. Three types of relationships are available:

• **Association:** Structural relationship between classes. For example, the Order class is linked to the Product and Customer classes. A Customer can place several Orders. An order includes several products. An order must necessarily contain at least one product.

• **Dependency:** Use relationship establishing that the instances of a class are linked to the instances of another element. For example, the Order class is using the Stock class: before adding a product 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.



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

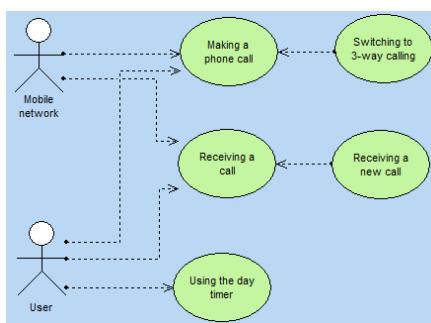
Via the generation of a class diagram, you have the ability to create the structure of WINDEV classes used in your application.

## 2.2 Use case diagram

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

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

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



A use case diagram contains the following elements:

- **Actor:** represents the role of application users. For example, a person who works in a bank will be the *loan manager*. If this person has an account in this bank, he will also play the role of *Customer*.

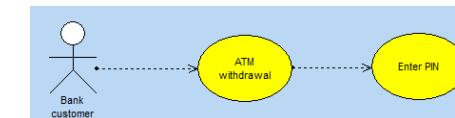
- **Use case:** describes a sequence of actions run by the application. For example, *Placing an order*, *Typing an invoice*, *Creating a new customer form*, ...

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 use cases. Three types of relationships are available:

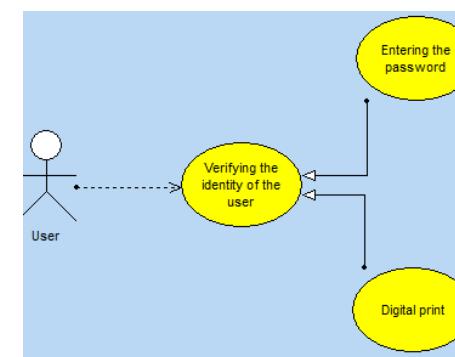
- **Association:** Structural relationship between two linked elements.
- **Dependency:** Relationship establishing that an element is using another one. For example, a bank customer can get cash from an ATM. In this case, the *Get Cash* action depends on the *Customer*.

In order to get cash, the Customer must enter his PIN number. In this case, the *Get Cash* action depends on the *Password Input*.



- **Generalization:** Relationship used to organize the elements according to a hierarchy. 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.



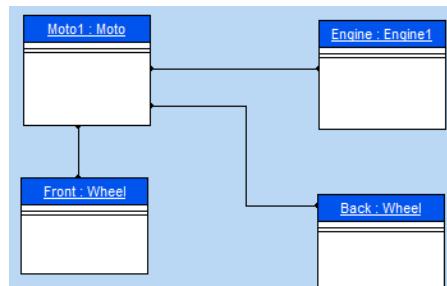
- **Package:** divides and organizes the diagram representation (like the directories organize the files). Each package can contain actors and use cases.

## 2.3 Object diagram

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

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

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



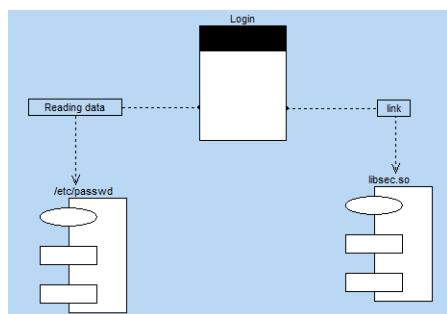
An object diagram includes the following elements:

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

## 2.4 Component diagram

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

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



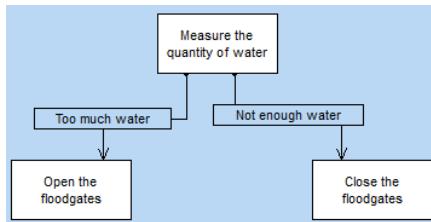
A component diagram includes the following elements:

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

## 2.5 Activity diagram

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

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



An activity diagram includes the following elements:

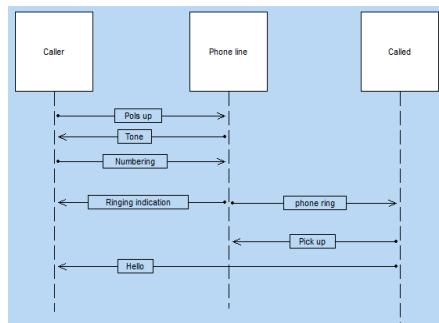
- **activity**: represents a specific step in the execution of a mechanism. For example: "Print an estimate", "Open the window", ...
- **synchronization bar**: used to synchronize the different activities:
  - by indicating the activities that must be performed before a given activity. For example: "Press the clutch" and "Change gear" before "Releasing the clutch".
  - by indicating the activities that must be performed in parallel.

- **object**: used to link activities to the object that performs these activities. For example, the "Order" and "Pay" activities are attached to the "Customer" object; the "Teach" and "Check knowledge" activities are attached to the "Teacher" object.
- **send signal**: represents the sending of a signal toward an object.
- **wait for signal**: represents the wait for a signal coming from an object.
- **transition**: represents the move from an ended activity to another one. For example: "Too much water", "Enough money", ...

## 2.6 Sequence diagram

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

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



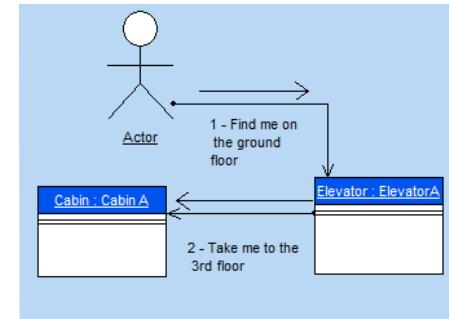
A sequence diagram includes the following elements:

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

## 2.7 Collaboration diagram

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

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



A collaboration diagram includes the following elements:

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

## 2.8 State-transition diagram

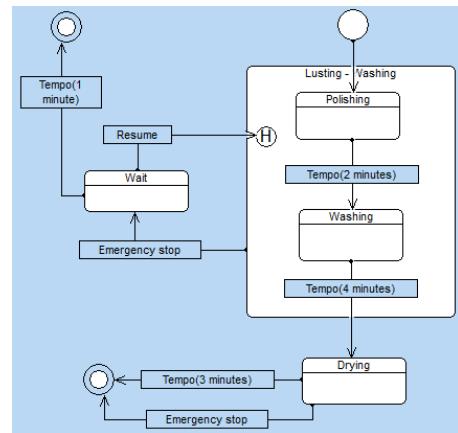
A state-transition diagram presents a sequence of states that an object goes through during its lifecycle. It is used to describe the changes of states for an object or for a component.

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

A transition is triggered:

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

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



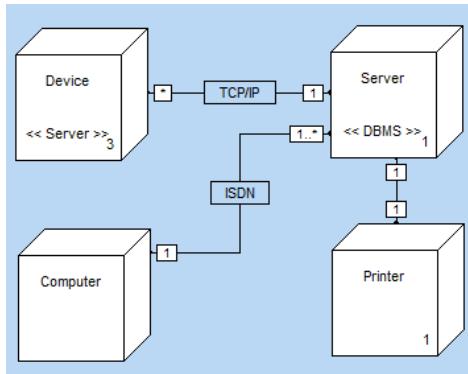
A state-transition diagram includes the following elements:

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

## 2.9 Deployment diagram

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

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



A deployment diagram includes the following elements:

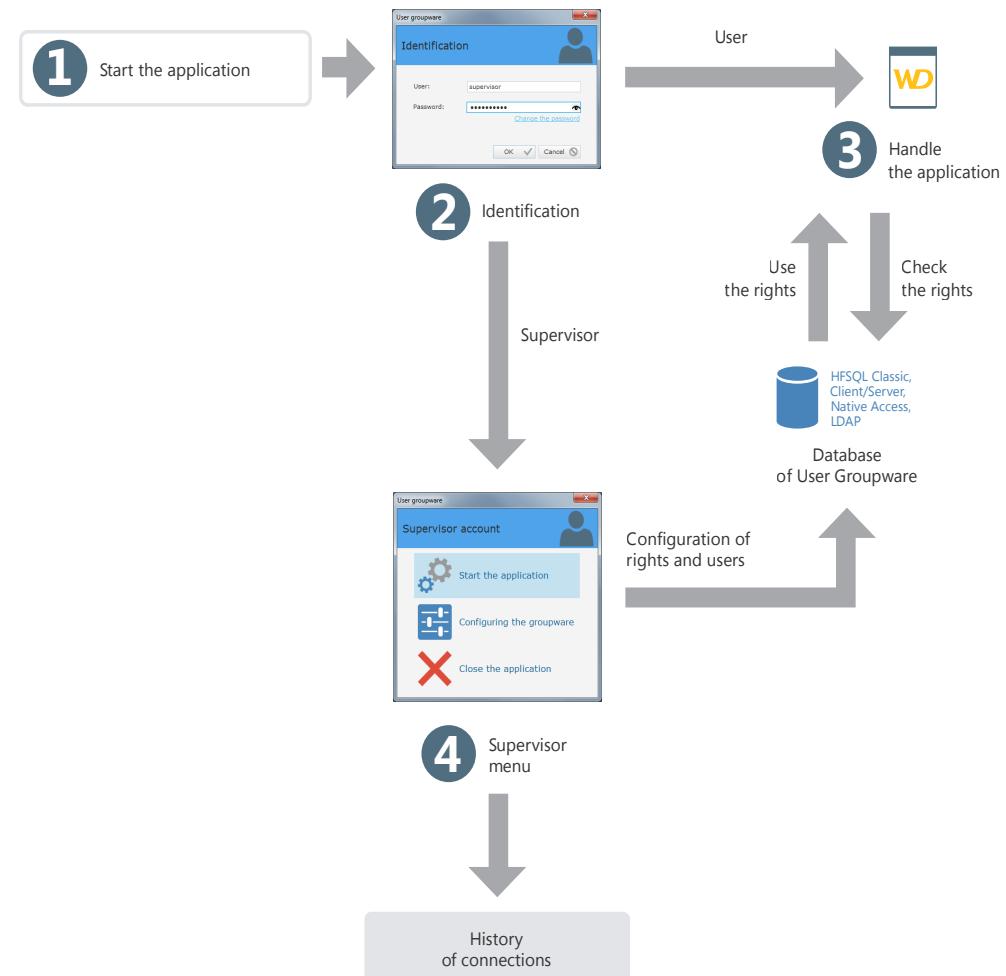
- **node class:** represents a class of hardware resources. For example: a server, a PC, a printer, ...
- **node instance:** represents a hardware resource. For example: the server #3, the printer #7, ...
- **connection:** describes the communication support between two nodes. For example: RNIS or TCP/IP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

# User groupware

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

Two types of users are taken into account:

- simple user, who is using the application directly.
- supervisor, who can configure the users and their rights.



# The user groupware in practice

## 1 Overview

An application must define the role of the different contributors. It is often necessary to define several access levels according to the user.

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

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

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

The accesses differ according to the user. Some examples:

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

To manage these access levels in your WINDEV applications, all you have to do is include the user groupware in your application. 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 starting the application, the manager will be able to create users (identified by their login and password) and to give them access to some application features.

## 2 How does the user groupware operate?

An application that uses the user groupware includes two levels:

- the user level
- the supervisor level

### 2.1 The user level

The user connects to the application via a login window and he can access the features that have been allowed for him.

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 window and he can access an advanced menu allowing him to configure the users and their rights, or to start the application.

The configuration window allows the supervisor to:

- start the application.
- configure the groupware.

Configuring the groupware consists in creating the users and groups, associating the users with the groups and managing the rights granted to each user (or group) on each application window. 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 application that is using the user groupware", page 134 for more details.

**Note:** The information regarding the users and their rights is stored in data files in HFSQL format (Classic or Client/Server).

## 3 Implementing the user groupware

### 3.1 Adding the user groupware into an application

To implement the user groupware in a WINDEV application, a single option is enough: on the "Project" pane, in the "Project" group, click "User Groupware". The window for configuring the user groupware is used to define:

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

#### Mode for integrating the User Groupware

Several modes can be used to integrate the user groupware:



- **Automatic user groupware:** the entire programming of user groupware is automatically included in your application. No customization is allowed. The skin template of project is automatically applied to all windows 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 windows required to manage the user groupware (login and user management) are automatically added to your project.** This allows you to customize all the windows used by the user groupware.

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

### Starting the User Groupware

Two modes can be used to start the user groupware:



- **Auto run:** The login window of user groupware is started before any other application window. The initialization code of project is run when the login window is opened and validated.

- **Manual start:** The login window will be opened only if `gpwOpen` is used. This option is used to run the initialization code of project before opening the login window. You have the ability to display a window requesting the runtime language of application for example.

### Automatic login in test mode

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

### Data files of User Groupware

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

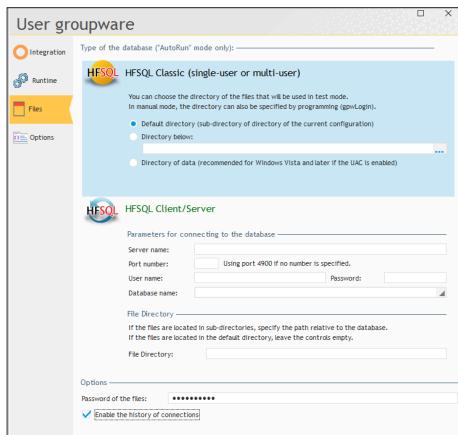
The user groupware can be used:

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

### The password of data files

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

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



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

See the online help for more details.

### Identification

The user groupware proposes several modes for identifying users:

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

### Identification managed by the user groupware

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

### Identification from an LDAP directory

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

Two modes are available:

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

See the online help for more details.

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



### Windows authentication

In this case, the login and password correspond to the ones defined for using Windows.

Note: You also have the ability to use a unique authentication. In this case, the login and password will be requested once only for each session.

### 3.2 Running the application test

When running the test of an application that is managing the user groupware, the first window displayed is the login window (regardless of the first window defined in your application).

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

- Name: SUPERVISOR
- Password: SUPERVISOR

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

#### Notes:

- If you do not want the first window of your application to be the login window, select "Manual start" in the "Execution" tab of the options of User Groupware. All you have to do is call `gpwOpen` to open the login window.
- The windows for managing the user groupware are run before the initialization code of the project.
- The window defined as first window of your application in the project will be opened after the login window (if the user is not the supervisor).
- **To avoid starting the user groupware when running the application tests,** select "No user groupware" in the "Integration" tab of the options of user groupware.

