

25

DEVELOP 10 TIMES FASTER

## How to discover WINDEV?

WINDEV is a powerful Windows-focused Integrated Development Environment, that includes 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** Read the “Concepts” (online or on physical documentation).  
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” (online tutorial + exercises).  
The tutorial provides a first “hands-on” approach to WINDEV. It allows you to familiarize yourself with the main editors of WINDEV.
- 3** Try the examples.  
Try the examples provided with WINDEV in the fields you are interested in (sales management, scheduling, etc.).

The online help, available on Internet at <https://doc.windev.com> or installed with WINDEV, allows you to quickly find the syntax of a WLanguage function, get help about the interface and more, ... For each programming theme, you will find a description of the associated feature and the list of the corresponding WLanguage functions.

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

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

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

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

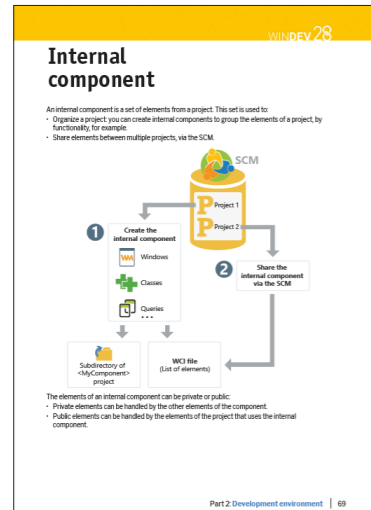
This documentation is not contractually binding. Modifications may have been made to the software since this guide was published. Please check the **online help**.

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

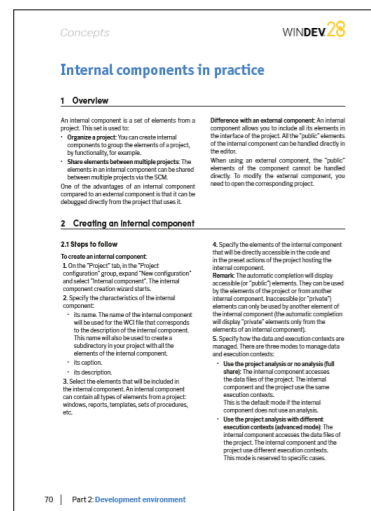
# Organization of WINDEV Concepts

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.

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



Concept page



Implementation page

# Table of contents

**PART 1 - MAIN CONCEPTS**

- Project and analysis ..... 9
- Application development cycle ..... 12
- UI: Windows ..... 14
- Modal, non-modal or MDI windows ..... 15
- Internal window ..... 17
- Window templates ..... 18
- Types of standard controls ..... 32
- Smart controls ..... 37
- Control templates ..... 48
- Edit: Reports ..... 49
- Report templates ..... 50

**PART 2 - DEVELOPMENT ENVIRONMENT**

- WINDEV editors ..... 53
- Project dashboard ..... 57
- WINDEV/WEBDEV/WINDEV Mobile: 100% compatible ..... 58
- Project configuration ..... 59
- Multiple generation ..... 60
- Custom-Folders: Organize your project ..... 61
- Source Code Manager (SCM) ..... 62
- Internal component ..... 69
- External component ..... 72
- Generation modes ..... 80
- Project Monitoring Center ..... 85
- Managing requirements ..... 86
- Managing tasks ..... 87
- Managing suggestions and incidents ..... 88
- Managing business rules ..... 89

**PART 3 - DATABASES**

- Analysis: Database structure ..... 93
- Types of accessible files ..... 102
- HFSQL Classic ..... 104
- HFSQL Client/Server ..... 105
- HFSQL Client/Server cluster ..... 106
- HFSQL: files physically created ..... 107
- Binding controls to data ..... 108
- Queries: concepts ..... 110
- Embedded queries ..... 111



Table/Looper controls .....112  
Universal replication .....114  
3-Tier architecture .....115

**PART 4 - ADVANCED CONCEPT**

RAD (Rapid Application Development) .....119  
UML models .....120  
User Groupware .....127  
Multilingual applications .....135  
Optimizing an application .....141  
Distributing "Reports and Queries" .....143  
UMC: User Macro-Code .....144  
Available tests .....146  
Debugger: concepts .....147  
Unit tests/Automated tests .....149  
Unit tests on the executable .....150

**PART 5 - SETUP**

The WINDEV Framework .....159  
Modes to install an application .....161  
Monitoring your applications .....170

**PART 6 - COMMUNICATION**

Managing emails .....173  
Transferring files via FTP .....175  
Managing sockets .....177  
WINDEV and telephony .....179  
Webservices: concepts .....180

**PART 7 - APPENDICES**

Components provided with WINDEV .....183  
Examples provided with WINDEV .....186

**PART 1**

**Main concepts**

28

DEVELOP 10 TIMES FASTER



# Project and analysis

The development of a **Windows Application** with WINDEV is based on two main elements: the Project and the Analysis.

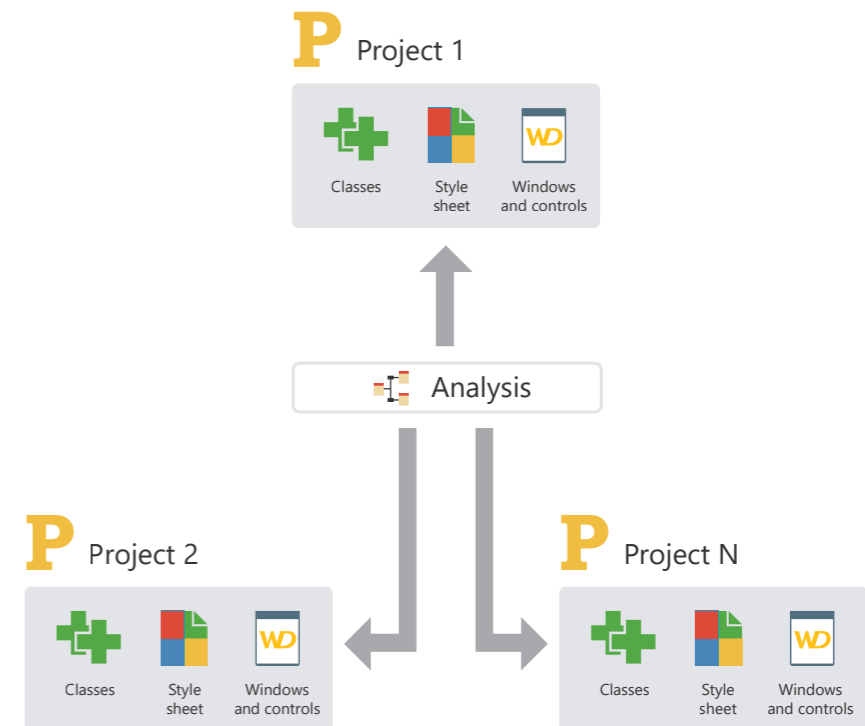
A **WINDEV Project** is a set of elements (windows, reports, controls, classes, components, etc.), which put together, make it possible to create a Windows application.

A **WINDEV Analysis** is a diagram of the data used by the application. The analysis groups the different data files (tables) 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.



# Projects in practice

## 1 Overview

The first step to create an application consists in creating a project.


When creating a project, you will be asked multiple questions so that your project meets all your

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

**Remark:** The analyses are presented in Analysis: database structure.

## 2 Creating the project

To create a project:

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

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

- **the type of project.** Multiple options are available: application, component, Java application, etc. These options will be presented later. In most cases, a project is used to generate an application (executable).
- **the operating system** (Windows or Linux).
- **the creation mode** (blank project or based on an example).
- **the runtime mode** and the type of executable to generate (if necessary).
- **the creation mode of the project.** You can create a blank project or a project based on an existing example.
- **the name and location.** These options cannot be modified. The project corresponds to a ".WDP" file. All elements associated with the project will be created in the specified directory.

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

- **the code style.** This code style is used to automatically prefix names of controls, windows, etc. It is also possible to automatically add prefixes when creating variables.

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

3. Specify **whether the project must use a database.** If yes, you can use an existing database, or create a new one.

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

**Remarks:**

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

## 3 Dashboard and project elements

### 3.1 Dashboard

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

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

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

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

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

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

To open the dashboard of your project:

1. In the open document tabs, open the context menu of the "P" ("Project") icon.
2. Select "Display dashboard".

For more details, see Project dashboard.

### 3.2 Project elements

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

To know the list of elements of a project, on the

"Project" tab, in the "Project" group, click "List of elements".

This option is used to:

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

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

## 4 Operations performed on a project

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



- Save a project,
- Restore a project,

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

For more details, see the online help.

## 5 Technical documentation of the project

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

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

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

For more details, see the online help.

### 5.2 Print areas

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

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

1. Display the print areas in the current editor. For example, in the data model editor, go to the "Display" tab, "Help for edit" group, and check "Print areas".

2. A series of borders that represent the printable areas in the documentation will appear in the current editor.

3. Reduce the graphic representation (change the value in the combo box with a percentage in the "Display" pane, or use the Ctrl + Mouse wheel shortcut) to see the entire graph.

4. With the mouse:

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

### 5.1 Type of technical documentation

There are different types of technical documentation:

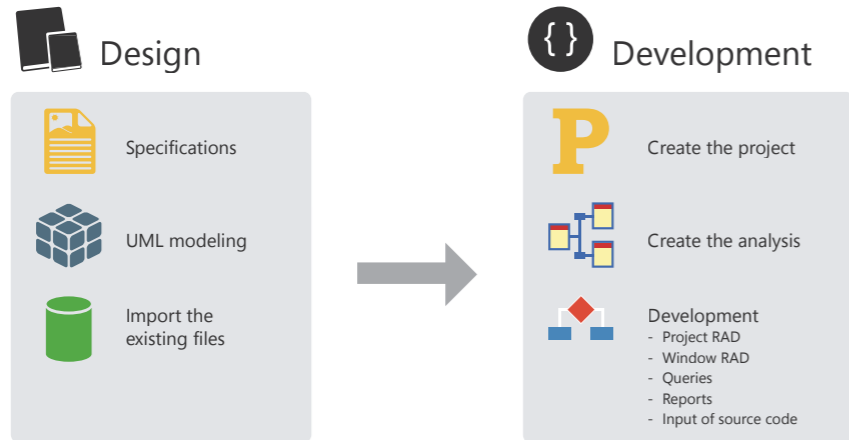
- Overall: Contains basic information about the project elements.
- Code: Contains the code of all the project objects and procedures.
- Full documentation: Contains all project information. This type of technical documentation can be very extensive.
- Custom: Corresponds to a selection of information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