If the user groupware is re-enabled later, the data files previously used by the user groupware will not be erased.

### 3.3 Creating the WINDEV executable

When creating the executable that is managing the user groupware ("Generate" on the "Home" pane), a screen specific to the user groupware is displayed. This allows you to define the location of the data files of user groupware.

### 3.4 Installing an application that is using the user groupware

An application that is using the user groupware is installed like any other application: on the "Project" pane, click "Setup procedure".

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 be able to login during the first application start.

**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 windows in order to get an optimized configuration.

## 4 Configuring the application that is using the user groupware

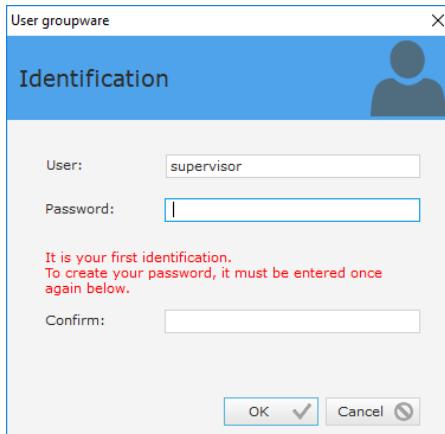
When the application is deployed, the user groupware is configured by the supervisor of the application.

This setting consists in creating the users and groups of users, and in granting them specific rights for each control found in each application window. These rights are used to gray controls, groups of controls or menu options, and to make them visible or invisible.

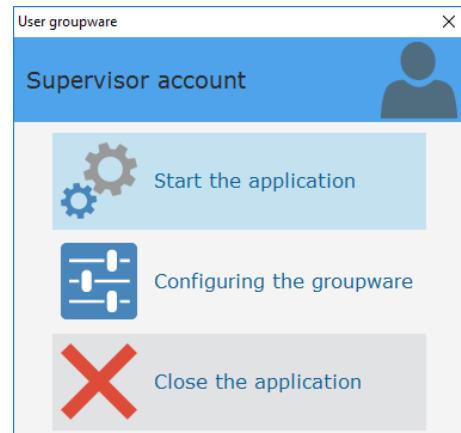
### To configure the user groupware, you must:

1. Start the application (executable or test from WINDEV) and connect as supervisor:

- Name: SUPERVISOR
- Password: SUPERVISOR



2. Select "Configuring the groupware".



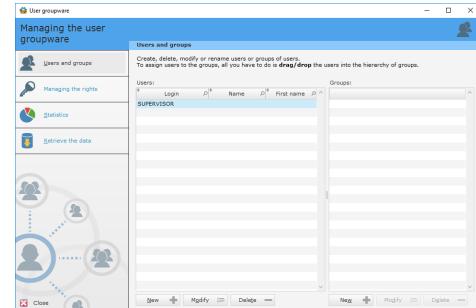
3. The window for managing the user groupware is displayed. This window is used to:

- manage the users and groups
- manage the rights
- display the statistics
- migrate the data coming from a user groupware version 17 and earlier.

### 4.1 Managing the users

The user management consists in:

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



To create a new user, you must specify:

- the last name of user.
- the first name of user.
- the user login (mandatory). This login corresponds to the identifier of the user when he connects to the application.
- the user 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 application.

You also have the ability to modify or delete a user. When deleting a user, you have the ability to delete the entire user or to delete his rights only.

**The users can be organized in groups.**

To associate a user with a group, all you have to do is Drag and Drop the user to the selected group.

A user can be associated with several groups.

### Notes:

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

### 4.2 Managing the rights

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

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

The elements managed in the windows, internal windows and window 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.

- **Inactive:** the element is displayed but no input can be performed.

- **Grayed:** the element displayed but it is grayed. No input can be performed.

- **Invisible:** the element is not displayed.

### 4.3 Displaying the statistics

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

### 4.4 Migrating data

From version 18, 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.

## 5 Tips for developing an application that supports the user groupware

### 5.1 Using groups of controls

To simplify the configuration of user groupware according to the users, the controls should be organized in groups of controls.

In your windows, you have the ability to create groups of controls corresponding to the controls that must be displayed for a 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 application 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 windows:

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

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

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

# Multilingual applications

## 5.3 Defining rights

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

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

A multilingual application is an application that can be distributed in several languages. WINDEV takes into account the different languages of the application during the entire development of an application.

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



# Multilingual applications in practice

## 1 Overview

A multilingual application is an application that can be distributed in several languages. The different application languages will be taken into account during the different development steps.

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

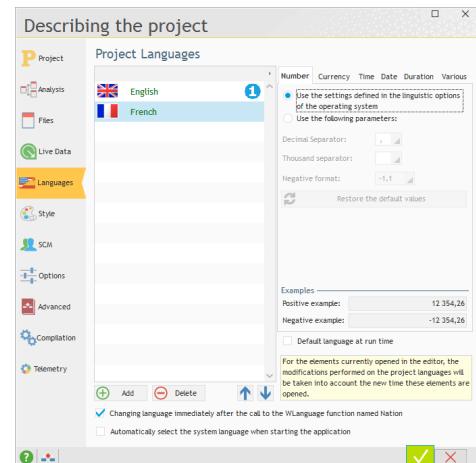
1. Choosing the languages supported by the project and the analysis.
2. Entering the different project elements (window, code, ...) in the different project languages.
3. Defining the project language by programming.
4. Managing specific character sets in the data files.
5. Creating the executable and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



#### Notes:

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

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

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

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

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

#### Linguistic options

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

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

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

**At run time**, when an edit control or a table column includes 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).

#### Executable program

When creating the executable program, you have the ability to specify the languages that will be taken into account by the executable. You can for example create a different executable program for each language supported by the project.

This feature is used to reduce the executable size.

### 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 in the description window of the analysis.

The different languages configured in the analysis

## 3 Translating the interface into several languages

When the different languages supported by the application have been selected, the information displayed by the application must be translated into these different languages.

To translate the interface into several languages, several elements must be taken into account:

- The selected translation mode.
- The supported languages (languages that use a specific character set or not)
- The use of specific windows (user groupware, automatic management of HFSQL errors, print preview)
- The messages displayed by programming.

will be proposed:

- when configuring the shared information of items. The description of the controls linked to items (options, captions, ...) can be typed in the different languages supported by the analysis. When generating a "full application RAD" or a window RAD, this information will be automatically taken into account for all the languages common to the analysis and to the project.
- for the information printed in the program documentation (notes of the data file or item).
- for the information managed by "Reports and Queries". Indeed, if "Reports and Queries" is supplied with your WINDEV application, you have the ability to translate the name of the data file and the name of each item. Therefore, the user will be able to use the names corresponding to his own language when creating his reports and queries. This information can be typed in the "Reports and Queries" tab of the description of the data file and items.

### 2.3 Languages supported by the different project elements

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

An element can support more languages than the project (when the element is shared between several projects that support different languages for example).

### 3.1 Selected translation mode

WINDEV proposes several translation modes:

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

### Translating interfaces

For each element, different multilingual areas are displayed in the description windows of element. The multilingual areas allow you to enter the information in the different languages supported by the project.

To translate this information from WINDEV:

1. Display the WINDEV options for configuring the translation options of the software used: on the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WINDEV".
2. In the "Translation" tab, you can define:
  - the default translation tool (and the transmission mode for the text to translate and for the translated text).
  - the source and destination languages.
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 application supports languages that use specific character sets (Greek, Russian, ...), you must type the translation of the different messages by using these character sets.

WINDEV 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 windows

#### Managing HFSQL errors

By default, the windows 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 windows in your project. The windows are supplied (with their WLanguage code) as example in \Programs\Data\Preset windows\HFSQL - Automatic help windows.
2. Customize the management of errors to use the HFSQL windows for error management (*HOnError*).
3. Translate the different messages (see previous paragraph).

#### Automatic management of errors

By default, the windows for automatic management of errors are supplied in English and in French. To translate them into another language, you must:

1. Include the default error windows in your application. These windows are supplied (as well as their WLanguage code) in \Programs\Data\Preset windows\Assertion - Error - Exception.
2. Translate the different messages (see previous paragraph).

#### User groupware

By default, the user groupware is supplied in English and in French.

To translate an application that is using the user groupware, you must choose a custom integration when configuring the user groupware in your application. Reminder: To configure the user groupware, on the "Project" pane, in the "Project" group, click "User groupware".

During the custom integration, the different elements of the user groupware will be included via an internal component.

Then, all you have to do is translate these elements when translating the project.

**Note:** If necessary, when configuring the user groupware, in the "Execution", tab, select "Manual run": this allows you to display a window for selecting the application language before starting the login window.

#### Print preview

By default, the preview window is available in English and in French.

To translate the preview window, the windows corresponding to the print preview must be included in your application. These windows are supplied (as well as their WLanguage code) in \Programs\Data\Preset windows\Preview. See the online help for more details.

Then, all you have to do is translate these windows.

#### System menu

A default popup menu is available for the edit controls and for the Table controls. This menu allows the user to easily perform several operations.

To translate these menus, you must own WDINT.

### 3.4 Translating messages found in the WLanguage code

In your code, several WLanguage functions allow you to communicate with the user via character strings. These messages must also be translated into the different languages of your application.

## 4 Choosing the language by programming

A multilingual application 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
- include an option (menu option or button for example) allowing the user to change the current application language.

To translate a character string entered in the code editor:

1. Select the character string to translate.
2. Select "Translate" from the popup menu of the selection. The window for translating messages is displayed.
3. Type the translation and validate. The code line is displayed in the code editor:



The flag indicates that translations exist for this character string. Click this flag to display the translation window.

The number (2 in our example) indicates the number of translations typed for this character string.

**Note:** To change all the messages found in your code into multilingual messages, on the "Code" pane, in the "Languages" group, expand "Translate the strings" and select "Convert the simple strings into multilingual messages".

## 5 Managing specific character sets in HFSQL data files

When an application is used to display or save data in HFSQL data files, this data is typed in the current language.

No specific management is required if the application is used in a country that is using a Latin character set.

If the application is used in a country that is using a specific character set, this character set must also be supported by the data files. Therefore, the searches, the sorts, ... will be performed according to the selected character set.

## 6 Managing the Unicode format and specific character sets

WINDEV proposes two management modes:

- the single charset mode:** In this mode, if your application manages languages that use specific character sets (Greek, Korean, ...), you have the ability to change by programming:
  - the character set used by all screen fonts (*ChangeCharset*). This character set can also be used for the data saved in the HFSQL files. See the online help for more details (keyword: Multilingual).
  - the language of keyboard handled by the application users (*ChangeKeyboard*).
- the full support of Unicode:** this mode is used to support the non-Latin character sets. In this mode, *ChangeCharset* and *ChangeKeyboard* are not required. The change of character set is automatically performed according to the language options defined in the project description. You also have the ability to mix different character sets.

The mode for managing the Unicode format is configured in the options of the current configuration ("Unicode" tab in the description window of configuration).

### Important

If you choose to switch your projects to Unicode, the ANSI/Unicode cohabitation generates some constraints: The data must be translated when switching from an encoding system to another one. Exchanging text strings outside the application must be adapted to take into account the new memory representation of strings (in Unicode, 1 character is coded on 2 bytes while it is coded on 1 byte in ANSI): call to API, read or write to the disk to exchange data, read or write to sockets containing strings, ...

If you choose to switch your projects to Unicode, a wizard measuring the impact of switching to Unicode is triggered on the project during the switch to Unicode.

# Optimizing an application

WINDEV proposes several methods for optimizing your applications:

- The Assisted Application Development (AAD)** that suggests code or interface improvements when developing the application.

**Developing the application**

Optimizing your queries, improving the anchors in your windows, ... These suggestions are made on a regular basis while you develop your applications.

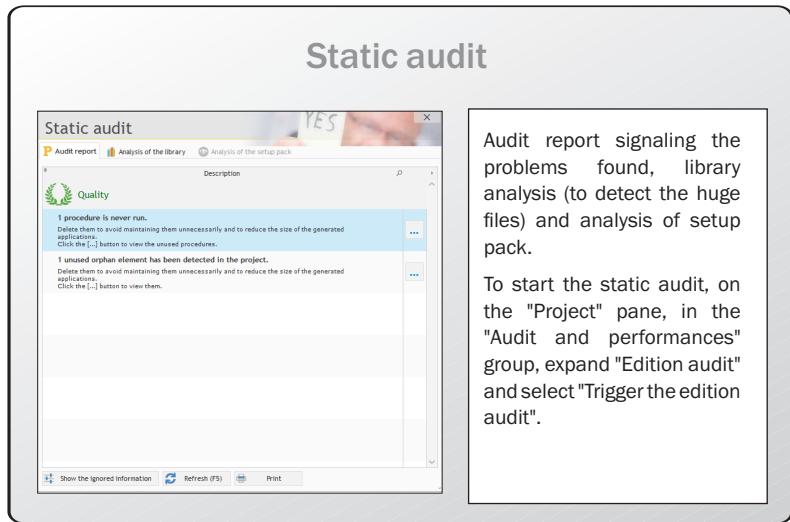
- The performance profiler** that is used to quickly view the time-consuming processes. Then, all you have to do is optimize the requested process.

**Application test**

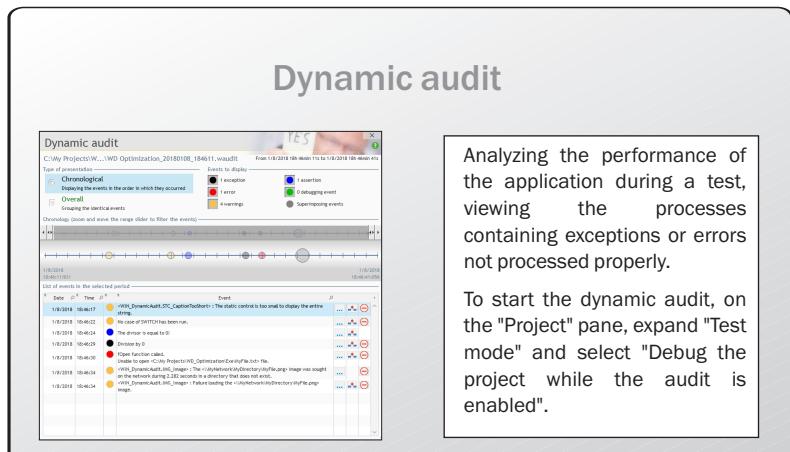
Analyzing the performance of the application while a test is running, viewing the longest processes to optimize the source code. To start the performance analyzer, on the "Project" pane, in the "Audit and performances" group, expand "Analyze the performance" and select "Analyze the performance".