#### Generating the technical documentation

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

# Application development cycle

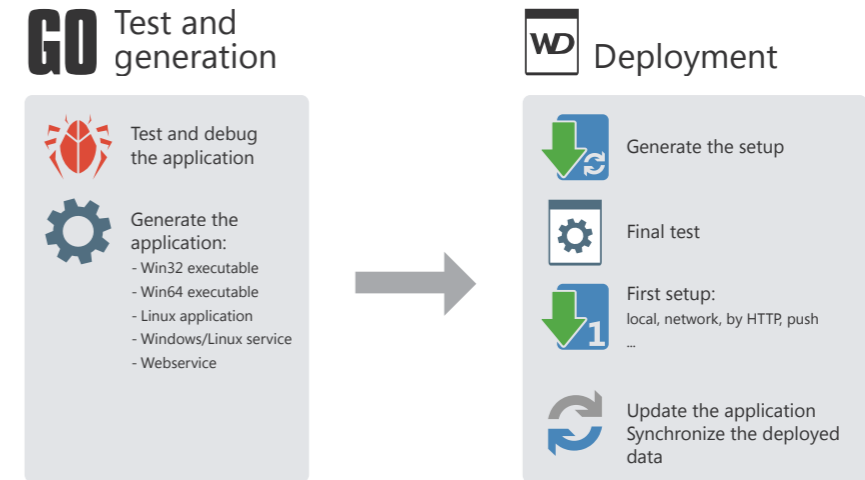
WINDEV covers the entire application development life cycle:



## Details of the different steps:

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

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



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

**Deployment phase:** A WINDEV application can be deployed using one of the following methods: download, over a local area network (including Push mode) or via the Internet. In any case, the HFSQL data files will be automatically updated based on the changes made to the application.



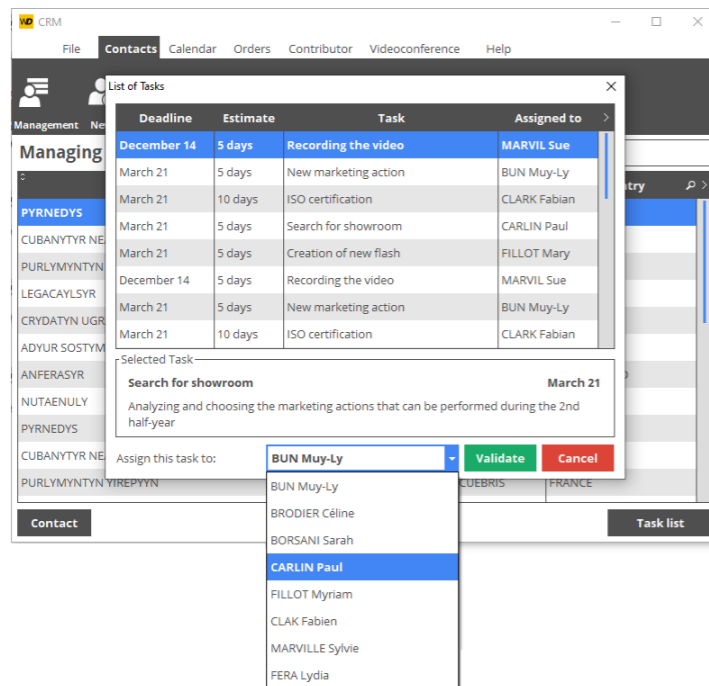
# UI: Windows

The UI (User Interface) of an application is mainly composed of windows.

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

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

- powerful controls,
- an anchoring mechanism allowing the UI to automatically adapt to the size of the display,
- a UI compilation system with error detection (empty titles, untranslated text, overlaps, etc.),
- etc.



# Modal, non-modal or MDI windows

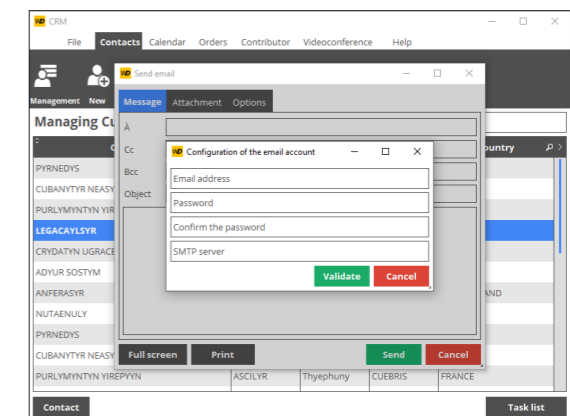
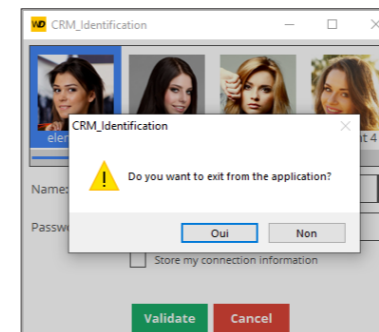
The applications created with WINDEV can use several types of windows (and mix each one of these types).

## 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, modal windows are used as:

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

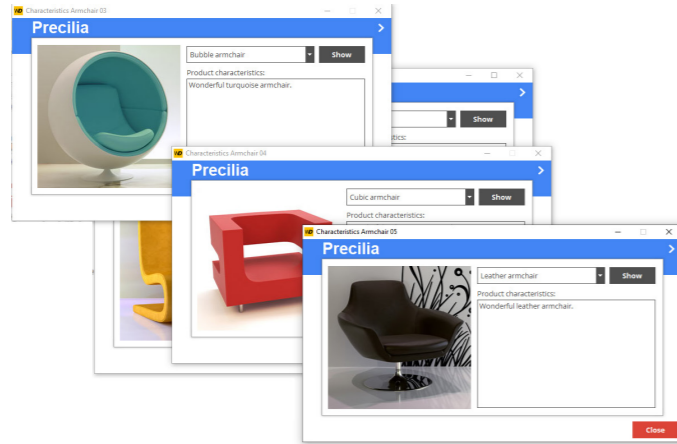
With WINDEV, when a modal window is opened, a gray shadow appears on the windows displayed in the background: it is the DDW feature or Dim Disabled Windows.



### Non-modal windows

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

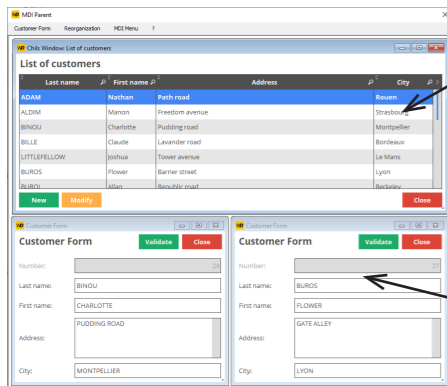
- as tool palettes,
- as dialog boxes that affect the main window when modified.



In this example, each description window of a chair opens the description window of another chair. All open windows remain accessible.

### MDI windows

In an MDI application, all the application windows are displayed in the same main window: the MDI parent window. There are two types of windows in MDI applications:



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

**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 runtime, the internal window will be dynamically merged with the host window.

#### 1. Creating an internal window.

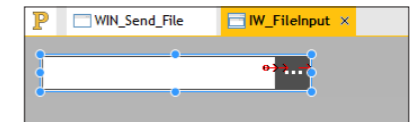
To create an internal window:

- Click in the quick access buttons.
- The creation window appears: click "Window" then "Internal window".

An internal window is a specific window that has no title bar or menu.

All types of controls can be used in this window.

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



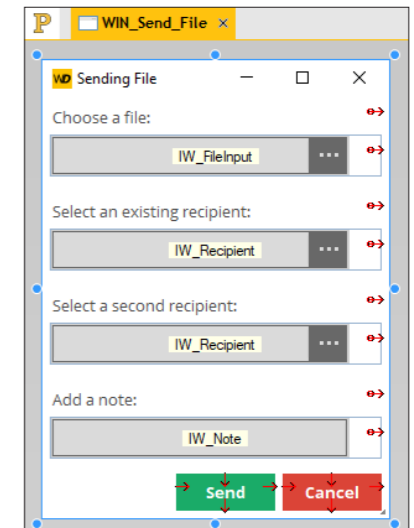
#### 2. Using an internal window.

To use an internal window:

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

#### Remarks:

- The internal window used in the "Internal Window" control can be modified through programming.
- Limitations: The host area is rectangular and no override is allowed. To perform overrides, it is recommended to use control templates.

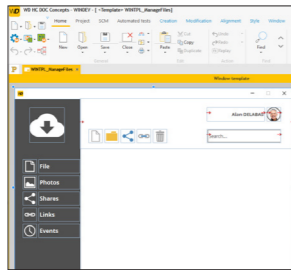


# Window templates

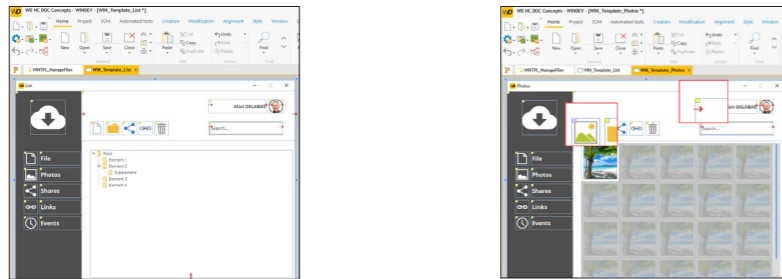
WINDEV allows you to create window templates. These templates contain all the graphic elements common to all windows of your application.

Changes made to a window template are automatically applied to all the windows that use the template.

A window template follows the style book defined for an application.



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



Template used in multiple windows.  
Elements from the template are identified by a yellow square.

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

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

## Windows in practice

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

There are multiple options to create the windows of a project in WINDEV:

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

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

For more details on how to manipulate controls in a window, see "Controls in practice", page 38.

This chapter presents the following topics:

- "Creating a window", page 19.
- "Simple operations on a window from the editor", page 20.
- "Main characteristics of a window", page 20.
- "Window and image", page 22.
- "Opening a popup window", page 24.
- "Dim disabled windows", page 25.
- "Internal window", page 25.
- "Window templates", page 26.
- "Events associated with windows" page 26.
- "Menus", page 29.
- "The status bar", page 31.

### 1 Creating a window


WINDEV allows you to create several types of windows:

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

This paragraph explains how to create blank windows, internal windows or windows based on a template. For more details, see the online help.

#### 1.1 Creating blank windows

To create a blank window:

1. Click  in the quick access buttons. The new element window appears: click "Window" then "Window". The window creation wizard starts.
2. Choose the "Blank" type.
3. Select the window skin template.
4. Validate the window creation. A blank window is created in the editor. The save window appears automatically.

5. Specify the window title and name. The window corresponds to a ".WDW" file. By default, this file will be created in the main project directory. This name will be used to handle the window.

6. Validate the window.

7. Create the controls in the window.


**Remark:** The main characteristics of the windows are available in the online help. The different types of controls are presented in Controls in practice.

#### 1.2 Internal window

An internal window is a window without title bar or 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  in the quick access buttons. The new element window appears: click "Window" then "Window". The window creation wizard starts.
2. Select the "Internal window" tab.
3. Select the type of window and the associated skin template.
4. Validate. The save window appears automatically.




### 1.3 Window based on a template

WINDEV also allows you to create a window based on a window template. In this case, simply select the desired window template.

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

To create a window based on a window template:

1. Click  in the quick access buttons. The new element window appears: click "Window" then "Window". The window creation wizard starts.

2. Select the "Based on a template" tab.
3. Select the desired template and the associated skin template.
4. Validate. The save window appears automatically.

Changes made to a window template are automatically applied to all the windows that use the template.

Window templates allow you to keep 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 actions on the windows:

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

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

For more details, see the online help.

## 3 Main characteristics of a window

### 3.1 Characteristics

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

- be **resizable**: the end user will be able to change the window size. Depending on the anchors defined for each control, the position of the controls can change when the window is resized (see the online help).
- include a **background image**: this image is displayed in the window background and the controls will be positioned over the image. You can also define the resize options via the 9-slice scaling mode. For more details, see the online help.
- be **clipped**: the window will have custom shapes (window shaped like a circle, a computer, etc.). For more details, see the online help.

- include a **drop-down menu**: this menu allows users to quickly access the main functionalities of the application. To create a drop-down menu in a window, go to the "Window" tab, "Bars and menus" group, expand "Main menu" and select "Add main menu". The first option appears in the context menu. Then, the menu options can be handled programmatically. For more details, see the online help.
- include a **status bar**: the status bar will display the help message associated with the different controls in the window. This bar can also be used to display additional information. For more details, see the online help.
- be **multilingual**: all the languages defined for the window will be available for all its controls. A window can be associated with more languages than the project (windows shared between several projects for example). For more details, see the online help.

- be automatically **dimmed** when it becomes inaccessible. This feature is called DDW, which stands for Dim Disabled Windows. For more details, see the online help.
- allow **drag and drop**. For more details, see the online help.
- allow end users to **change the UI**. For more details, see the online help.

**Remark:** To force an action on a window, simply set a Button control of this window to "Automatic execution" mode. For more details, see the online help.

### 3.2 Description window

The description window includes several tabs that group all the characteristics that can be configured for the window.

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

### 3.3 Displaying the characteristics

To open the description window:

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

### 3.4 Characteristics by tab

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

For more details on the options available in each tab, see the online help.

#### General tab

The "General" tab allows you to specify the window name and title.

#### UI tab

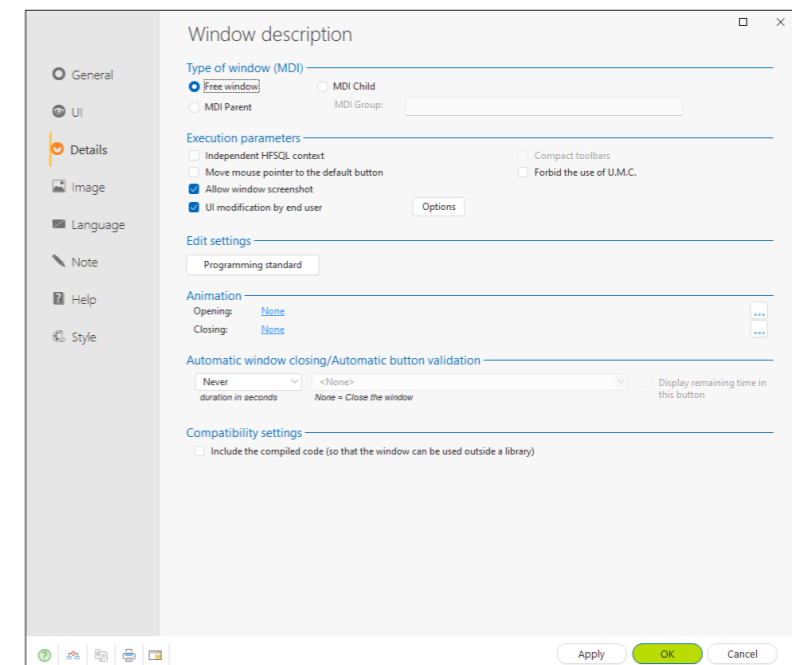
The "UI" tab allows you to define the parameters of the window interface:

- visibility of the window,
- position of the window,
- right click, mouse cursor,
- association with a context menu, etc.

#### Details tab

The "Details" tab allows you to define the different window settings:

- type of window
- UI modification by end user,
- animation,
- automatic closing.



**Image tab**

This tab allows you to define:

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

**Languages tab**

This tab shows the languages supported by the window as well as the language displayed in the editor. You can add, change or remove languages.

**Notes tab**

The "Notes" tab allows you to describe the window. This information will be printed in the technical documentation of the project, window, etc..

**Help tab**

The "Help" tab allows you to associate a context-sensitive file with the window. This help file contains the context-sensitive help pages displayed by each control.

**Style tab**

The "Style" tab allows you to define the window style:

- Show the title bar.
- Show icons.
- Show the status bar, etc.

**4 Window and image**

**4.1 Overview**

To customize the style of the windows in your applications, WINDEV allows you to use:

- background colors. Several standard colors can be used. The background color is configured in the "Style" tab of the window description.
- images. These images are used as the window background. The controls are displayed over the image. The background image can be set in the "Image" tab of the window.

**Remark:** To customize your windows, you can also use a skin template or a window template.

**4.2 Setting the background image**

The settings are defined in the "Image" tab of the window description.

You can set the background image by selecting a file or by choosing an image from the image catalog. The main image formats that can be displayed are: BMP, JPEG, TIFF or GIF.

You can also drag the image from the file explorer and drop it onto the WINDEV window.

The options available for the image are:

- Management of multilingual images.
- Display mode.

- Position.
- Transparency.
- Symmetry.
- 9-slice scaling.
- Clipping the window according to the background image.

**Management of multilingual images**

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

**To define a multilingual background image:**

1. Click the button to the right of the background image name.
2. In the menu that appears, select "Multilingual".
3. Choose the appropriate image for each language defined in your window.
4. Validate. The "Multilingual Value" text appears in the background image field.

**Display mode**

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

- **100%:** the image is displayed at 100% from the upper-left corner of the window.
- **100% Centered:** the image is displayed at 100% and is centered in the window.
- **Stretched:** the image is stretched to occupy all the space in the window.

- **Tiled:** the image is displayed at 100% and is repeated in the background of the window to occupy all the available space.
- **Homothetic:** the image is proportionally enlarged to be entirely displayed in the window.
- **Homothetic centered:** the image is homothetic and it is centered in the window.
- **Homothetic extended:** the image is proportionally enlarged so that the smallest side of the image is entirely displayed in the window.
- **Homothetic centered extended:** the image is proportionally enlarged and centered in the window.
- **Homothetic without enlargement:** if the image can be entirely displayed in the window, it is displayed at 100%. If the image is larger than the window, it is automatically resized to be displayed in its entirety.
- **Homothetic centered without enlargement:** the image is centered without enlargement.

In homothetic mode, the "High-quality display (HQ)" option allows improving the image quality when it is reduced from its original size.

**Position**

This parameter sets the position of the upper-left corner of the image, relative to the window area. These options will be kept even if the image is modified programmatically (**Image** property).

**Transparency**

The transparency options for the background image are as follows:

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

Transparency: none      Transparent magenta



- **Top left pixel:** all pixels with the same color as the top left pixel of the image will be transparent. These options will be kept even if the image is modified programmatically (**Image** property).

**Symmetry**

The symmetry options are used to change the orientation of the image in the image file. These options will be kept even if the image is modified programmatically (**Image** property).

Initial image:



Vertical symmetry

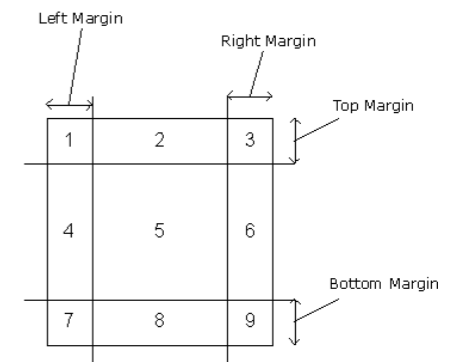
Horizontal symmetry

Horizontal and vertical symmetry



**9-slice scaling**

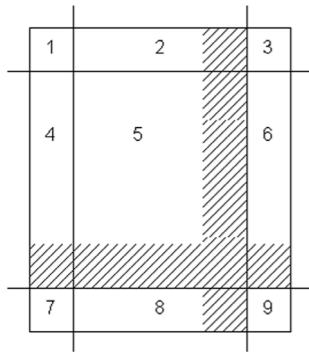
If this option is selected, the 9-slice scaling mode will be used on the background image of the window. The 9-slice scaling method consists in dividing the background image into 9 areas.



**For each area,** you can define whether the section of the image corresponding to the area must be displayed or not: simply click the area to make it visible or invisible. If the image section is not displayed, it will be replaced by the background color of the window or control.

The margins are used to define the size of each area. You have the ability to enter the values of the different areas or to move the line corresponding to the margin.

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



The 9-slice scaling method is used to manage the behavior of each area when resizing the window. For each area, the image will be enlarged:

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

### 4.3 Clipped window

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

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 the window is displayed at runtime: the window will be clipped.

**Caution:** The image borders must be as sharp as possible (no gradient, etc.) 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.

## 5 Opening a "popup" window

### 5.1 Overview

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

A "popup" window is a small window that opens above the current window. In most cases, this window allows users to select a value. Then, it returns the value selected by the user to the main window. This window closes automatically when the user clicks inside or outside the popup window.

For example:



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

- by default, it appears below the control that opens it (above or to the left if there is not enough space below the control). The initial position can be changed programmatically.
- it closes automatically when it loses focus or when **Close** is called.
- only the closing icon is displayed in the title bar of the window (if the title bar was defined in the style).