Function	Internal time	No of calls
Local Procedure UpdateProductStock	2 160 ms	1
Local Procedure OrderDetailChanger	199 ms	105
Click EPL_CartesianJoiner	113 ms	1
Initiating WD Optimization	73 ms	1
Local Procedure UpdateCustomerDatabase	70 ms	136
Click EPL_PerformanceOptimizer	17 ms	1
Declaration of COV_DataProcedures	42 µs	1

- The static audit of the project that signals the potential programming problems (empty groups, orphan elements, etc.) as well as the huge files that can take a very long time to load.

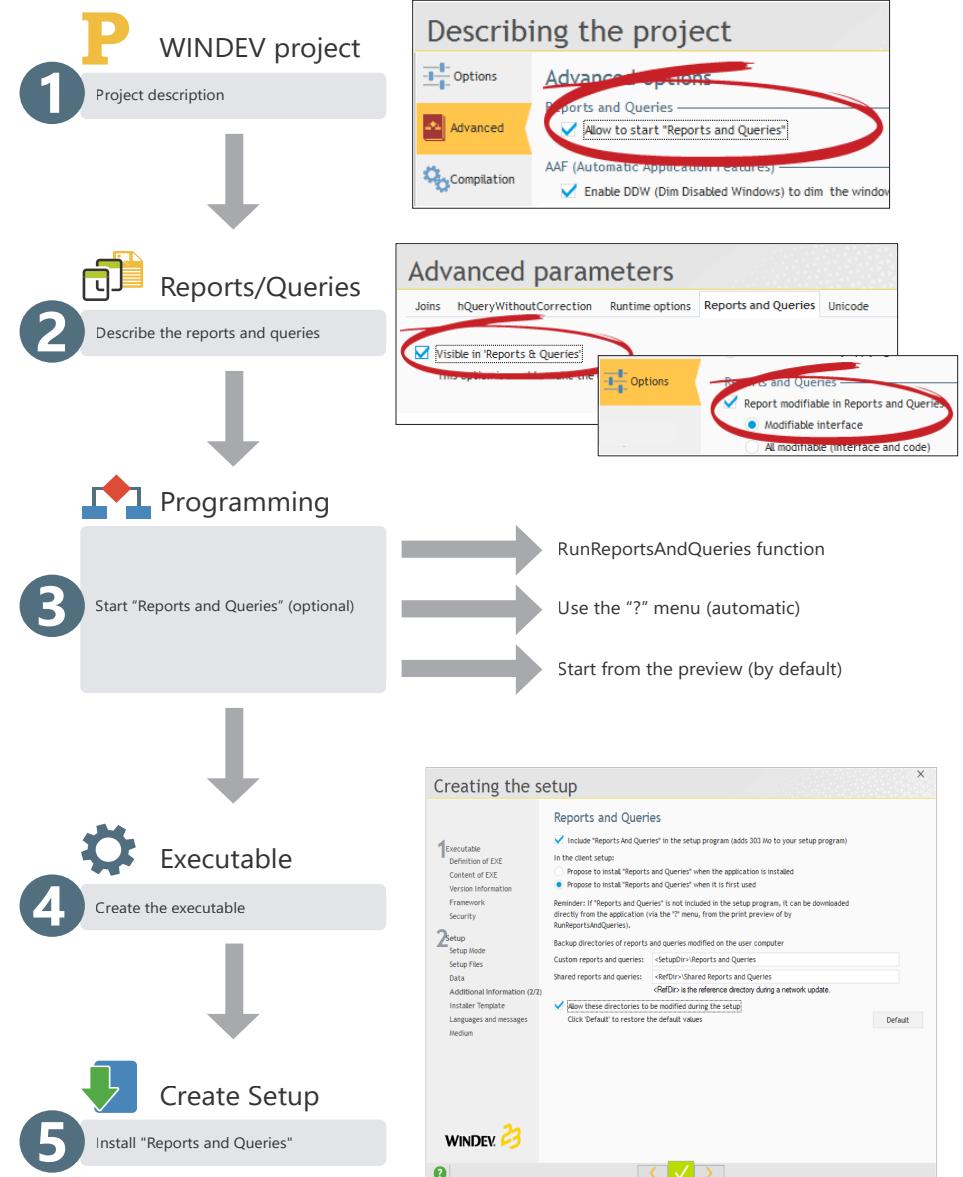


- The dynamic audit of the project that will detect the loss of resources, the unprocessed errors and the slowness of the application in its production environment.



# Distributing "Reports and Queries"

With "Reports and Queries", that can be freely distributed with your applications, the users will be able to customize the reports according to their own requirements. Only the selected reports can be customized.

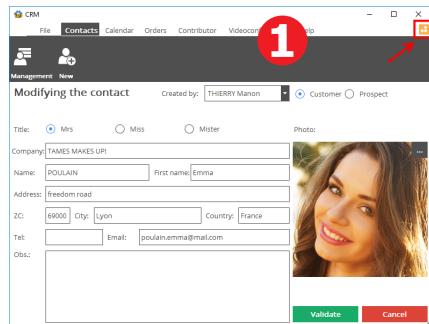


# UMC: User Macro-Code

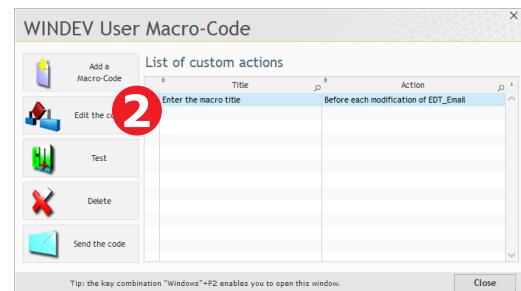
The User Macro Code (UMC) allows the end user to type his own WLanguage code and to include it in an existing application.

Let's consider the following case:

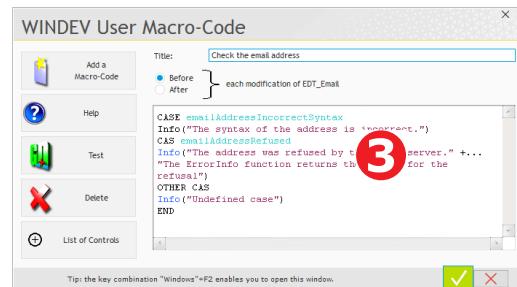
A window for entering a customer form requires the presence of an email address but it does not check its format. The user wants to check the presence of the "@" character in the address.



An icon found at the top of the window allows the user to customize the processes.



The configuration window of UMC is displayed. A new action can be added by the user for this window. In this case, the email address is checked.



The user types the WLanguage code. The F1 key is used to display the WLanguage help.

To enable this feature (which is disabled by default), simply specify that the project supports UMC:

- in the project description,
- when creating the executable.

## User Macro-Code: UMC in practice

### 1 What is the UMC?

The UMC is an option allowing the advanced users to create their own processes in the application. The end user can type WLanguage code and include it in an existing application.

The added processes can be forwarded to the developer so that they can be added into the source code of the application.

### 2 How to enable/disable the UMC?

This feature can be disabled for a window or for the entire application.

To disable the UMC on a window, check "Forbid the use of U.M.C." in the "Details" tab of the window description.

To manage the UMC on the project:

**1.** Display the description window of project: on the "Project" pane, in the "Project" group, click "Description".

**2.** Display the "Advanced" tab. By clicking the "User Macro Code" button, you have the ability to allow or forbid the input of User Macro Code for the entire application. If you authorize it, two options are

available:

- Allow the use of HFSQL commands (HReadFirst, HFilter, HAdd, ...).
- Accept the suggestions made by the users.

This second option allows the application to send an email when the user creates a macro code. The "\*.UMC" file corresponding to the macro code is sent in attachment. This allows you to easily retrieve the suggestions mode by the users and to automatically include them in your project! Indeed, WINDEV includes an options for importing "\*.MCU" files (on the "Project" pane, in the "Project" group, expand "Import" and select "User Macro-Codes...").

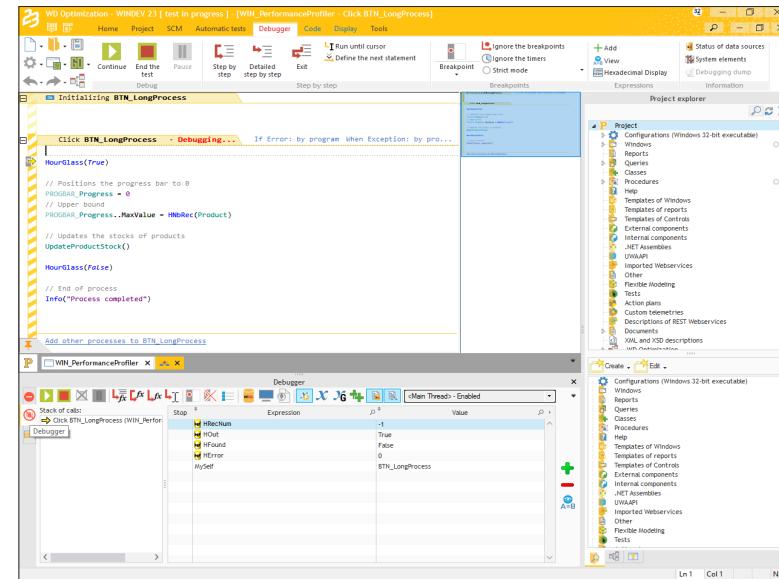
# The available tests

WINDEV proposes several methods to run the test of your application. The table below presents these different modes.

Type of test	Description
Test of entire project 	Simulates the start of the application by the executable. Used to run the test of the entire application, even if its development is not completed. As soon as a problem occurs in the project execution, the debugger can be started to identify and fix the problem.
Test of current element 	Used to run the test of this element as soon as its development is completed. The debugger can be run as soon as a problem occurs in the reports or windows.
Test of a project step by step (via the debugger)	Used to run the debugger when starting the application. This solution is used to monitor the operating mode of application via the debugger. You also have the ability to debug a specific executable, an executable that is already run or an executable run on another computer.
Unit test	Used to run the test of windows, procedures and classes found in an application at different development levels. The test scenarios are automatically generated in WLanguage from the application test or from the element test. The scenarios can be modified in the code editor.
Non-regression test	Used to check the non-regression of an element between the different application versions. To do so, select (or create) the test scenario that will be run.

# The debugger

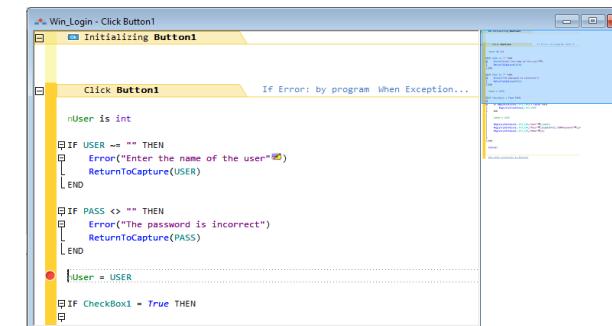
To help you optimize your code, a debugger is supplied with WINDEV. The debugger allows you to run your applications step by step, by viewing the WLanguage code run, the content of variables, ...



## Debugging the current project

### 1. Using a breakpoint, placed 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.



# Unit tests/ Automatic tests

## 2. Using the STOP keyword in the WLanguage code

The debugger will be automatically started when this code line is run.

## 3. Using the "Trace the project" option

The debugger is started when the application test is run. The "Trace the project" option is available on the "Project" pane, by expanding "Test mode".

## 4. Pressing [CTRL] + [PAUSE] while during the test of window or project

The debugger is started during the next action performed after the use of [CTRL] + [PAUSE].

## Debugging an executable

WINDEV also allows you to debug the executable associated with the current project in WINDEV. This executable can:

- be run from WINDEV, on the development computer.
- be currently run on the development computer.
- be run from WINDEV, on a remote computer.
- be currently run on a remote computer.

## Debugging a component

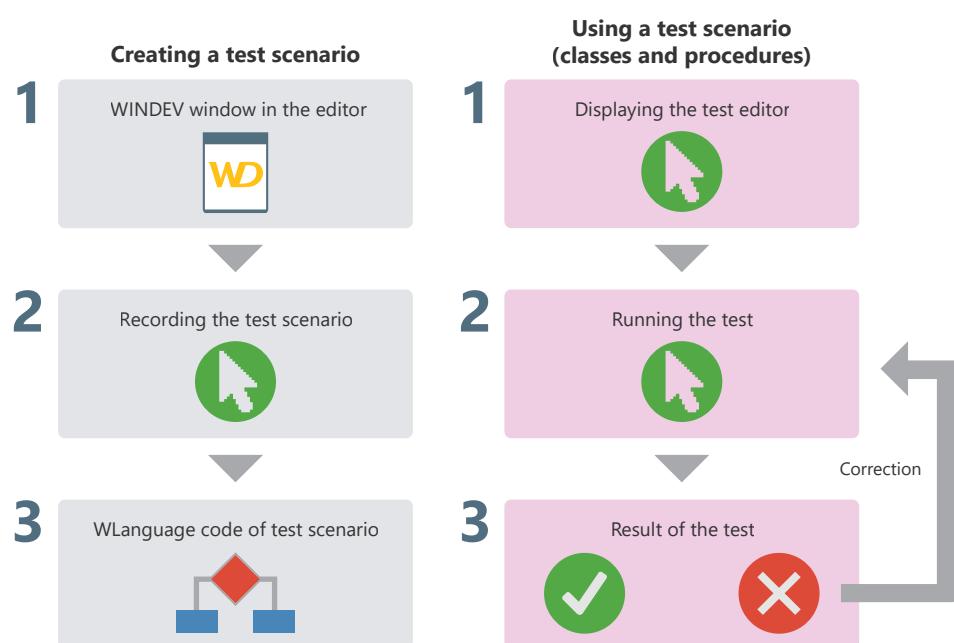
A component is debugged from the component project. You can:

- Run the component test from the component project.
- Run the component test by running another project found on the development computer. This allows you to run the component test from the project that is using it.
- Run the component test by running an executable found on the development computer. This allows you to run the component test from the executable that is using it.

The unit tests (also called automatic tests) are used to run the test of windows, procedures and classes found in an application during the entire development. To do so, select (or create) the test scenario that will be run.

These scenarios are generated in WLanguage and they can be directly modified.

These scenarios are grouped in the test editor. The test editor examines the result of unit tests and it calculates the validation rate of application.

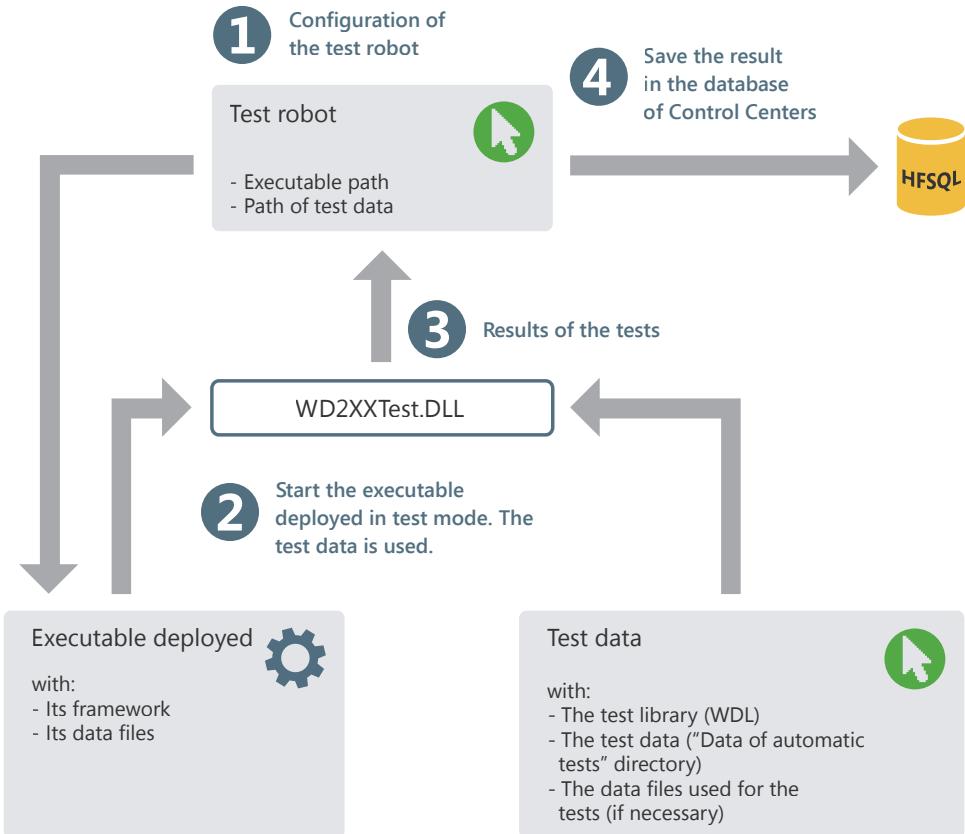


To create a unit test on the current window, on the "Automatic tests" pane, in the "Tests" group, expand "New" and select "Save a new scenario".

To create a unit test on a procedure or on a class, select this procedure or this class in the "Project explorer" pane and select "Unit tests .. Create a unit test" from the popup menu.

# Unit tests on the executable

The unit tests created with WINDEV can be run on a deployed executable, via the test robot.



## Application test in practice

### 1 Overview

WINDEV proposes several methods to run the test of your applications:

- test of entire project,
- test of single window,
- test of single query (see the "Reports and Queries" guide for more details),
- test of single report (see the "Reports and Queries" guide for more details),
- running the project step by step,
- test of performances of your application,
- regression test/automatic test.

The **test of entire project** is used to simulate the application start by the executable. This allows you to test the entire application, even if its development is not ended. As soon as a problem occurs, you have the ability to start the debugger to find out and fix the problem.

The **test of a window** is used to run the current window. This allows you to run the test of your project from a given window or to check the operating mode of a window as soon as its development is ended. Like for the project test, the debugger can be started as soon as a problem occurs.

The **test of a single query** is used to run the current query. This allows you to check the operating mode of a query as soon as its development is ended.

The **test of a single report** is used to run the current report. This allows you to check the operating mode of a report as soon as its development is ended. Like for the project test, the debugger can be started as soon as a problem occurs.

**Running the project step by step** allows you to start the debugger when launching the application. This solution allows you to monitor the progress of the application.

**The performance test** of your application is used to check and optimize the execution time of your application.

**The regression test** (or automatic test) is based on the execution of scripts. It is used to check that the existing features are still supported when running your applications.

In addition to these different methods, WINDEV also proposes to find out the "Code coverage" of application, which means the measurement of coverage for the tests run on an application. See the online help for more details.

### 2 Running the test of a WINDEV project and debugging it (development computer)

#### 2.1 Overview

Running the test of the entire WINDEV project is used to simulate the start of the application by the executable. This allows you to run the test of the entire application, even if its development is not finished yet. As soon as a problem occurs in the project execution, the debugger can be started to identify and fix the problem.

WINDEV proposes a complete set of application tests that can be run from the development computer:

- Running the test of current project and debugging it in the editor.

- Running and debugging the executable corresponding to the current project.
- Debugging an executable that is already run on the development computer.

**Note:** The test of a project can be run regardless of the current element in the editor.

#### 2.2 Running the test of current project and debugging it

To run the project test from the editor, on the "Project" pane, expand "Test mode" and select "Debug the project" ([Ctrl] + [F9]). The editor is automatically minimized and the project is run.

## 2.3 Running the executable and debugging it on the development computer

Some problems can only be reproduced when running the test of the executable in real-use conditions. This test mode allows you to debug an executable once it is installed on the development computer.

In this case, the code run comes from the libraries (and not from the code found in the elements found on the computer). Therefore, the code of the current project and the code of the executable can become out-of-sync: the out-of-sync code is highlighted in pink during the debug operation.

This test mode is used to run a specific executable installed on the development computer and to debug it directly.

See the online help for more details.

## 3 Running the test of a WINDEV project and debugging it on the user computer

WINDEV proposes several methods for running the test of a project and for debugging it on the development computer. However, in some cases, you may have to debug on the end-user computer (connected by network or by Internet).

From your office in London, you have the ability to debug an application that is run in Taiwan. The debug operation is performed without having to go anywhere, directly on the configuration of the user.

## 4 Running the window test

### 4.1 Running the window test from the editor

To run the window test from the editor:

1. Open the window whose test must be run.
2. Click  among the quick access buttons (or press [F9]). The editor is automatically minimized and the window is run.

## 2.4 Debugging an executable that is already run on the development computer

Some problems can only be reproduced when running the test of the executable in real-use conditions.

This test mode allows you to find a problem that only occurs at run time (and not in test mode).

The principle is straightforward:

- The developer starts the executable on his computer and directly goes to the requested location in the application.
- In WINDEV, the developer opens the project corresponding to the executable and asks to be attached to the executable run.
- The debugger in WINDEV is used to monitor the program execution step by step.

In this case, the code run comes from the libraries (and not from the code found in the elements found on the computer). Therefore, the code of the current project and the code of the executable can become out-of-sync: the out-of-sync code is highlighted in pink during the debug operation.

See the online help for more details.

Two features are available:

- Running and debugging the executable on a remote computer
- Debugging an application that is currently run on a remote computer.

For these two features, a specific configuration is required for the remote computer.

**Note:** to remotely debug a WINDEV application, the WINDEV application must include the WD230CPL.DLL library in its framework.

See the online help for more details.

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

## 4.2 Stopping the test of a window

Several methods can be used to stop the test:

- 1st method:  
Close the application whose test is currently run. WINDEV displays the editor that was used at the beginning of test.

- 2nd method:  
• Go back to the editor with the taskbar or with [Alt] + [Tab].  
• Confirm that the test must be stopped. WINDEV displays the editor that was used at the beginning of test.

## 5 Tracing a project

### 5.1 Principles for debugging

Debugging an application consists in:

- checking the operating mode of a process,
- understanding the operating mode of an existing program,
- checking the value of variables,
- checking the operating mode of special cases in an application.

The debugger is used to perform these operations.

**Note:** WINDEV also includes several trace tools (trace window, information box, ...). See "Debugging without debugger", page 155 for more details.

### 5.2 Overview of debugger

The debugger is used to trace the WLanguage programs in order to help you improve these programs.

The source code run is viewed on the screen. The processes run are sorted in hierarchical order in the "Debugger" pane.

The value of variables can be viewed:

- individually in the rollover tooltip of each variable.
- in the "Debugger" pane.

### 5.3 Features of debugger

The debugger allows you to:

- find out the call stack,
- view the content of variables or expressions,
- run the code step by step with ability to skip blocks,
- use conditional breakpoints,
- modify the code while continuing the execution,
- ...

### 5.4 Debugging without debugger

In some cases, running a program with or without debugger may be different.

Indeed, the debugger introduces pauses in the process execution during which several tasks are performed by WINDEV.

Therefore, the debugger cannot be used in a procedure called by timer or in the scrollbar code.

**Note:** See the online help for more details about the debugger limits.

To debug these applications, you may want to follow the evolution of a value, how different procedures are called, ...

This information can be:

- displayed on the screen.
- stored in a trace file.

**Caution:** If the information is displayed on the screen, it must be displayed during the tests of the application only.

#### Displaying information

Two tools can be used to display the information:

- **the information boxes:** *Info* function.  
Caution: Displaying an information box is a locking operation.
- **the trace window:** *Trace* function.  
The trace window is displayed in the top left corner of the screen, without interrupting the program execution.

#### Managing the display of debug information

Displaying the debug information on the screen is useful in test mode only.

Any unsuitable display must be removed before distributing an application.

To avoid any oversight, we advise you to manage the display of debug information via a global procedure. For example:

```
PROCEDURE MyTrace(StringToTrace)
IF InTestMode()=True THEN
    Trace(StringToTrace)
END
```

In this code, depending on the result of **InTestMode**, the trace window appears during the application test only.

Such procedure allows you to leave the call to the trace windows in the code of the application without any risk of displaying it on the end-user computers.

The call to the trace procedure is similar to the use of **Trace**:

```
MyTrace("Customer: " + ...
        Customer.CustomerNum)
```

### Creating a trace file

During long processes (batch processes, ...), to check the operating mode of the program, you must keep a physical trace of the processes run (a text file for example).

The following procedure is used to manage the display of the trace:

- on the screen (/DEBUG parameter in command line).
- in a text file (default mode).

```
PROCEDURE MyTrace(StringToTrace)
FilePath is string
FilePath = ...
    fDataDirUser() +...
    ProjectInfo(piProjectName) +
    ".txt"
File is int
DebugMode is boolean = False
IF Position(CommandLine(), ...
    "/DEBUG") > 0 THEN
    DebugMode = True
END
```

```
IF DebugMode = True THEN
    Trace(StringToTrace)
ELSE
    File = fOpen(...,
        FilePath, ...
        foCreateIfNotExist +...
        foWrite + foAdd)
    IF File <> -1 THEN
        DateTimeTrace is DateTime
        DateTrace is Date
        TimeTrace is Time
        DateTimeTrace = SysDateTime()
        DateTrace = MyDate..Date
        TimeTrace = MyDate..Time
        fWriteLine(File, ...)
        DateToString(DateTrace) +...
        " - " + TimeToString(TimeTrace))
        fWriteLine(File, StringToTrace)
        fWriteLine(File, " ")
        fclose(File)
    END
END
```

#### Notes:

- The trace file is created by default in the data directory of user. This file is named like the project. The information is completed by the date and time of each "Trace". This allows you to detect a potential problem during the process.
- Example of content of trace file:

```
01/12/2015 - 10:53:25:20
Customer name: Montgomery
```

## 6 Performance test

### 6.1 Overview

The performance profiler is used to check and optimize the execution time of your application.

The principle is straightforward:

You run the test of your application. During this test, the performance profiler keeps track of all the actions performed and the corresponding processes run.

At the end of the test, the performance profiler displays:

- the 10 most time consuming operations
- all the actions performed in the application 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 application.

To go back to the editor, all you have to do is close your application.

Then, the performance profiler displays the result of the analysis.

**Note:** We recommend that you use the performance profiler to optimize your application (before it is distributed for example).

### 6.3 Reading the result of the 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 the main processes.

- the "Details" tab presents all the processes run during the test of the application (from the slowest one to the fastest one).

- the "Calls" tab is used to view the details of the operations performed in a process.

The following information is displayed for each process:

Function	Function, process or procedure run.
Total Time	Execution time of function.
Nb of calls	Number of calls made to the function (procedure or process).
Time 1 call	Execution time of a call to the function (procedure or process).
% code	Percentage of code run outside the call to a WLanguage function or outside the call to a custom function or procedure.
Parent	Process that called the function.

#### Notes:

- The "Full execution" caption represents the total amount of time for running the application test with the performance profiler.

- The "Total for XXX Window" caption represents the total amount of time for running the XXX window (from its opening to its closing).

### 6.4 Choosing a process to optimize

The process to optimize is chosen according to several criteria:

- **the execution time of the process.** The longest processes must necessarily be optimized.
- **the percentage of time spent in the process of the function or procedure.** The higher this percentage is, the greater the number of processes that can be optimized in the code.

**Note:** If the process corresponds to a WLanguage function, it is fairly hard to optimize it.

## 7 Regression tests

### 7.1 Overview

Several test tools are available to guarantee the quality of your applications:

- The test mode (Go on a project or Go on a window) is used to immediately check a modification in your application.
- The automatic tests are used to save a test scenario that can be easily re-played.
- WDTest that is used to create validation tests and regression tests in WINDEV.

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 creating the executable in order to check the operating mode of an application.

The following elements can be checked:

- the WINDEV windows.
- the sets of procedures.
- the classes.
- the WINDEV executables.

Each test is associated with a WLanguage code: the scenario of the test. 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 included in the executable and they are not supplied to the end users. The number of tests for an application has no incidence on the size of the application supplied to the users.

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

### 7.3 WDTest

WDTest is a tool used to record and automatically run test scripts of applications (WINDEV or not).

A test script (also called "macro") contains the different operations (mouse position, mouse click and keyboard input) performed on an application.

Once recorded, the test script can be run as many times as necessary. When a script is run, a result script is automatically generated. The differences found between the two scripts (source script and result script) are highlighted.

This feature allows you to check whether your WINDEV applications operate properly between two updates.

Therefore, WDTest is used to automate the test procedures of your applications.

Important: To run a test successfully, the same configuration must be used:

- on the computer where the test is created.
- on the computer where the test is run.

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

PART 5

Setup

# The WINDEV framework

The WINDEV framework is the set of libraries (.DLL files) required to run a WINDEV application.

When creating the executable, you can choose to use:

- the framework beside the executable,
- the framework included in the executable,
- the common WINDEV framework.

## Using a framework beside the executable

When using a framework beside the executable, the libraries are installed in the directory of each application. Therefore, each WINDEV application is using its own version of the libraries. You have the ability to rename the libraries when using a framework beside the executable.

### Benefits of framework beside the executable:

- Each application can exploit a different version of framework DLLs.
- The framework of a given application can be updated without impacting the other applications.
- The framework can be renamed.

## Using the framework included in the executable

When using the framework included in the executable, the executable is stand-alone. The framework DLLs are not extracted when running the executable.