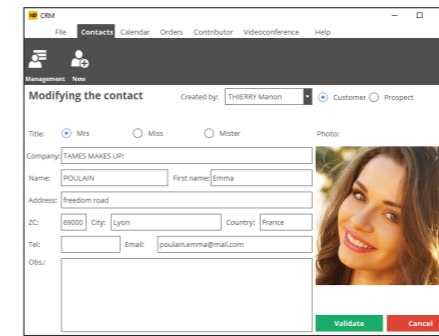
## 6 Dim disabled windows

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

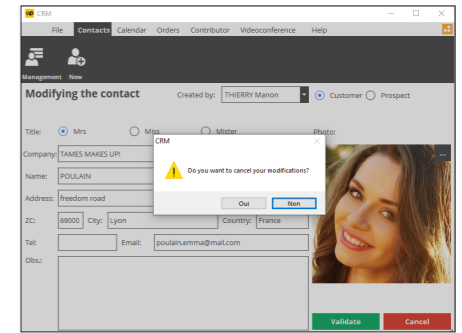
The inactive windows can be automatically dimmed. This feature allows you to always know which window to use.

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

For example:



Step 1: You change the customer information and then click "Validate".



Step 2: A window prompts you to confirm or cancel the changes. The main window (in the background) has become unavailable. It is automatically dimmed. You immediately identify the window that must be used.

The DDW features are available for the entire project or they can be customized for each window. You can also set the DDW feature programmatically. For more details, see the online help.

## 7 Internal window

Internal windows allow you to dynamically share the same UI section in one or more applications.

The UI to be used multiple times is created in an "Internal Window".

This UI can be used in the different windows of your application via the "Internal Window" control.

**Remark:** The internal window can come from a component.

For more details, see the online help.



## 8 Window templates

WINDEV allows you to create window templates. These templates contain the graphic elements and code common to all the windows in your application. Changes made to a window template are automatically applied to all the windows that use the template.

Window templates allow you to keep the style book defined for an application.

Using window templates in your applications allows you to:

- create windows more easily.

- set the layout of the windows.
- easily update the application style book.

A window template is as easy to create as a window. To create a window from a template, select "Based on a template".

By default, all changes made to the template are applied to the windows that use it. However, it is possible to define particular cases in a specific window by overriding the template elements.

For more details, see the online help.

## 9 Events associated with windows

### 9.1 Default events

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

- **Global declarations:**  
Declaration of variables global to the window.
- **End of initialization:**  
Executed when opening the window, before the code for gaining focus. The initialization event of the window controls is executed after this code.
- **Request for refreshing the display:**  
Used to group the code for updating the window controls: contents of controls, state (visible, grayed out...), text to refresh, etc, ...  
This event is started upon request by the following functions:
  - **RequestRefreshUI:** Executes the event at the end of the current event (asynchronous execution).
  - **ExecuteRefreshUI:** Executes the event immediately.
- **Close a child window:** Run when closing a child window opened from the current window.
- **Closing:** Executed when closing the window. Remark: resuming the input in one of the window controls in this code prevents the window from closing.

- **Set the Value property (internal windows only)**  
Executed when a value is assigned to the **Value** property in the Internal Window control.
- **Get the Value property (internal windows only)**  
Executed when the **Value** property is used in read mode in the Internal Window control.

### 9.2 Optional events

Several optional events are supported. Simply:

1. Click "Add other events to" at the bottom of the list of events displayed in the code editor.
2. The optional events window appears.
3. Select the optional event(s) to add and validate.
4. The selected events are automatically added after the default events.

You can manage, for example:

- gain and loss of focus,
- changes in one of the window controls,
- window size changes,
- mouse hover over the window,
- left mouse button down, up, double click,
- right mouse button down, up, double click,
- mouse wheel, etc.

## 10 Context menus

The user can open a context menu by pressing the right mouse button. This menu allows users to easily access the specific features of a window or control.

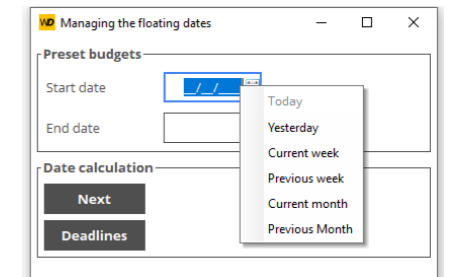
WINDEV allows you to manage two types of context menus:

- **system context menus** managed by Windows. These context menus appear when the user right-clicks on some types of controls (Edit control, Table, etc.). For more details, see Default system context menu.
- **custom context menus** managed by WINDEV. These menus are created in the window editor and can be handled programmatically.

WINDEV allows you to create custom context menus and associate them with:

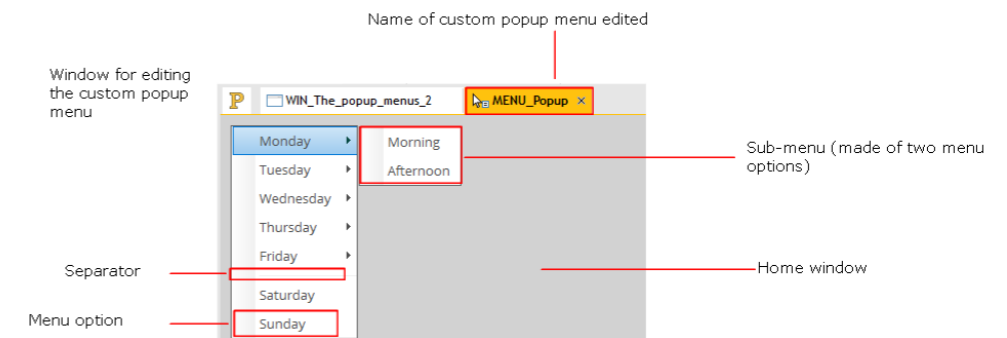
- one or more controls in a window.
- a window.

For example:



### 10.1 Context menu edit windows

WINDEV allows you to create and modify custom context menus in a specific edit window. To better understand how to manage custom context menus, let's take a look at the vocabulary related to this edit window:



### 10.2 Available features

WINDEV allows you to:

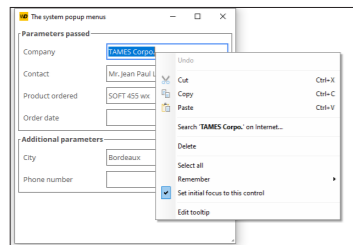
- Create a custom context menu.
- Associate a custom context menu with one or more elements (window or controls).
- Edit a custom context menu.
- Open the description window of a custom context menu.
- Delete a custom context menu.

For more details, see the online help.

### 10.3 Default system context menu

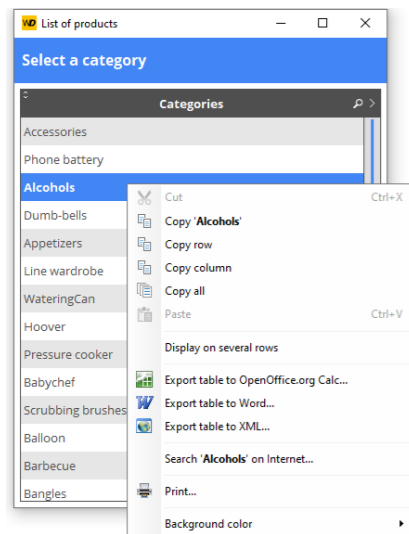
The following types of controls have a default system context menu (displayed if no custom context menu is associated with the control):

- Edit control:



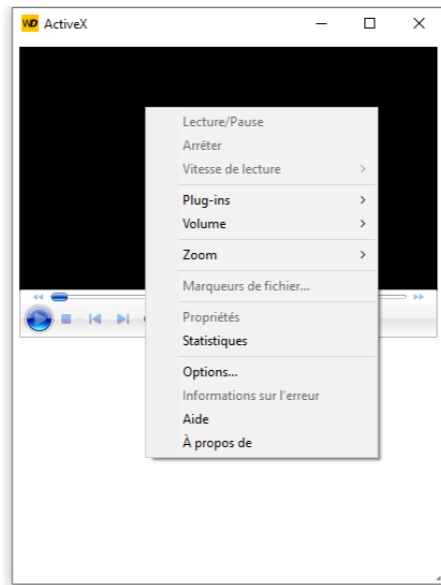
For more details on the translation of this menu, see the online help.

- table column:



For more details on the translation of this menu, see the online help.

- ActiveX control:



**Remark:** The system context menu displayed depends on the ActiveX program associated with the control.

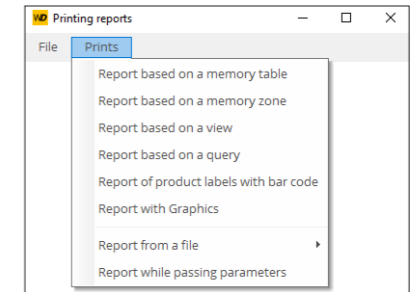
This default context menu can be disabled from the editor or in the code.

## 11 Main menus

### 11.1 Overview

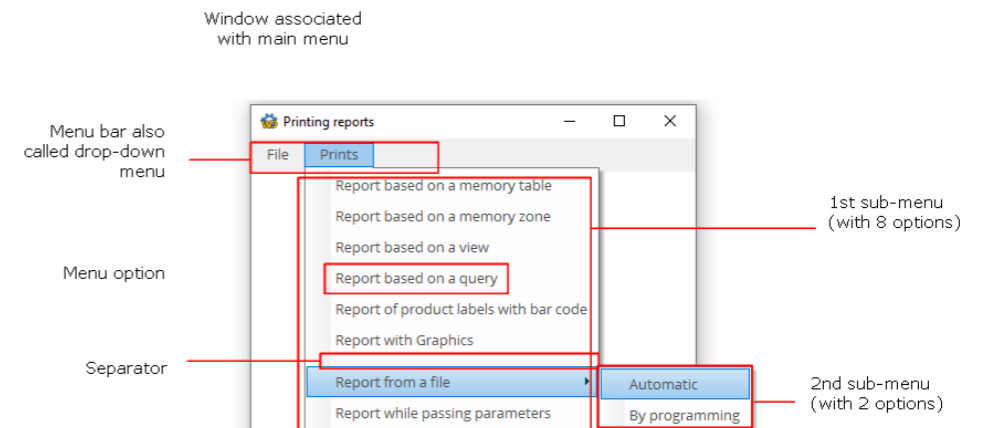
A main menu is always associated with a window. In WINDEV, this menu is displayed below the window title bar. In most cases, the applications contain a menu window. This window only contains a main menu and is used to access the other windows of the application.

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.

For more details, see the online help.

## 12 Menu options

### 12.1 Overview

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

In the editor, you can handle menu options in the same way for both main menus and custom context menus.

To handle the menu options, simply 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.
- the custom context menu of a window: on the "Window" tab, in the "Bars and menus" group, expand "Popup menus" and select "<ContextMenuName>".

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

### Special case: Tool menu

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

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

The description window of the menu options allows you to:

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

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

For more details, see the online help.

### 12.3 Handling menu options programmatically

Options are handled in the same way in context menus and main menus.