### Benefits of the framework included in the executable:

- Stand-alone executable.
- No extraction of framework DLLs.

**Note:** Depending on the features used by the application, it may not be possible to use the framework included in the executable.

# The different modes for installing an application

## Using the common WINDEV framework

When using the common framework, the libraries are installed in a common directory. These libraries are shared by all the WINDEV applications installed on the same computer.

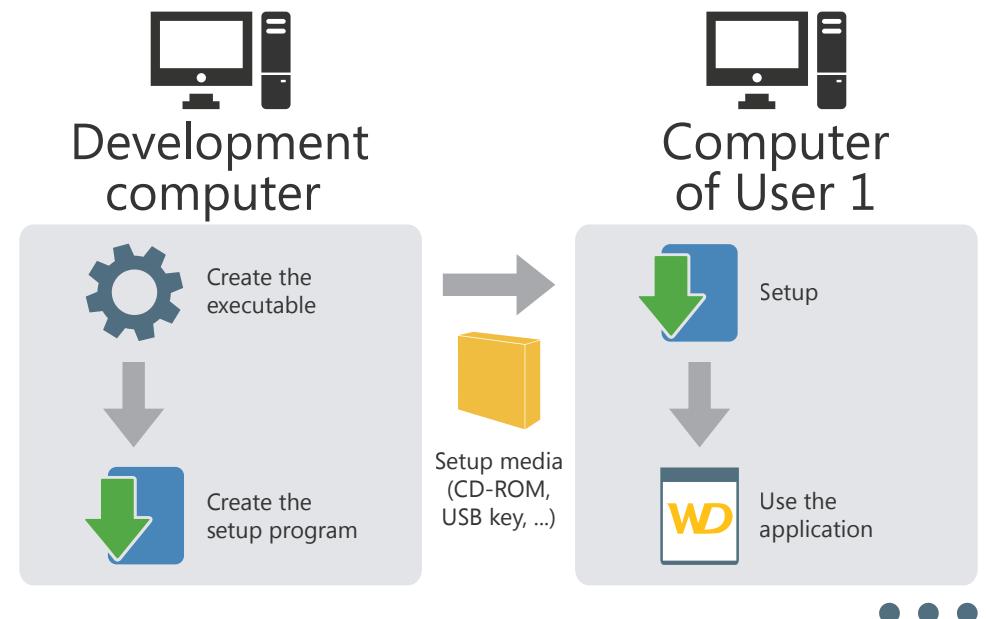
### Benefits of the common framework:

- The disk space used by the framework (about 30 MB) is shared by all applications.
- The framework is updated once only for all the applications installed.
- The common framework can be renamed.

**Note:** When using the common framework, you have the ability to replace one or more DLLs (if a specific version is required for example).

### Individual setup

The individual setup consists in creating a setup program that will be installed on each user computer via a setup media.

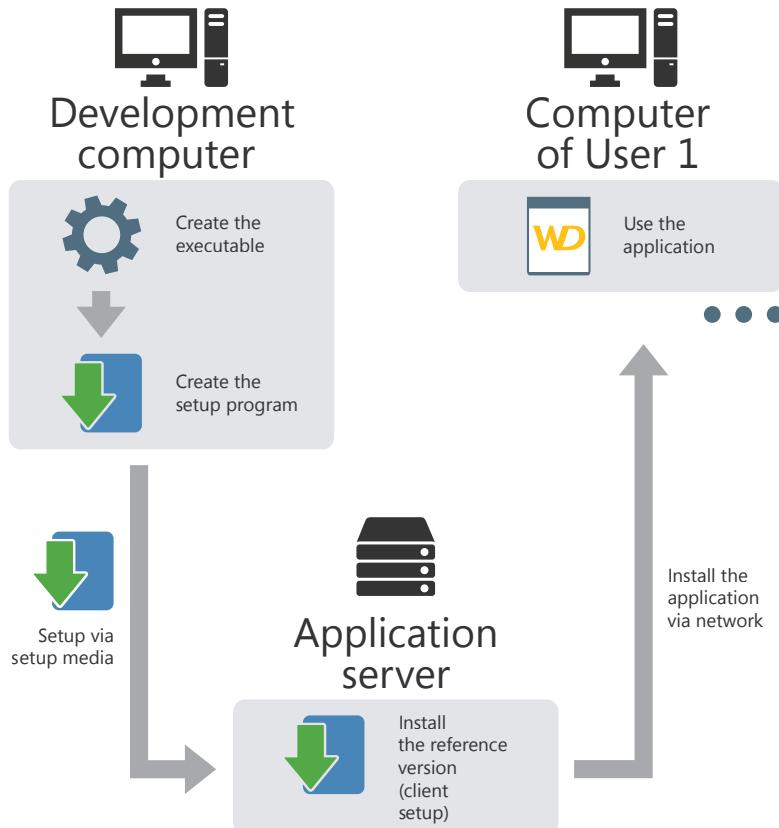


This setup mode is compatible with:

- data files in HFSQL Classic format installed locally on the user computer,
- data files in HFSQL Classic format installed in network and accessible from the user computer,
- data files in HFSQL Client/Server format installed on a server.

## Network setup (with automatic update)

The network setup consists in creating a setup program that will be installed on a server. All the user computers that want to install the application will start the application setup via the network.



If an update is available on the application server, the application update will be proposed when the application is started on the user computer.

A network setup allows for an extensive check of the user computers (versions deployed, mandatory updates, disconnection of users, etc.)

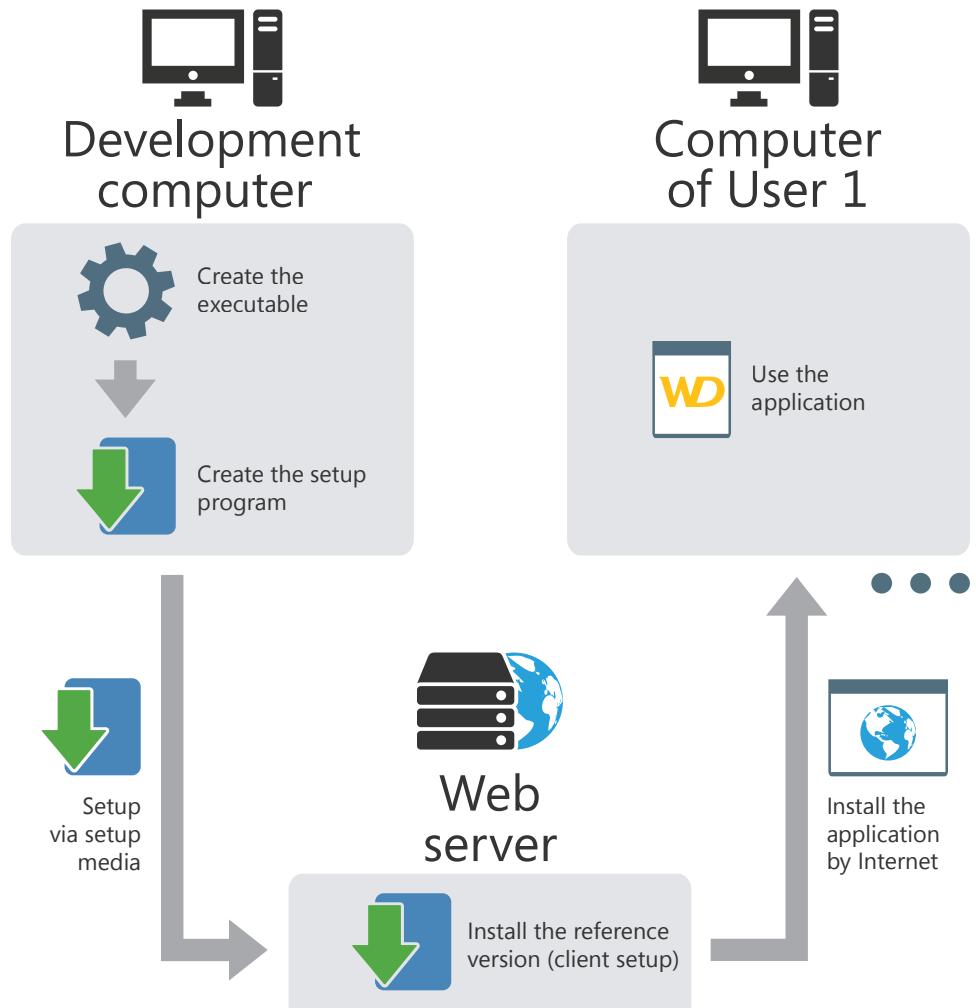
This setup mode is compatible with:

- data files in HFSQL Classic format installed locally on the user computer,
- data files in HFSQL Classic format installed in network and accessible from the user computer,
- data files in HFSQL Client/Server format installed on a server.

## HTTP setup (with automatic update)

The HTTP setup consists in creating a setup program that will be installed on a Web server.

All the user computers that want to install the application will start the application setup via an HTTP URL. If an update is available on the Web server, the application update will be proposed when the application is started on the user computer.



## Multi-site setup (with automatic update)

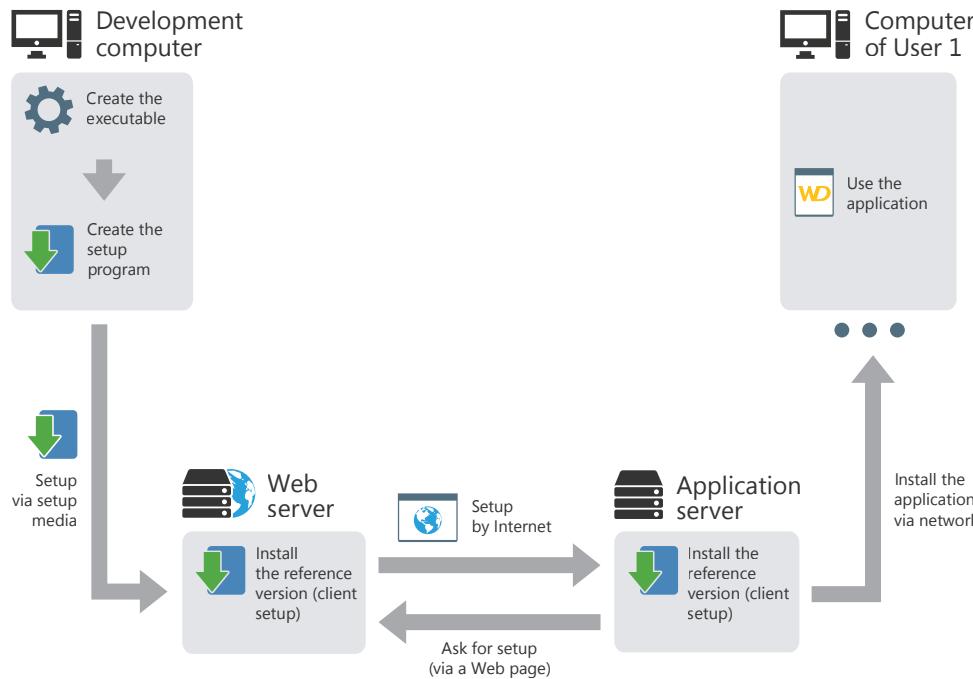
The multi-site setup allows you to install an application on several different sites (a site can represent a subsidiary for example).

The multi-site setup consists in creating a setup program that will be installed on a Web server.

All the sites that want to install the application will start the application setup via an HTTP URL. In this case, the setup program will be installed on a server of the site.

All the user computers that want to install the application will start the setup via the network.

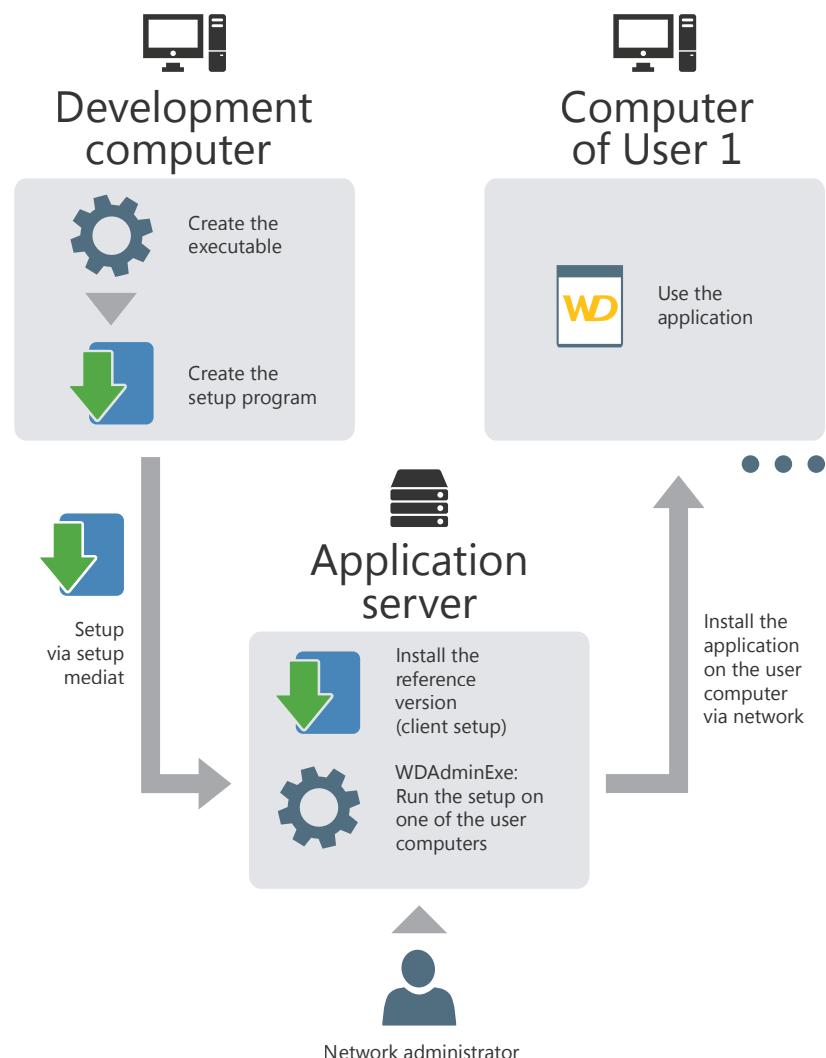
This setup combines the benefits of the network setup and HTTP setup by allowing a large-scale deployment as well as a granular control of local setups.



If an update is available on the Web server, the application is automatically updated on the application server.

## Push network setup

The network setup consists in creating a setup program that will be installed on a server. The setup will be performed from the server, by the network administrator on the user computers selected by WD2xxAdminExe.



A Push setup can be updated either in Push mode, or as part of a network setup.

# Deploying a WINDEV application

When developing an application, one of the important steps consists in deploying this application on the user computers.