To handle a menu option programmatically, use the following syntax:

- "**<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

The status bar of a window can display:

- an **information message linked to the current control**. This message can be entered in the "Help" tab of the control description window. The message will be displayed in the status bar when the user selects the control.
- a **specific information message**. This message is displayed programmatically with **Message**.
- a **progress bar**. This progress bar is displayed programmatically with **eye**.

To display different types of information simultaneously, you can divide the status bar into several cells ("Insert a cell" in the context menu of the status bar).

### 13.1 Status bar description window

The description window of a status bar allows you to:

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

For more details, see the online help.

### 13.2 Type of cell in the status bar

The type of allows you to display:

- The progress bar. This progress bar will be run when using **eye**.
- The help message associated with the control that has focus. This help message is defined in the "Help" tab of the control or in the code using the **Message** property.
- The date of the current computer.
- The time of the current computer.

- The cursor position in the current Edit control ("Column and row of the caret" option).
- The status of the Caps Lock key that indicates whether it is enabled or not.
- The status of the Insert key that indicates whether it is enabled or not.
- The name of the current user in an application that uses the User Groupware.

Two specific options are also available:

- Programmatically: The information displayed in the status bar is specified programmatically (using **Message** for example).
- The update event: The "Whenever updating" event is automatically associated with the cell of the status bar. This event is run on a regular basis. This event can contain the elements that will be displayed in the cell, for example.

### 13.3 Characteristics of the cells in the status bar

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

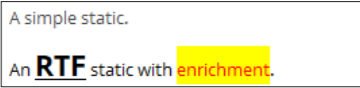

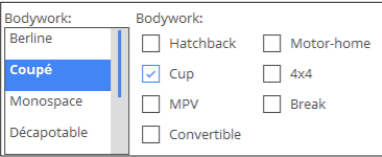
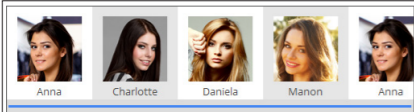
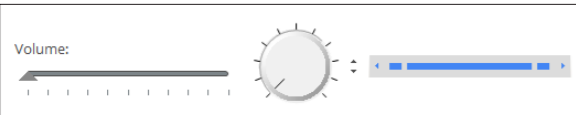
- The position of the cell in the status bar.
- The background image of the cell.
- The background color of the cell.
- The width of the cell.
- Whether the cell is editable.
- The anchors of the cell (its behavior when the window is resized).

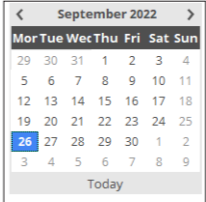

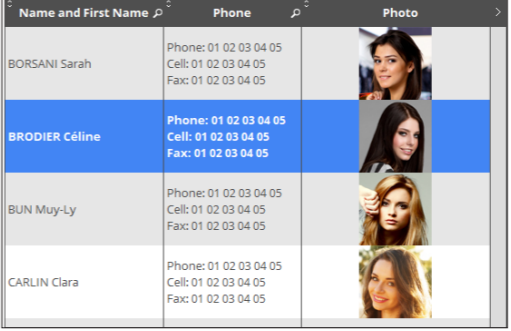

All these characteristics can be handled programmatically via the properties available for the cells in the status bars.


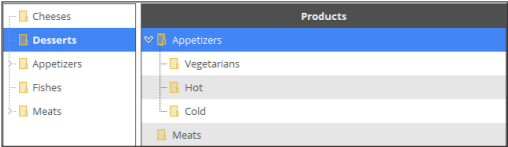
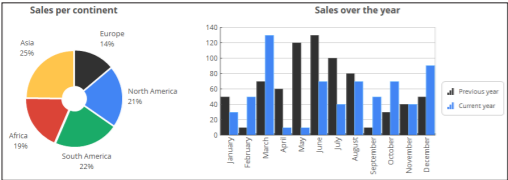
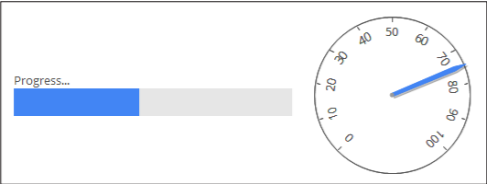


# Types of standard controls

WINDEV includes different types of controls. Let's see how to use them according to the operation to perform:

In order to...	Use the following control...
Display a text, a title, etc. 	Static
Read a PDF file	PDF Reader
Display and edit a Word document.	Word Processing
Select a value from a list (Country, City, etc.) 	Radio Button, Combo Box, List Box
Select multiple values from a list (message recipients, etc.) 	Check Box, List Box
Select one or more values from a ListView (picture directory, etc.) 	ListView
Select a value in a preset range. 	Slider, Round Slider, Spin, Range Slider

Select a date. 	Calendar
Display an image (photo, etc.), an animation or use drawing functions. 	Image
Display the contents of a data file in a table (list of customers, order details, etc.) 	Table
Show the same controls multiple times in a window (e.g. product catalog with images, etc.) 	Looper

<p>Display a video or animation.</p> 	<p>Web Camera Multimedia Conference</p>
<p>Enter information.</p> <p>Email: <input type="text" value="mike.durand@mail.com"/></p>	<p>Edit control</p>
<p>Program an action in a window (display another window, start a print job, etc.).</p> <p><input type="button" value="Ok"/> <input type="button" value="Print"/> </p>	<p>Button</p>
<p>Display information in a hierarchy (directory).</p> 	<p>TreeView TreeView Table</p>
<p>Display a column chart, a line chart, a pie chart, etc. ...</p> 	<p>Chart</p>
<p>Display a progress.</p> 	<p>Progress Bar</p>

<p>Group the controls by theme and display the themes one by one.</p> 	<p>Tab Sidebar Drawer</p>
<p>Display a bar code.</p> 	<p>Bar Code</p>
<p>Display events over a period of time.</p> 	<p>Organizer</p>
<p>Create statistical tables.</p> 	<p>Pivot Table Spreadsheet</p>

# Smart controls

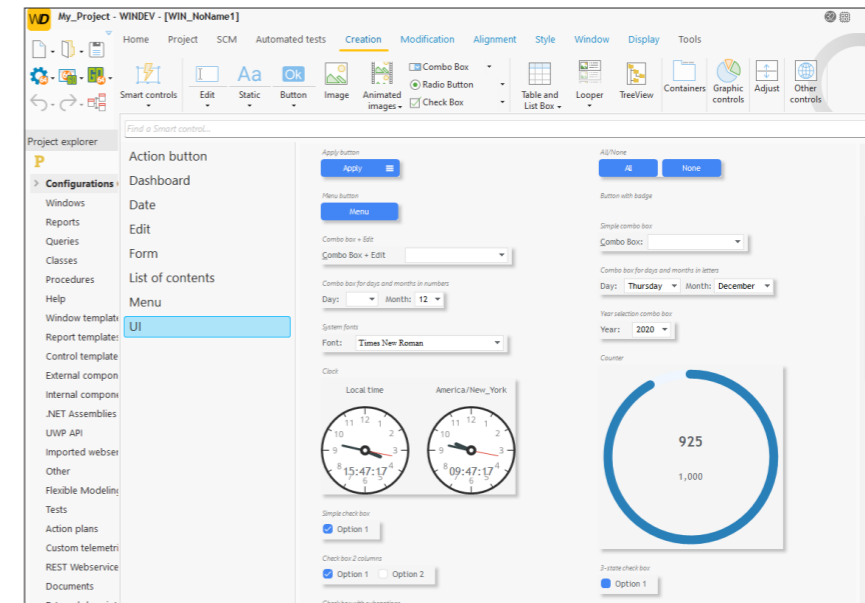
Smart controls are an innovative concept for creating UIs. A Smart control is a ready-to-use control (or group of controls) that includes a "business logic" WLanguage code where necessary. Thanks to the Smart controls, you can add sophisticated behaviors to your WINDEV applications with a single click.

WINDEV includes several Smart controls: Twitter, Instagram, Signature, Countdown, Date range selection,

Menu with drawers, Workflow, Cookie consent, Process monitoring, and more.

To integrate a Smart control into your interfaces:

1. On the "Creation" tab, in the "Smart" group, click "Smart controls".
2. Select the desired control in the window that appears:



You can:

- Browse through the different types of controls,
- Enter keywords in the search box.

Remark: The "Go" button that appears when hovering over the Smart controls allows you to see and try the control before integrating it in your window.

3. Select the desired control, drag it and drop it onto the window editor.

Represent weighted data.		TreeMap
Display important elements in a dashboard.		Dashboard
Display HTML data.		HTML Display control
Enter HTML data.		HTML Editor control
Select intervals or change zoom		Range Slider
Display a map or an itinerary.		Map
Edit an image		Image editor
Display an ActiveX control		ActiveX

# Controls in practice

WINDEV includes different types of controls that can be easily used in your windows. These controls

can be created via the "Creation" tab of the ribbon. All controls can be handled through programming.

## 1 Creating a control

### 1.1 Different types of window controls

The following controls are available in the WINDEV window editor:

- Edit control,
- Static,
- Button,
- Image,
- Image Editor,
- HTML Editor,
- Radio Button,
- Check Box,
- List Box,
- Combo Box,
- ListView,
- Sidebar,
- Drawer,
- Web Camera,
- Conference control,
- OLE control,
- ActiveX,
- Shape,
- Bar Code,
- HTML Display control,
- Supercontrol,
- Organizer,
- Scheduler,
- Rating,
- Pivot Table,
- Dashboard,
- Table,
- Spreadsheet,
- Word Processing,
- Gantt Chart,
- TreeView Table,
- TreeView,
- Diagram Editor,
- Organization chart,
- Progress Bar,
- Looper,
- Multiline zone,
- Scrollbar,
- Slider,
- Spin,
- TimeLine,
- Tab,
- Ribbon,
- Internal Window control,
- Toolbar,
- Splitter,
- Chart,
- Map,
- Carousel,
- Widget,
- Xaml,
- Calendar,
- Control template,
- Layout control,
- Dockable panel,
- PDF Reader.

All controls can be handled through programming. For more details, see the online help about each type of control, their main characteristics and their programming in WLanguage.

### 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" tab of the WINDEV menu.
2. The shape of the new control appears under the cursor.
3. Click where you want to create the control. The control is automatically created.

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

### 1.3 Creating a control associated with an item

All the controls in a window can be bound to a data file item, EXCEPT the following:

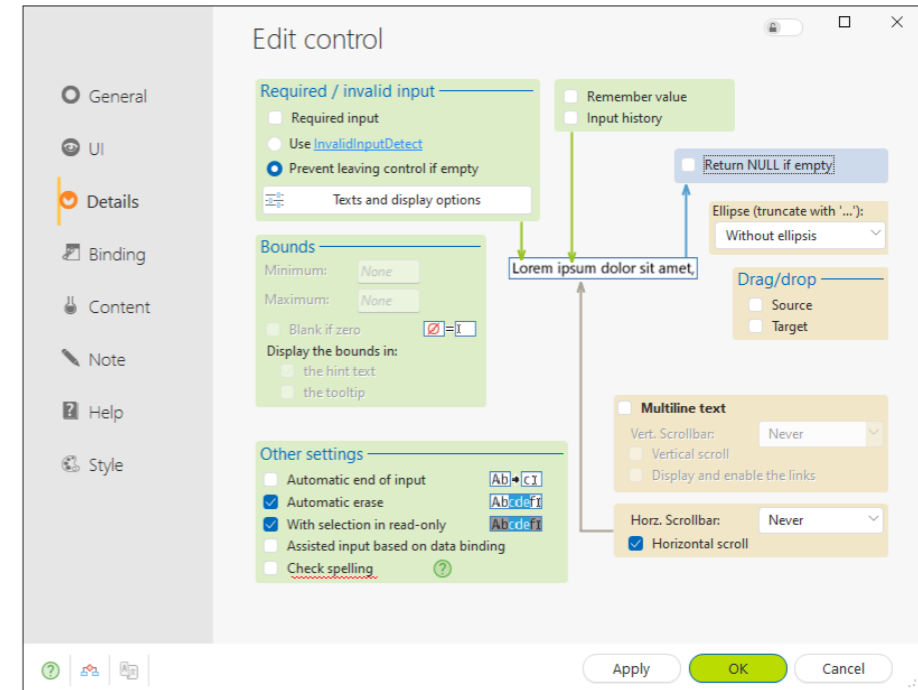
- Button
- Shape
- Tab
- Toolbar
- Map
- Ribbon
- Splitter
- Timeline
- Dashboard
- Xaml
- Carousel
- Cube
- HTML Display control
- Layout control

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

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

## 2 Control characteristics

All the controls in the window editor include a description window. This window has several tabs that group the characteristics that can be set on one or multiple controls.



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

### 2.1 Displaying the characteristics

To open a control description window:

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

**Remarks:**

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

## 2.2 Characteristics by tab

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

### General tab

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

### UI tab

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

- Initial state of the control when the window is opened,
- Control visibility,
- Control size,
- Alignment, etc.

### Details tab

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

- input parameters,
- Drag and drop, etc.

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

### Binding tab

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

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

### Content tab

The "Content" tab allows you to define:

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

### Notes tab

The "Notes" tab allows you to describe the control operating mode in detail. This information will be included in the technical documentation (project documentation, window documentation, etc.).

### Help tab

The "Help" tab allows you to enter and set the types of help systems associated with the control. A control can have:

- A tooltip, displayed when the control is hovered over by the mouse cursor.
- A help message, displayed in the status bar when the control gains focus.
- A help window with a specific title, displayed when pressing the F1 key.

A control can also be associated with a specific help page from a CHM help file. This help page will be displayed:

- as a context-sensitive help (tooltip displayed upon request), if a help file was defined for the window.
- as a help page. In this case, use **WHelp** to display the help page.

### Style tab

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

- Change the control aspect: simply select the control element to be modified, and then choose the style characteristics. Only the aspect of the current control is modified.
- Apply an existing style to the control: simply select the style to apply.
- Create or change 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 Control handles

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

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

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

- The **black handles** (different color depending on the skin template used) allow you to distinguish and resize the selected control.
- **White handles** are used in two cases:
  - simple selection: the selected control is in read-only mode. For example, these handles may appear if the control is selected and its description window is open. In this case, the control cannot be directly modified or moved in the editor.
  - multiple selection: the selected control is part of a multiple selection, but it is not the first control selected.

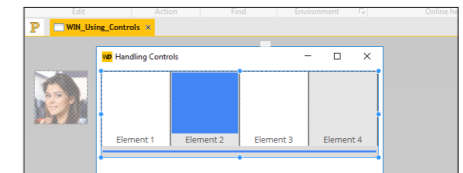
This type of handles is also displayed when a window is in read-only mode for the developer: the controls cannot be modified.

- The **gray handles** indicate the first control selected in 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 selected controls,
- Move controls,
- Change the text of the controls,
- Show an advanced tooltip (when the cursor hovers over the control) with: the control name, position, size, anchors and initial state.
- Use a control outside the window. This control will be visible in edit mode and invisible at runtime.



Remark: You can use a control outside the window to:

- associate it with the window without displaying it at runtime. This allows you to use the content of the control programmatically (like a variable).
- display new controls in a window when resizing it.

## 4 Aligning controls

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

- rulers.
- grids.
- real-time interface checker (appears automatically when positioning controls).
- advanced interface checker.
- alignment options.

### 4.1 Rulers

You can display alignment rulers in the window editor.

These rulers have snap-on guides: any control that comes close to a guide is automatically "snapped" by it.

This feature allows you to easily position, align or resize controls in a window.



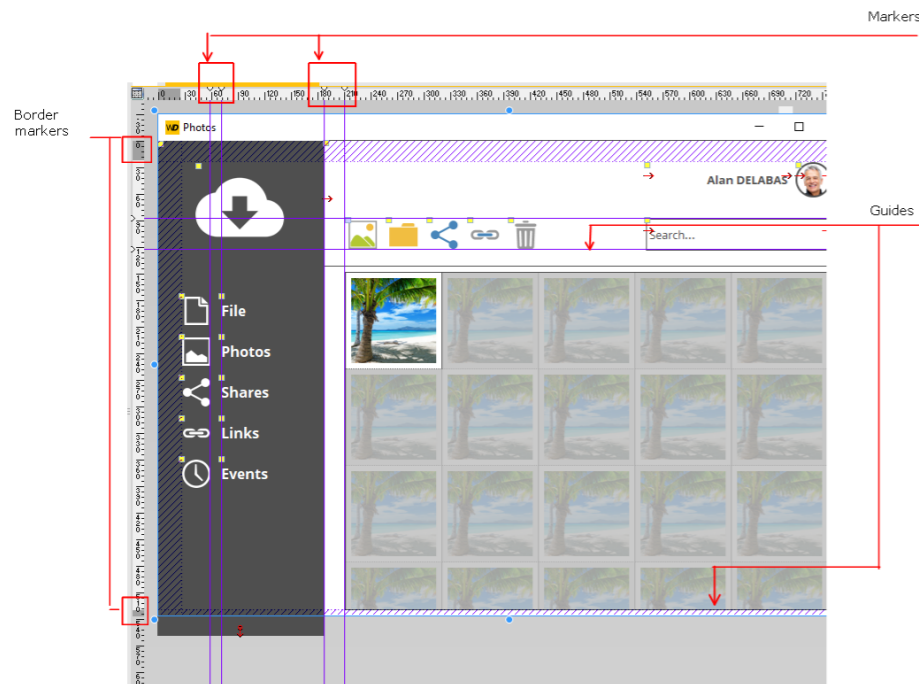
### Displaying the rulers

To display the rulers, go to the "Display" tab, "Help for edit" group and check "Rulers". Two types of markers can be used:

- markers, which allow you to align and place the controls inside the windows using the associated guides.
- border markers, which allow you to obtain a border of the same size on each side of the window. A guide allows you to view the border area.

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

Example in a window:



### Handling markers

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

### 4.2 Snap-on grid

The snap-on grid is used to 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 improve the layout of the controls in the different windows of the application.

This interface checker also enables you to apply the interface rules from the Windows standard to the current window or to the selected controls: control alignment, standardization of Button control sizes, etc.

### 4.5 Alignment options

To align multiple controls, WINDEV includes:

- a preset alignment (available on the "Alignment" or "Modification" tab, in the "Depth and Alignment" group).
- a custom alignment (available on the "Alignment" tab, in the "Other alignments" group).

The custom alignment allows you to use specific alignment properties. In the custom alignment, you can define:

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

## 5 Control edit options in a window

The edit options allow you to set:

- the display options.
- the snap-to behavior of the controls.
- the clicks and double-clicks on the controls.
- the control selection options.
- the position of the window when it is opened.

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

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

### 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" tab, in the "Help for edit" group, check/uncheck "Visible (editor)".

### 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" tab, in the "Help for edit" group, check/uncheck "Locked".

A locked control cannot be selected by clicking and dragging over it. Let's look at an example with a Tab control. When you click and drag over multiple controls in a tab pane, the Tab control is not automatically selected.

All other operations can be performed on a locked control.

## 6 Anchoring controls

The windows of a WINDEV application can be resized at runtime. With the anchoring mechanism, controls are automatically resized and positioned when the window is resized.

The anchors of a control include two settings:

- **the type of anchor:** defines the control behavior based on the new window size. The control can be moved to the left and/or down, or be stretched horizontally or vertically, ...
- **the anchor rate:** defines how much a control stretches or moves. This option allows you to:
  - keep a control centered horizontally in the window regardless of its size.
  - proportionally enlarge some controls.

These settings can be defined:

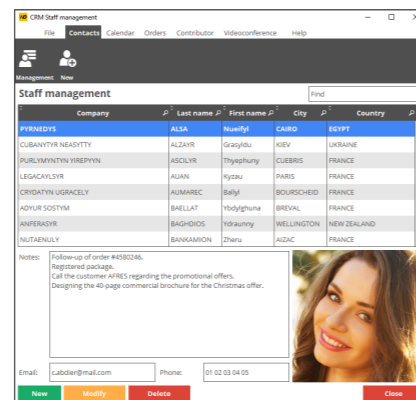
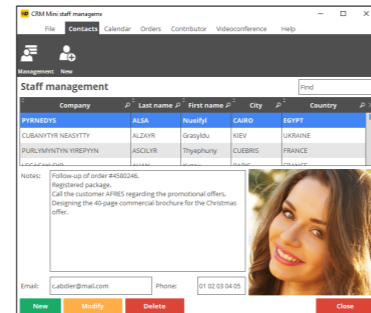
- in the window editor.
- through programming with the WLanguage properties.

### 5.3 Framing a control

To frame a control in the window editor:

1. Select the control in the editor.
2. On the "Modification" tab, in the "Help for edit" group, check/uncheck "Outlined".

Framing a control allows you to easily view all the controls of a window (especially Image controls).



### 6.1 Anchoring a control in the editor

To anchor a control:

1. Select the controls you want to anchor.
2. Define the control anchoring options:
  - in the "UI" tab of the control description window ("Description" in the context menu).
  - via "Anchor" in the context menu of the control (right click).
3. In the window that appears, select the type of anchors to be applied. The most common options are represented by different icons:



**No anchors:** The control does not change when the window is resized. No anchoring option is selected.



**Width:** The control stretches to the right when the window is enlarged. Set the "width" anchor rate to define how much a control stretches relative to the window.