WINDEV optimizes this deployment by offering a large choice:

- in the setup media.
- in the tools used to create the setup program.

- in the different setup modes (single computer or network).

This chapter briefly describes the different types of setup proposed by WINDEV and their specific features.

See the online help for more details.

## 1 Setup: Summary table

### 1.1 Setup media

Depending on the setup medium used (Internet, in-house network, ...), WINDEV proposes several modes for generating the setup.

To optimize your setup, choose the recommended generation mode.

Modes for setup generation proposed by WINDEV				
	Self-extracting executable	Directories, to the size of diskettes	Single directory	CD DVD
Internet	X (recommended)			
CD/DVD			X Autorun possible	X (recommended)
In-house network	X		X (recommended)	X

### 1.2 Tools for creating the setup program

Several tools are available for creating the setup program, for maintaining it and for customizing it.

### 1.3 The different types of setup

Wizard	WDINST, setup editor	Customizing WDSETUP
X (recommended)	X	
X	X	
	X	X (multilingual setup for example)
On the "Project" pane, in the "General" group, click "Setup procedure".	On the "Tools" pane, in the "Utilities" group, click "WDInst".	WDSETUP project available in the "Examples" sub-directory of the setup directory of WINDEV.

Two types of setup are available in WINDEV:

- **Individual setup:** recommended for the single-user applications.  
The setup program installs the application on the user computer.
- **Setup with automatic update:** recommended for the multi-user applications. The setup with automatic update is performed in two steps:

- The reference application is installed on the server. The reference application corresponds to the setup program of application on the user computers.
- The final application is installed on each user computer by running the setup program found on the server. The updates will be automatically performed when the reference application evolves on the server.

## 2 Creating the setup program

### Individual setup

Different types of setup are available:

- **Standard setup**  
The setup program is independent of the application. All the files required by the application are installed on the user computers. To take an update into account, this update must be run on each user computer.

- **Compact setup**

A standard setup is performed. However, the setup program is smaller because it does not use specific DLLs and some customization options are not available.

- **Setup via MSI package**

A standard setup that uses the features of the integrated Windows installer is performed.

#### Setup with automatic update

Different types of setup are available:

- **Setup in local area network (LAN) with "Push" possibly enabled.**

A first application setup will be performed on the network, in a shared directory. This directory will contain the setup program used by the end users and the reference version used to manage the updates.

This setup is used to:

- manage the automatic update of application installed on network.
- find out the characteristics of users connected to the application at a given time.
- configure the location of the files required by the application. Different cases can be managed, for example:

- Installing all the application files on the server.
- Installing the database on the server and installing the files required to run the application ("\*.EXE", "\*.WDL", "\*.DLL", ...) on each user computer.
- Installing part of database on the server and another part on each user computer. The files required to run the application ("\*.EXE", "\*.WDL", "\*.DLL", ...) are also installed on each user computer.
- If the "Push" is taken into account, this setup also allows you to:
  - deploy a new application on a network from a single management computer.
  - deploy an update for the application on a network from a single management computer.

- **Setup by HTTP (on a proprietary server or in the CLOUD).** A first application setup will be performed on an HTTP server. The users will connect to a Web page in order to download the setup program of application.

This setup is mainly used to:

- manage the automatic update of the application installed on an HTTP server. .
- configure the location of files required by the application. Different cases can be managed, for example:
  - Installing all the application files on the server.
  - Installing the database on the server and installing the files required to run the application ("\*.EXE", "\*.WDL", "\*.DLL", ...) on each user computer.
  - Installing part of database on the server and another part on each user computer. The files required to run the application ("\*.EXE", "\*.WDL", "\*.DLL", ...) are also installed on each user computer.
- **Multisite setup.** This setup combines a local network setup with an HTTP setup. The HTTP setup is used to update the reference setup on the local network.

## 2.1 Creating a setup program

To create a setup program allowing you to:

- **deploy a WINDEV application**, you can:
  - use the wizard for creating the setup program from WINDEV (on the "Project" pane, in the "Generation" group, click "Setup procedure"). This wizard allows you to create the different types of setup programs.
  - use the setup editor WDINST (on the "Tools" pane, in the "Utilities" group, click "WDInst"). This editor is used to customize the setup program.
- **provide other files**, you must use the wizard for creating the setup program started from the setup editor WDINST ("File .. New" from WDINST). Reminder: To start the setup editor WDINST, on the "Tools" pane, in the "Utilities" group, click "WDINST".

## 2.2 Installing a WINDEV application

### Standard, compact or MSI setup

To install your application, all you have to do is start the setup program of the application on the user computer. The files required by the application are installed in the specified setup directory.

To install an update of your application, all you have to do is run the setup program of update on the user computer. The application files modified by the update are replaced.

#### Setup with automatic update

**To install your application:**

1. Start your setup program on the server. The files required by the reference application are installed in the specified setup directory. The reference application corresponds to a client setup program.
2. Run the client setup program from each user computer. The files required by the client application are installed in the specified setup directory.

**To install an update of your application:**

1. Start your setup update on the server. The files of the reference application modified by the update are replaced. These replaced files can be saved.
2. When the application is started from a user computer, the application update is automatically proposed.

#### Setup with automatic update in PUSH mode

**To install your application in PUSH mode:**

1. Start your setup program on the server. The files required by the reference application are installed in the specified setup directory. The reference application corresponds to a client setup program.
2. The program for managing the network applications is automatically started.
3. Select the network computer on which the setup will be performed.

**To install an update of your application in PUSH mode:**

1. Start your setup update on the server. The files of the reference application modified by the update are replaced. These replaced files can be saved.
2. Start the program for managing the network applications.
3. Select the network computer on which the update will be performed.

# Monitor your applications

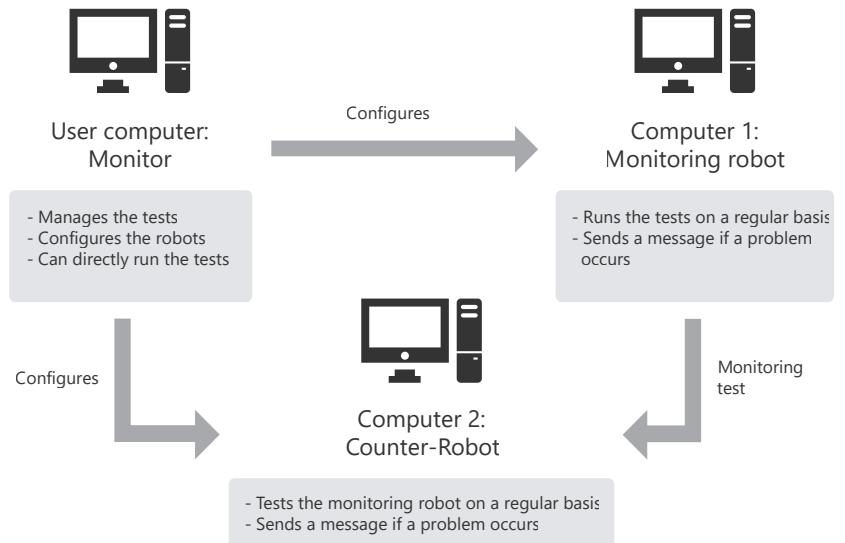
An application depends on several external parameters: Web site, HFSQL server, network, ... To optimize the management of incidents, WINDEV allows you to use a monitoring robot.

Made of three executables started on different computers, the monitoring robot is used to run different tests: checking the Internet connection, network tests, ...

If problems occur when running a test, different methods can be used by the monitoring robot to inform you:

- Message sent in the PC SOFT messaging (WDBAL).
- Email
- 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		Network test (SNMP)
	Test of FTP server		Test of news server		
	Custom test (in WLanguage)	...			
				Test of WINDEV application	

DEVELOP 10 TIMES FASTER

PART 6

## Communication

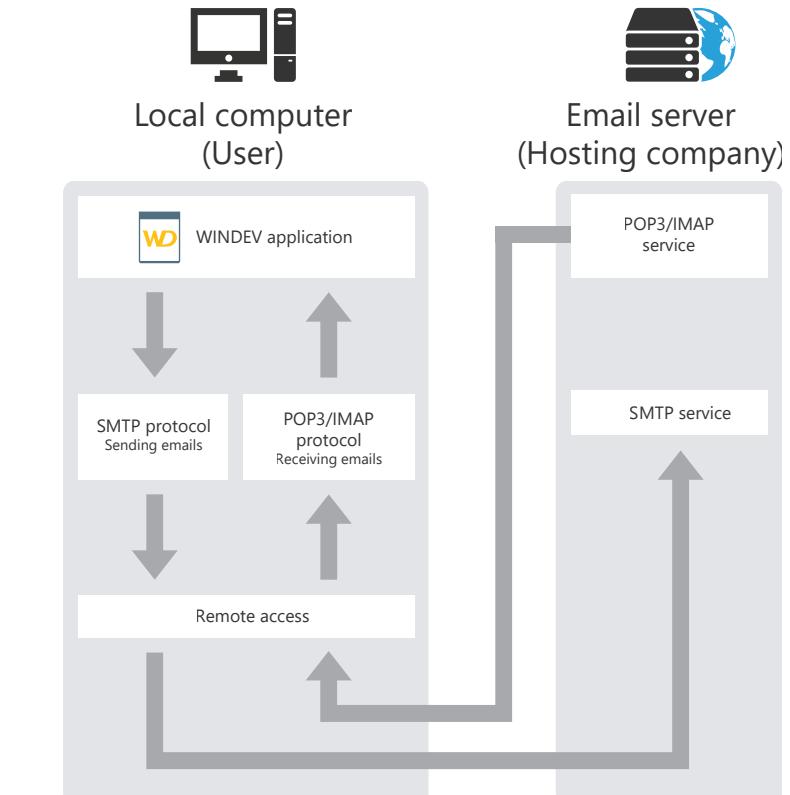
# Managing emails

WINDEV proposes several methods for managing emails:

- the POP3 or IMAP protocol and the SMTP protocol (most common method)
- the "Simple Mail API (also called SMAPI or Simple MAPI)"

## POP3 or IMAP and SMTP protocols

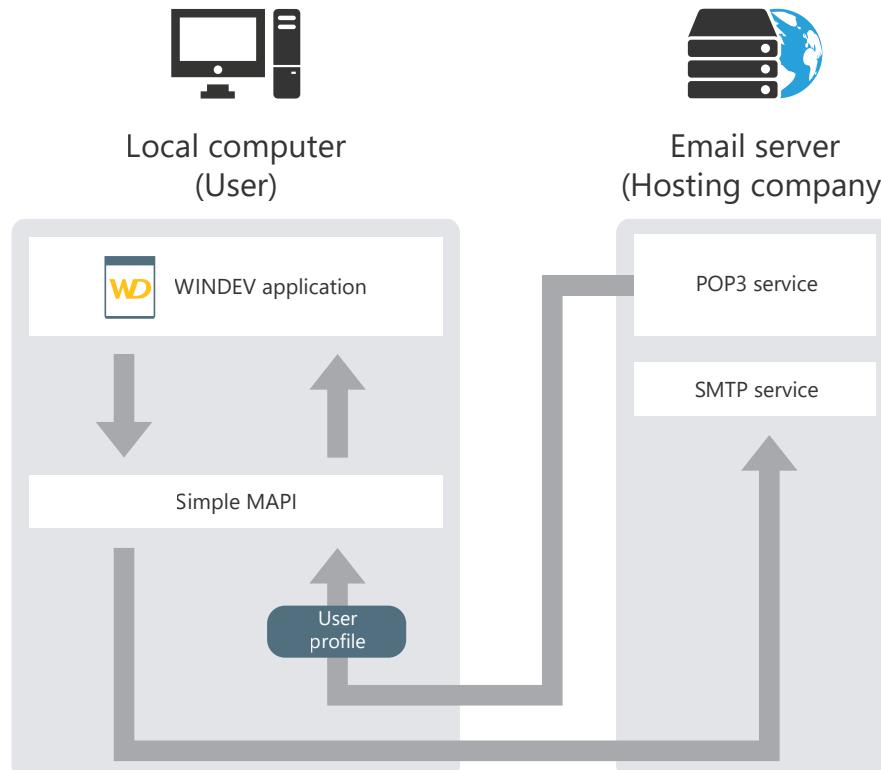
These protocols are the standard Internet protocols for email management. They are standardized by the RFC 1939, 3501 and 5321 documents. These protocols allow you to directly communicate with the server, available at your ISP.



- POP3 (Post Office Protocol) is a protocol for reading emails.
- IMAP (Internet Message Access Protocol) is a protocol for reading emails remotely.
- SMTP (Simple Mail Transfer Protocol) is a protocol for sending emails.

**"Simple Mail API (also called SMAPI or Simple MAPI)"**

This mode for email management is used by most Microsoft applications, especially MS Exchange.



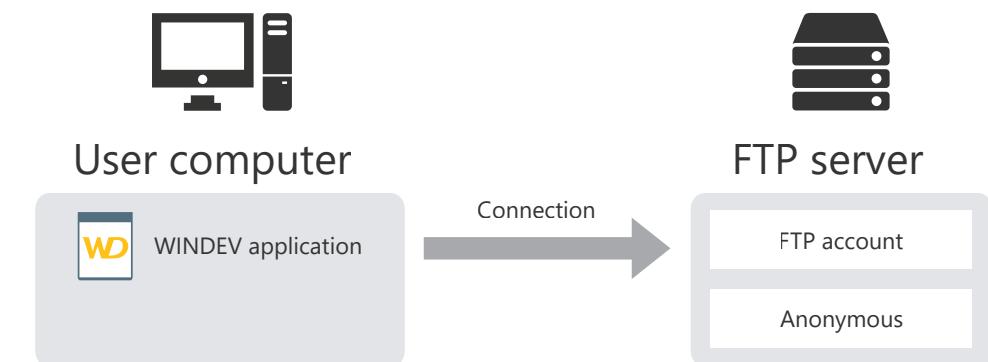
# Transferring files by FTP

The FTP (File Transfer Protocol) is a protocol used to transfer files from a site to another remote site. This protocol is used to exchange files by TCP/IP or Internet.