**Right:** The control moves to the right when the window is enlarged. Set the "right" anchor rate to define how much the control moves.



**Horizontally centered:** The control remains centered horizontally regardless of the width of the window.



**Height:** The control stretches downward when the window is vertically enlarged. Set the "height" anchor rate to define how much a control stretches relative to the window.



**Width and height:** The control stretches to the right and downward when the window is enlarged. Set the "width" and "height" anchor rates to define how much a control stretches relative to the window.



**Height and right:** The control stretches downward and moves to the right when the window is enlarged. Set the "right" anchor rate to define how much the control moves. Set the "height" anchor rate to define how much a control stretches relative to the window.



**Vertical distribution:** This option is available only when multiple controls are selected. The selected controls stretch downward and are distributed proportionally when the window is enlarged vertically.



**Bottom:** The control moves down when the window is enlarged downward. Set the "bottom" anchor rate to define how much the control moves.



**Width and bottom:** The control stretches to the right and moves downward when the window is enlarged. Set the "bottom" anchor rate to define how much the control moves. Set the "width" anchor rate to define how much a control stretches relative to the window.



**Right and bottom:** The control moves to the right and down when the window is enlarged. Set the "bottom" and "right" anchor rates to define how much the control moves.



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

Set the "bottom" anchor rate to define how much the control moves.



**Vertically centered:** The control remains centered vertically regardless of the height of the window.



**Horizontal distribution:** This option is available only when multiple controls are selected. The selected controls stretch to the right and are distributed horizontally when the window is enlarged.



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

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

Set the "right" anchor rate to define how much the control moves.



**Centered:** The control remains centered vertically and horizontally regardless of the size of the window.

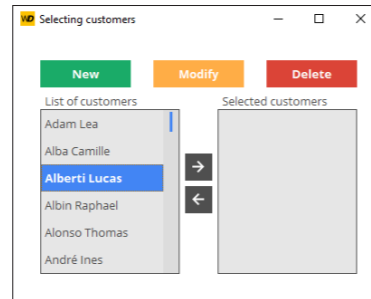
4. Define (if necessary) the different anchor rates.

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

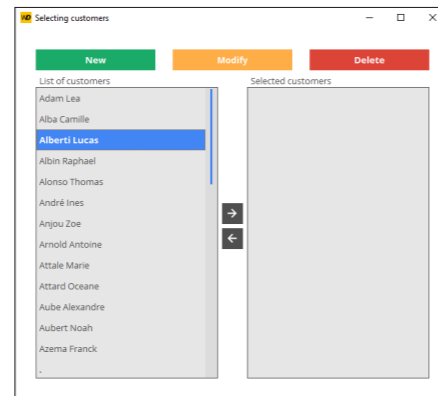
## 6.2 Example of control anchors

The following example shows how controls are anchored in a window.

Initial window:



Enlarged window:



To obtain this result:

1. Select the [-] and [->] Button controls: center these controls vertically and horizontally (☐☐).
2. Select the two List Box controls: anchor these controls to the right, horizontally and vertically (check the "right", "width" and "height" options).
3. Select the three Button controls: these controls must stretch to the right proportionally (☐☐☐).

Remark: The anchors of the controls can be set programmatically using different WLanguage properties.

## 7 Automatic button execution

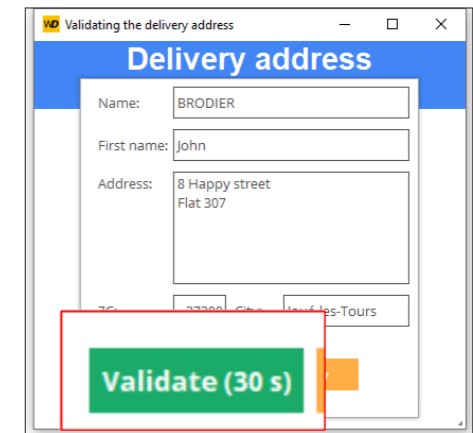
To force an action on a window, simply set a Button control of this window to "Automatic execution" mode. The principle is very simple, if no button has been used for a given period of time, the click code of the Button control associated with the "Automatic execution" will be automatically executed.

At runtime, the duration is automatically decremented. When it reaches zero, the code of the Button control is automatically run.

### 7.1 Implementation

To run the code of a Button control automatically:

1. Define which Button control must be run automatically.
2. In the window description ("Details" tab of the description window), specify:
  - the time before the automatic execution of the Button control.
  - the name of the Button control to be run.
3. At runtime, the remaining time is automatically displayed.



### 7.2 Programming

You can also define an automatic validation Button control using **DelayBeforeClosing**.

### 7.3 AAF: Automatic Application Feature

This feature is also directly accessible to the end users. They can request a Button control to be automatically run after a specific duration.

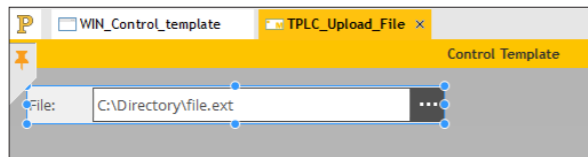
# Control templates

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

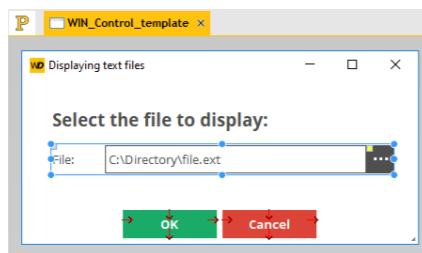
Changes made to a control template are automatically applied to all the windows that use the template.

A control template is used to:

- group a set of controls for a specific purpose.
- make the controls independent of the host window.



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



Template used 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.

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

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

# Edit: Reports

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

Reports can be displayed on the screen, saved in an HTML file, printed, etc.

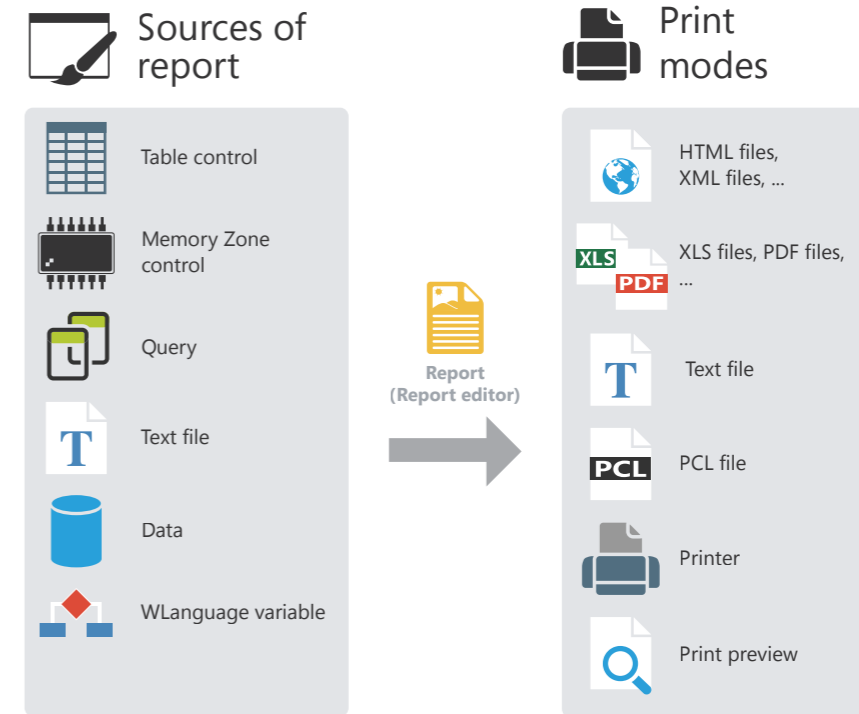
A report can be used to summarize and synthesize data.

You can:

- group data.
- sort data based on different criteria.
- perform calculations (means, statistics) or even create charts.

The diagram below illustrates the concept of reports:

- the data to print comes from a data source (data file described in an analysis, HFSQL view, query, memory area or text file).
- the report groups, sorts and formats the data.
- the report can be represented on the screen, as a file (HTML, RTF, etc.) or sent to the printer.

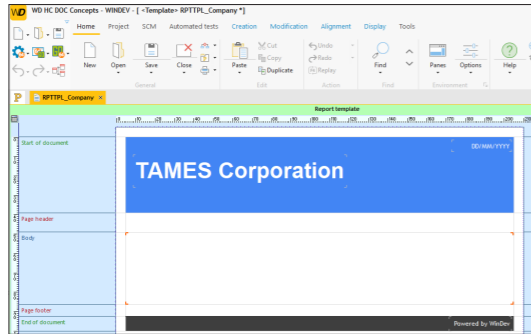




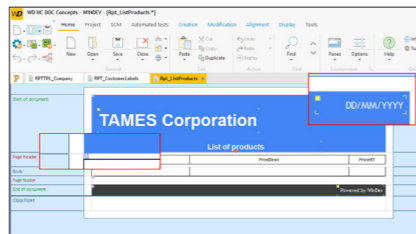
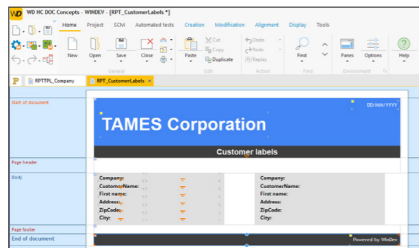
# Report templates

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

Report templates are used to easily standardize report layouts.




View of a report template in the report editor.



Template used in different reports.

The elements that belong to the template are identified by a yellow square. Overridden template elements are identified by a blue square.