## Principle

To handle the files found on an FTP server from a WINDEV application, you must:

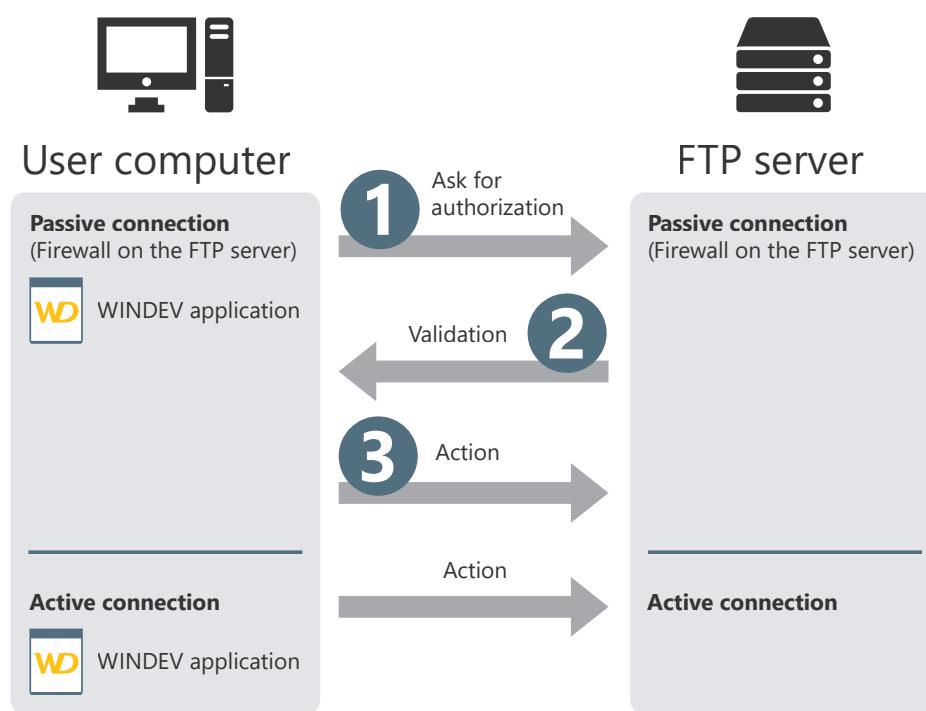
1. Connect to an FTP server (*FTPConnect*).



2. Send and retrieve files (according to the connection mode). For a passive connection, the WINDEV application must request the authorization from the FTP server before each file operation on the server. You can also:

- find out the characteristics of files found on an FTP server: attributes, size, ...
- handle the files found on an FTP server: creation, deletion, ...
- list the files of a directory found on the FTP server by running a procedure used to perform a process for each file found.

# Managing sockets



WINDEV proposes several functions used to perform an advanced management of sockets.

A socket is a communication resource used by the applications to communicate between computers regardless of the network type.

This communication mode can be used, for example, to establish a communication between computers connected by Internet.

## Different possibilities

A WINDEV application can manage the sockets according to different modes:

- **Client WINDEV application:** the application connects to any server and it exchanges data via a socket.
- **WINDEV "Simplified server" application:** the WINDEV application is a server, exchanging information via sockets with a single client computer.
- **WINDEV "Standard server" application:** the WINDEV application is a server, exchanging information via sockets with several client computers.

## Principle of client application

A client application of a socket server connects to a standard server in order to exchange information via socket.

Example: A client WINDEV application can connect to a standard news server on Internet.

### Step 1: Connecting to the server

To connect to a socket of the server, use **SocketConnect**. This function is used to perform a request for connecting to the server. The socket is identified by its port and by its address.

### Step 2: Exchanging data

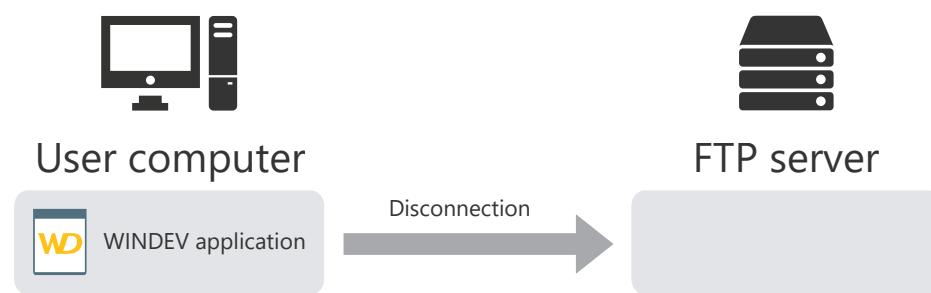
Once two computers have connected their socket, a communication channel is established between these two computers. These two computers can read and write character strings on the socket.

To read and write on the server socket, the WINDEV client application must use **SocketRead** and **SocketWrite**.

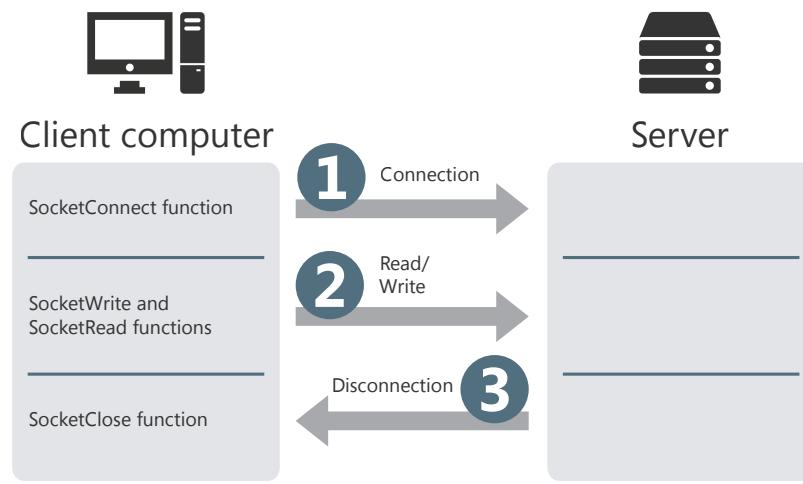
### Step 3: Ending the communication

To end the communication, all you have to do is close the socket from the client computer with **SocketClose**.

3. Disconnect from the server (**FTPDisconnect**).



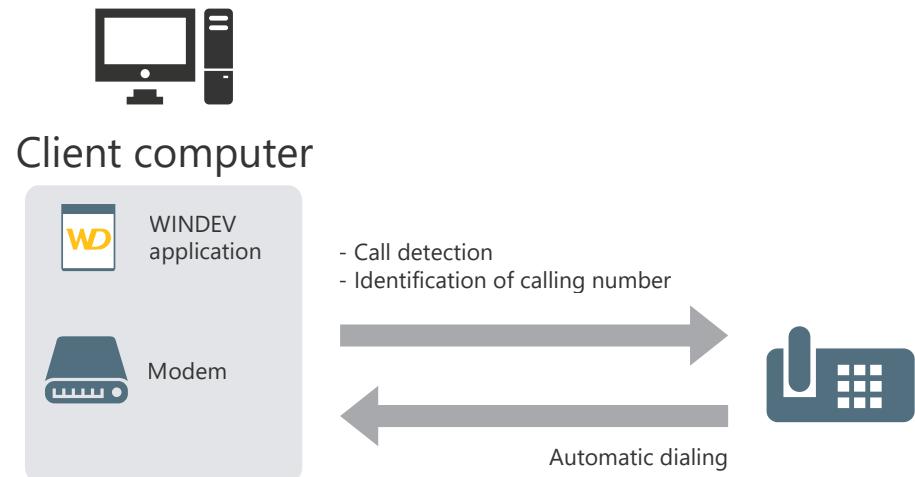
The different steps can be represented as follows:



**Note:** the SocketXXX functions can also be used to manage secure sockets according to the SSL protocol. To do so, you must use **SocketCreateSSL** and **SocketConnectSSL**.

# WINDEV and telephony

WINDEV allows you to easily manage the incoming and outgoing phone calls via the WLanguage telephony functions. These functions are used to manage a voice box, an answerphone, ... in a WINDEV application directly.



## Required configuration:

To be able to use the telephony features, you must have:

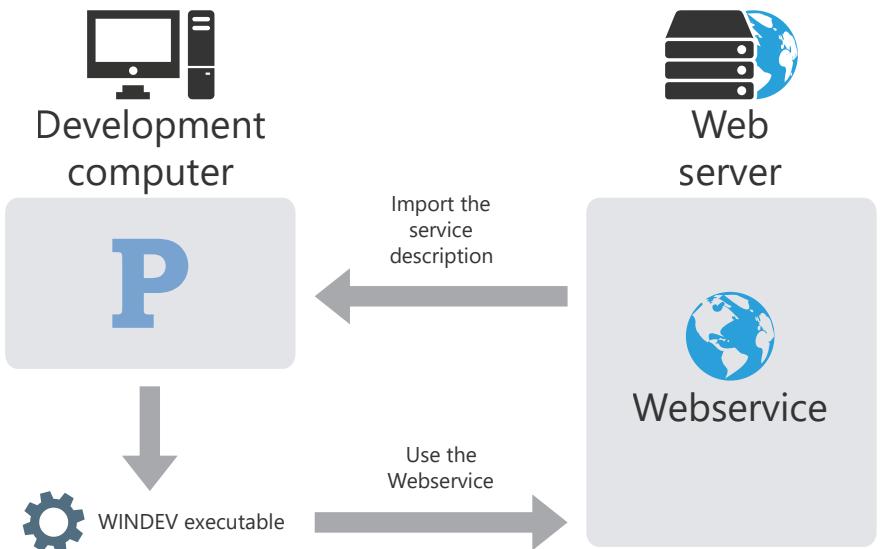
- a modem.  
To save and play recordings, this modem must include voice support.  
To identify the caller, the modem must support Caller ID.
- the Tapi 2.0 protocol.

# The Webservices

WINDEV allows you to easily use the Webservices (or XML Web services). A Webservice is an application available on the Web. WINDEV allows you to access SOAP Webservices (Simple Object Access Protocol) as well as REST Webservices (Representational State Transfer).

Any WINDEV application can use an existing Webservice.

To use a SOAP Webservice, all you have to do is know the Webservice address and import its description into the application.



To use a REST Webservice, all you have to do is find out the Webservice characteristics and use **RESTSend**.

**Note:** you also have the ability to expose one or more sets of procedures found in a WINDEV application as a SOAP or REST Webservice. To do so, you must:

- create a project configuration dedicated to this use,
- for a REST Webservice, create a description of REST Webservice,
- generate the Webservice,
- deploy it on a WEBDEV Application Server.

PART 7

## Appendices

# Components supplied with WINDEV

WINDEV 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 the component.
- an internal component.

The list of main components supplied with WINDEV is as follows:

Component name	Description
WD BGInfo	Used to write information on the background of a computer screen (IP address, name of the computer, today's date, etc.).
WD Communication	Used to establish a communication between two applications. These two applications can be: - On remote computers. The communication is performed via sockets. - On the same computer. The communication is performed via Windows messages.
WD Countries of the World	Contains the list of all countries of the world as well as their characteristics: - ISO code of each country, - list of regions, states or countries making up a country (including the time difference for each state, region or country), - time difference (GMT), - main currency of the country, - main languages, - flag of each country, - capital, - dialing code.
WD Entering emails	Used to simplify the input of emails.
WD EventLog	Used to: - List and read the events saved in the event log of Windows. - Add an event into the event log of Windows.
WD Explorer bar	Allows you to manage an address bar similar to the address bar of the explorer found in Windows Vista.

WD Extraction	Used to extract and store the content of several documents (Open Office, Word 2007, ...) in order for them to be found by the full-text search.
WD Floating Windows	Allows you to display temporary information to the user without interrupting his work.
WD Glossary	Allows you to organize and store the written expressions regularly used by a customer. This allows you to propose an access to a "Glossary" that can be easily customized by the end user. All the glossary entries can be organized in tree-like categories.
WD Graphic Popup	Several software for image editing allow you to select a shape to draw and to choose a thickness for this shape via an advanced graphic popup combo box. This type of graphic popup combo box offers a real-time preview of the shape with the selected thickness. This component allows you to include this type of interface in a WINDEV application.
WD HTML pages import	Used to import HTML pages with the WLanguage functions.
WD LastDocuments	Allows you to propose a menu option used to open the last documents opened in the application.
WD Limits	Used to limit an application by different methods: - Time limitation (the application can be used during a 30-day period) - Limitation to a single computer (the application can be run only on a single computer) - Limitation on the number of users (the application cannot be run by more than n users)
WD Managing Logs	Allows you to define, for an application, a configuration for the runtime logs: - number of log files that must be stored, - maximum size for each log file, - information that must be logged. This example simulates different processes to observe the management of log files (.wlog files).
WD Photocopier	Allows you to photocopy documents via a scanner and a printer.

WD Printout Parameters	Presents the use of iParameter. This function is used to retrieve the different settings defined for a specific printer. The printer used is defined by iConfigure(Printer Name). This component is used to modify different settings: - The paper orientation - The scale - The management of TrueType fonts - The number of copies - The management of color (color or black/white print) - The duplex print - The output style (sheets grouped by copy or by page number)
WD PrintModel	Allows you to manage the print profiles.
WD PrintSpooler	Defines the list of less used printers in order to perform one or more prints.
WD Scanner	Allows you to digitize a document via a user-friendly interface.
WD Select RTF	Used to perform all types of searches in an RTF control. You have the ability to search for: - the words in bold, - the words in italic, - the words of a specific color, - a character string containing specific words.
WD Sending faxes	Allows you to manage the faxes (operates in Windows 2000 and Windows XP only).
WD Signature	Allows you to easily include a ready-to-use 'Signature' control in your WINDEV applications. This control is very useful for the applications intended to operate on touch-screen devices, TabletPC in Windows XP, interactive devices, ...
WD Simplexe	Allows you to perform calculations from the Simplexe algorithm.
WD SystemInfo	Proposes three categories of functions used to group information about the current computer, the user and the network to which the current computer is connected.

# Examples supplied with WINDEV

The examples supplied with WINDEV are intended to help you learn the features of WINDEV.

Their source code is presented in details.

These examples are found in the "Examples" sub-directory of the setup directory of WINDEV and they can be opened from the home window of WINDEV.

Different types of examples are supplied with WINDEV:

- Complete examples: these examples correspond to full applications that can be used without any adaptation.
- Training examples: these examples illustrate a specific feature.

Features of some examples supplied with WINDEV.

## Complete examples

CRM	Used to manage the customers, orders and estimates as well as a task list and a schedule.
WD Answering Machine	Uses the telephony features to manage an answerphone. This answering machine can be interrogated and configured remotely via a menu.
WD Burner	Used to burn audio CDs and data CDs, DVDs or Blue-Ray disks.
WD Burner Lite	Used to burn data CDs, DVDs or Blue-Ray disks.
WD File Synchronization	Used to synchronize two file directories. After the synchronization, the destination directory is identical to the source directory. The comparison is performed on the presence and on the date of the files.
WD Gantt	Presents an advanced use of a Gantt chart. Used to create a Gantt chart with tasks, links, cumulative tasks and milestones. The tasks can be edited in a specific window. The created chart can be saved in a file in order to be reloaded thereafter.

WD HTTP Proxy	This example is a simplified HTTP proxy. You have the ability to save the connections and to find out the pages that have been viewed.
WD Internet Search	Used to control different search engines on Internet (Yahoo, Alta Vista, Voilà, Lycos, Excite, Nomade, Google, Euroseek, etc.). The result of the search is displayed via an ActiveX.
WD Live Messaging	Presents the functions for managing the sockets.
WD Loan	Enables you to simulate loans and to: - calculate the amount of monthly payments according to the amount of the loan - calculate the amount that can be borrowed according to the amount of a monthly payment - calculate the interest on investment from a monthly deposit For each case, you have the ability to print the depreciation schedule corresponding to your parameters.
WD Magnifier	Used to zoom a section of the screen via a magnifier. The zoom value can be multiplied by 8.
WD Mail	This application is a full email client. It is based on the Email objects. This email client is used to send and retrieve emails by using the POP, IMAP and SMTP protocols.
WD Managing a computer fleet	Allows you to manage a computer fleet. For each computer, you have the ability to define its characteristics, its components and its users. These features are implemented via the use of queries, reports and combo boxes.
WD Managing Contacts	Used to manage contacts and to send emails to them. You also have the ability to synchronize these contacts with the ones found on the mobile device linked to the PC and with the ones of Outlook.

WD Multimedia	Enables you to read and play: - videos (AVI, MPEG, ...), - sound files (MP3, WAV), - audio CDs, - Flash animations.
WD Office Image Editor	Presents the operating mode of Image Editor control of WINDEV. Open your images (WDPIC, JPG, PNG, ...) with the tool and modify them via the Image Editor control.
WD Office PDFReader	Presents the operating mode of PDF Reader control of WINDEV. Open your PDF documents with then tool and view/export their content via the PDF Reader control.
WD Office Spreadsheet	Presents the operating mode of Spreadsheet control of WINDEV. Open your Excel workbooks (XLS or XLSX) with the tool and modify the data with the Spreadsheet control.
WD Office Word Processing	Illustrates the operation of the Word Processing control of WINDEV. Open your documents (docx, rtf) with the tool and edit them using the Word Processing control.
WD Organizer	Used to synchronize the appointments in your Outlook, Lotus Notes and Google calendars.
WD Outlook	Enables you to read and write in the Outlook folders: - Messaging, - Calendar, - Contacts, - Tasks. This example is intended to operate with the standard version of Outlook supplied with Office.
WD Password	Presents a “box” of passwords. It can be used to manage the passwords created when using the Internet sites but also in applications or in everyday's life (codes, ...). It can also be used as a generator of passwords to produce secure passwords.

WD POP3Proxy	This example presents a POP3 proxy. In this example, the proxy is used to automatically archive the emails in a HFSQL database.
WD RSS Reader	Presents the use of classes for reading the RSS streams
WD Sales Management	A “trendy” application for sales management that includes dynamic tabs and ribbon.
WD Scanner - Twain	Used to: - Preview a section of a document, - Digitize a document (GIF, JPEG, BMP format), - Digitize a document and copy it into the clipboard, - Digitize a document and print it.
WD Schedule	Used to create a schedule containing click areas.
WD Telephony	Uses the telephony functions of WINDEV to: - Dial a phone number, - Detect and identify the incoming calls.
WD Video conference	Used to start a conference with a remote user. Used to send a file to a recipient and to establish a dialog with him.
WD Video surveillance	Used to broadcast the images recorded by a Web camera via a network (Internet, Intranet, etc.).
WD WebCam	Presents the integration of a Web Camera in a WINDEV window. You can: - View the image broadcasted by the camera, - Perform an immediate capture, - Capture the scene broadcasted as a Windows animation (“AVI” file).
WD Zip	Used to create and read compressed archives in “WDZ” format (WINDEV ZIP).

## Training examples

WD Active Directory DotNet	Allows you to view the content of Active Directory. This viewing is performed via .NET objects found in the System.DirectoryServices assembly.
WD Animated Menu	This example proposes different menu animations.
WD Auto Error Management	This example is an educational example about the automatic management of errors. This example presents the case of fatal errors, non-fatal errors, errors on several levels.
WD Bluetooth	This example presents the use of these Bluetooth functions: - to list the accessible Bluetooth devices - to list the services proposed by a device - to connect to a Bluetooth device - to send a file to a device via the “OBEX” functions of WLanguage.
WD ChainedList	Supplies the tools required to build chained lists (simple sequencing). This type of list is used to store and browse a set of values in an organized way.
WD Changing Skin Templates	This example presents a solution allowing the users to dynamically change the skin templates.
WD Chart	This training example explains how to use the chart control. All the available types of charts are presented.
WD Coloring Search	This example explains how to highlight the sought words in the results of a Full-Text search.
WD Controlling Excel	Used to control Excel via OLE Automation. All the Excel functions can be used.
WD Controlling IE	Used to control IE via OLE Automation. All the IE functions can be used.
WD Controlling Open Office Base	This example proposes a class used to easily handle an Open Office database without having to handle any OLE objects.

WD Controlling OpenOffice Calc	This training example explains how to control the OpenOffice.org spreadsheet. Indeed, the workbooks opened in OpenOffice.org can be created, opened and used from a WINDEV application.
WD Controlling OpenOffice Writer	This training example explains how to control OpenOffice Writer. Indeed, the documents opened in OpenOffice Writer can be created, opened and used from a WINDEV application.
WD Controlling Outlook	Used to control Outlook via OLE Automation. All the Outlook functions can be used.
WD Controlling Spreadsheets	This example explains how to control the OpenOffice Calc spreadsheet and the Excel spreadsheet with a generic code.
WD Controlling word processor	The purpose of this example is to propose a universal interface used to control these applications without having to worry about the application to control.
WD Controlling Word	Used to control Word via OLE Automation. All the Word functions can be used.
WD CSGroupware	Presents the user groupware with a HFSQL Client/Server database.
WD Dashboard	The “WD Dashboard” example explains how to use the Dashboard control.
WD DataBinding	The “WD DataBinding” example explains how a control can be linked to a WLanguage variable.
WD Detecting HFSQL Errors	Presents the module for assisted management of HFSQL errors: - the default operating mode, - the customization options.
WD Device Opening	This example is used to add or delete an application into/from the automatic opening system for the removable devices by programming.
WD DirectoryObserver	This example presents the use of the .NET delegates (.NET events) in WLanguage.
WD DirectX	This example allows you to use DirectX 9.0 in your WINDEV applications.

WD Discovering MVP - Part 1	This training example explains how to implement an MVP architecture (Model View Presenter) in a project.
WD Discovering MVP - Part 2	This training example explains how to implement an MVP architecture (Model View Presenter) in a project for a simple analysis (Customer - Order - Order line - Product).
WD DOTNET Delegate	This educational example explains how to interact with the .NET delegates
WD DOTNET WMI	The WMI (Windows Management Instrumentation) is used to exhaustively manage the applications and the devices. The WMI can be compared to a database that groups various information about a computer
WD Drawing on Chart	This example is based on the use of an internal component allowing you to draw elements on a chart control: Min/Max areas, Various areas, Flags, Plots and monitoring trackers,...
WD DuplexPrintout	This example presents a solution for duplex printout that can be disabled, which means that the reverse sides may not be printed.
WD Duplication	This example explains how to duplicate a database on a server.
WD Evaluation period	Allows you to limit the use of an application to a given period (trial period).
WD Example of unit test	This example presents the use of the test editor supplied with WINDEV via two different cases: - the test of a window - the test of a global procedure
WD External file	Explains how to handle files not described in the analysis.
WD Flowchart	This example is used to create organization charts.

WD Graphic objects	This example is an editor of graphic objects. You can: - move, resize the objects - perform selections with the lasso or with the mouse - cut/copy/paste the objects - delete/duplicate the objects - save the created documents and print them.
WD Handling the Organizer control	Example for using the Organizer control. The following features are used: Creating an appointment, Saving an organizer, Restoring an organizer
WD HFSQL Explain Analyzer	This tool is used to visually represent the details of the runtime progress of a SQL query. To do so, it uses the SQL EXPLAIN command.
WD HotKey	This example built using WINDEV lets you define your own HotKeys.
WD HTML Export	Manages the external files. Also explains how to handle the files in HTML format.
WD Input checks	Checks the data typed by the user (name, company, city, zip code, email address, etc.) in a form window.
WD Interactive Application	Explains how to use the WinEdXXX functions in an interactive application. This application is used to manage a connected house (switch the lights on, find out the temperature).
WD JAVA Calls	Presents the call to the methods of classes of a JAR archive via the WLanguage JavaExecuteFunction function. It also explains how to fill a memory table (same operating mode in Java and in Windows).
WD Java Chart Designer	This example explains how to create charts via the drawing functions. Three types of charts are presented: column, scatter, line.
WD JavaMail	Used to create an applet. This applet can then be used in an Internet page. This applet enables you to read and send emails..

WD Jump Lists	JumpList in Windows 7 A JumpList is a menu accessible from the Start menu or from the application icon in the taskbar.
WD Lasso	Uses a selection lasso in an Image control to perform a zoom in a preview control.
WD Mailshot	Used to send a "mailing" by Internet.
WD Managing HFSQL Errors	This example presents the management of HFSQL errors (duplicate, integrity). This example explains how to customize the process when an error occurs.
WD Managing the Java errors	This example presents the operating mode and the implementation of the mechanism for managing the WINDEV errors.
WD MIMEExtraction	Used to extract the attachments found in an email.
WD Multi-windowing	Used to understand the principle of multi-windowing in WINDEV.
WD Multiple Clipboards	This example automatically manages a multi-level clipboard between all the WINDEV applications started on the computer.
WD Multitouch	This example is using an internal component that gives you the ability to use the Windows Touch features in a WINDEV application.
WD Network frames	Analyzing the frames that circulate on a network allows the administrators to audit the network and to detect the problems. To listen to the network, the application uses the Winsock 2 API and the RAW Sockets.
WD OAuth	The example proposes to connect to Google and Microsoft via the AuthIdentify function and the OAuth2Parameter type.
WD Office365	This example presents the connection to Microsoft Office365 APIs and it allows you to use the Outlook contacts.
WD OLEDB	Used to access external databases via OLE DB.
WD OPC	Presents the implementation of OPC protocol in WINDEV.

WD OSD	OSD means: On-Screen Display. This is the name given to the interfaces that appear on the computer screens or on the TV screens for example. They are used to perform settings in most cases. These types of menus are displayed above all the other elements found on the screen. This example explains how to create this type of interface in WINDEV.
WD PC Info	Used to find out the language of the computer on which the application is run.
WD Persistence	This example presents the use of InitParameter, LoadParameter, SaveParameter. These functions are used to configure the backup of controls, variables and parameters.
WD PivotTable	This example presents the use of the Pivot Table control. This Pivot Table control is used to display dynamic statistics by performing groupings. These statistics are calculated from the database.
WD PowerPoint Generation	This example explains how to generate a PowerPoint presentation from: - the data found in a WINDEV application such as the data found in tables, charts .. - and from a base PowerPoint document (.pptx).
WD Printing TreeViews	Allows you to print a TreeView. This example uses a "TreePrnt" class that can be easily re-used in any treeview control.
WD Pure Menu	This example proposes a menu that can be customized via a control template.
WD Queries	Allows you to perform calculations, to pass parameters or to print the result of a SQL query on a HFSQL data file.
WD Reflexive link	Uses a "reflexive" link to manage the family trees: each person has several parents and several children All these records are stored in the same file.
WD RemoteControl	Used to implement the remote control of a WINDEV application.

WD Reports	This example presents the different methods for creating a report: - prints based on different data sources (queries, variables, ...) - prints based on controls (Table, Spreadsheet, PVT, ...) - printing composite reports - specific prints (portrait/landscape, report with watermark, report with bar code, ...)
WD Riak	WD Riak presents an example for using the Riak database in WLanguage.
WD RibbonSearch	The "WD RibbonSearch" example proposes a solution to find an element in a Ribbon control from its caption and tooltip.
WD SchedulerWithBreak	The "WD SchedulerWithBreak" example proposes a solution for managing breaks in a Scheduler control
WD Screen saver	Used to manage a screen saver via the management of timers and Windows events.
WD Search Control	This example explains how to manage an edit control used to perform "Google" FullText searches in your database.
WD Search on composite keys	Presents the use of filters and the use of composite keys in order to optimize the search times on a HFSQL data file.
WD Server trigger	This example presents the use of server triggers.
WD Service	This example presents the creation of a "service" application with WINDEV.
WD Sharepoint	This example presents the authentication method toward the Webservice as well as the use of the search.
WD SharingMemory	This example explains how to use the functions for sharing memory.
WD Simple OOP	Example about the OOP with WINDEV.
WD Syntactic Coloring	This example proposes a base class used to perform a syntactic coloring on a text.

WD SystemAPIs	Presents the use of Windows APIs.
WD Text-To-Speech	This example explains how to perform a speech synthesizer in a WINDEV application by using the SpeechSynthesizer class of the .NET framework.
WD Timeline	Proposes an internal component used to handle a 'Timeline' control template. A timeline is used to display in a linear way a set of events that occurred over a given period of time.
WD Transaction	Illustrates the operating mode of transactions when placing an order.
WD Treemap	Uses the TreeMap control in order to view the content of a directory and the space occupied by each file included in this directory.
WD Trigger	Used to enter the expenses for the contributors of a company. Two modes are available: - View with ability to perform modifications (read/write) - Read-only The triggers are used to check the access to the files according to the selected mode.
WD Universal Replication	Explains how to synchronize the data of different sites by using the universal replication.
WD User groupware	Presents the operating mode of User Groupware.
WD Using DotNet classes	Allows you to use the .NET classes in WLanguage. Used to: - analyze a character string, - configure the dialog boxes displayed.
WD Using DOTNET	Used to convert a currency into other currencies. The basis of this project (the cEuro class) is used to create a .Net assembly.
WD Using sockets	Presents the use of sockets in client/server mode. Each socket is handled via a thread.

WD Using Webservices	Client project: This example explains how to retrieve, from a WebService, images according to keywords. This project is the client that connects to the "WD Webservice Server" WebService. Server project: This example explains how to create a WebService that provides images according to keywords. This project is the Webservice that is used by the WD Webservice Client project.
WD WDSETUP Plugin	This project contains the elements required to create a setup plugin (or "extension") for WDSETUP.
WD Who is locking	Signals to the users of a HFSQL database in network who is locking a record.
WD WindowsUpdate	This example explains how to manage the updates available on Windows Update.
WD Wizard	Manages a "Wizard" window.

Additional examples will be available for download from our site ([www.windev.com](http://www.windev.com)).