A report template can be created:

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

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

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

Development environment



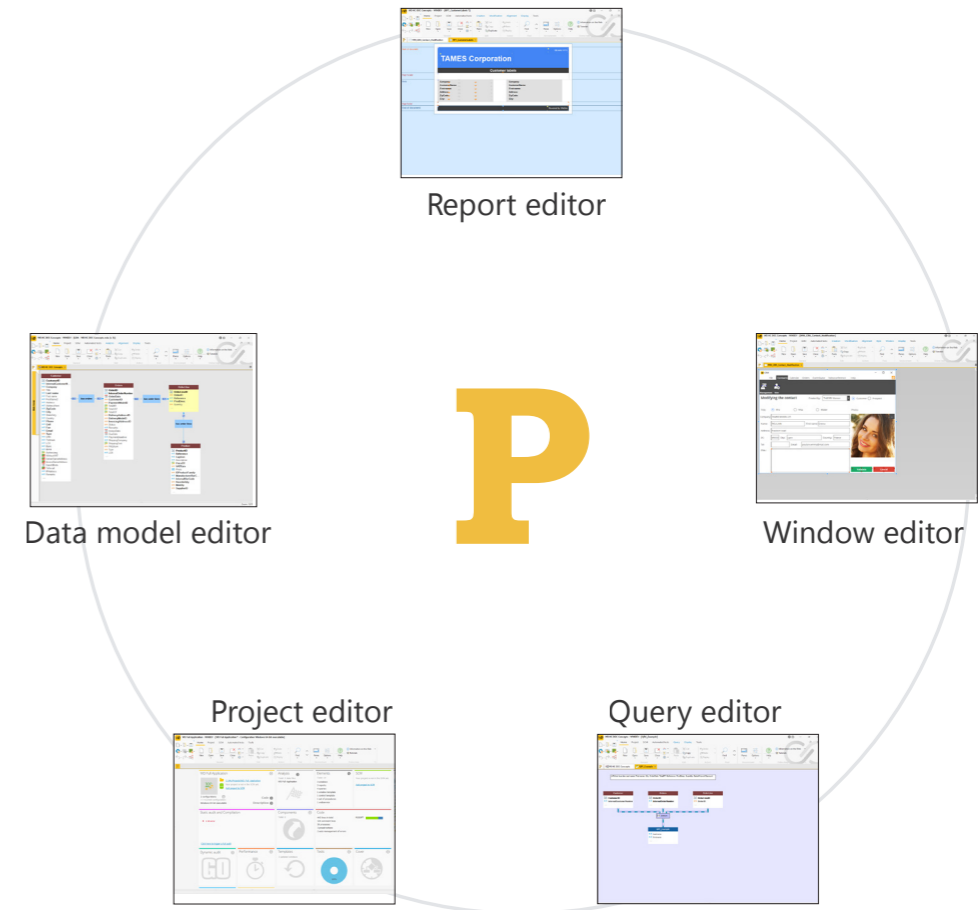


# WINDEV editors

To handle a project, WINDEV proposes several editors:

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

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



# User-friendly editors in practice

## 1 Introduction

WINDEV is built on a set of editors that operate in a single environment and adapt to the needs of developers:

- 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, etc.) used by the application.

## 2 Types of editors

### 2.1 Project editor

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

Various widgets allow you to see:

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

### 2.2 Data model editor

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

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

The data model editor supports two types of database descriptions:

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

Let's see the online help:

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

### 2.3 UML editor

UML is a visual language that can be used to:

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

The UML editor is mainly used to:

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

Let's see some characteristics of the UML editor:

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

### 2.4 Query editor

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

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

Let's see some characteristics of the query editor:

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

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

### 2.5 Window editor

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

Multiple skin templates (style sheets associated with the window) and types of windows are available. They help you make your windows and applications more user friendly.

For more details on how to create a window, see the online help.

Let's see some characteristics of the window editor:

- WYSIWYG editor ("What You See Is What You Get"). You can directly view your windows as they will appear to the user.
- Creation and description of windows and controls.
- Drag and drop to copy, paste or move controls from one window to another.
- Window skin templates and preset control styles.
- Icon catalogs to associate images with controls.
- Real-time interface checker to position controls more easily.
- Anchoring of controls in the window.
- Ability to enter the captions of the controls directly in the editor work area.
- Context-sensitive help in the windows.
- Zoom in the window editor.
- Relation between the different panes and elements of the editor.

### 2.6 Report editor

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

Let's see some characteristics of the report editor:

- WYSIWYG editor ("What You See Is What You Get"). You can directly view the reports as they will be printed.
- Easy-to-create, zero-code reports.
- Drag and drop to copy, paste or move controls from one report to another.

- Report skin templates and preset control styles.
- Real-time interface checker to position controls more easily.
- Use of a form in a report background.
- Creation of multi-column labels.
- Ability to edit a report in HTML (to publish it on the Internet for example) or RTF (to use it in a word processing software).
- Zoom in the report editor.

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

### 2.7 Code editor

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

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

Let's see some characteristics of the code editor:

- Automatic formatting.
- Automatic completion.
- Glossary of functions.
- Typing error detection and spelling checker.
- Incremental compilation.
- Insertion of WLanguage events specific to the use of the mouse or keyboard.
- History of code changes and ability to roll back.
- Ability to open multiple code windows (to compare code, for example).

- Zoom in the code editor.
- Relation between the different panes and elements of the editor.

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

### 2.8 Help editor

The help editor allows you to create a set of help pages ("CHM" file) based on your project.

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

Let's see some characteristics of the help editor:

- WYSIWYG editor ("What You See Is What You Get"). You can directly view the help pages of your project.
- Compilation of help pages and creation of a CHM file.
- Ability to create an index and a table of contents.
- Insertion of images.
- Insertion of links in the help pages.
- Ability to display the titles of the pages in a bar.
- Zoom in the help editor.

### 2.9 Image editor

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

Let's see some characteristics of the image editor:

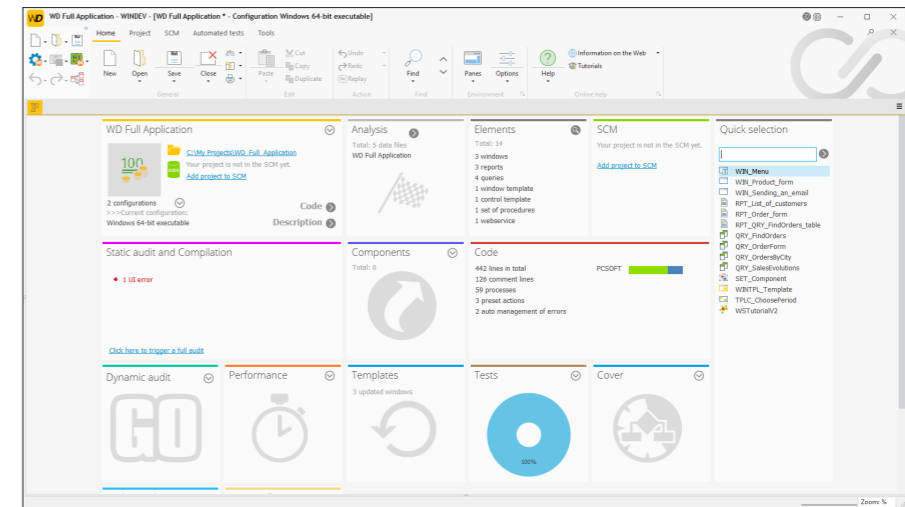
- Crop.
- Resize.
- Symmetry, rotation.
- Text.
- Layers.
- Access to image catalog.
- Transparency.

# Project dashboard

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

The project dashboard includes several indicators about the project content:

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



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

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

# WINDEV/WEBDEV/WINDEV Mobile: 100% compatible

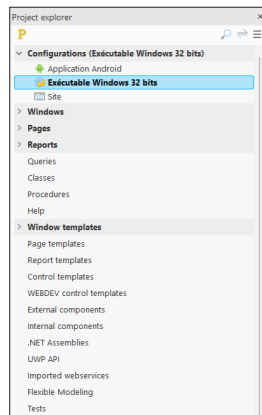
New projects are often cross-platform.

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

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

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

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



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

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

You can also view the elements of each platform from each environment. A project in WINDEV displays the thumbnails of the WEBDEV pages and 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).

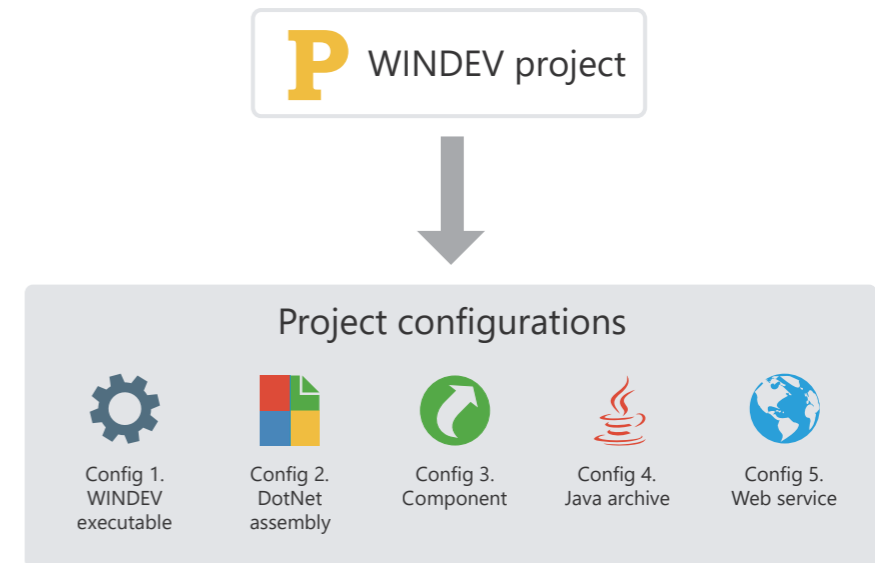
**Remark:** If you have WINDEV and WINDEV Mobile, you can handle WINDEV Mobile configurations from WINDEV. You can open, modify and test mobile windows.

# Project configuration

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

From the same project, you can create:

- executables with different elements, names, etc.
- different components.
- Webservices.
- .Net assemblies.



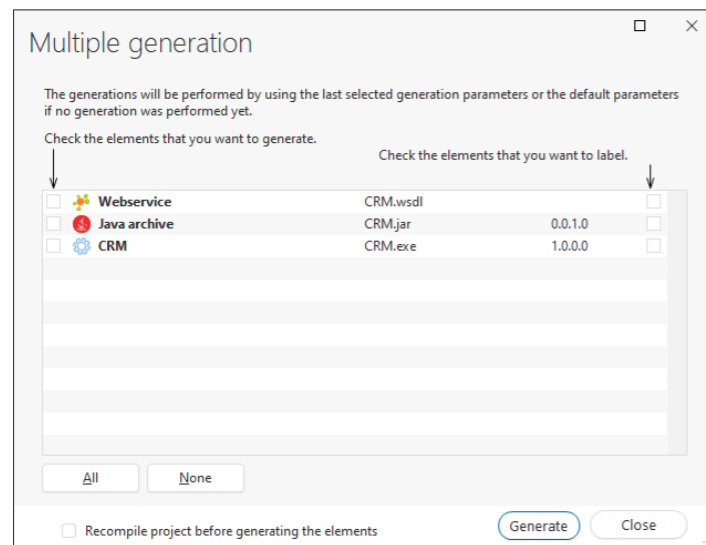
The multiple generation allows you to generate all project configurations (or some of them) in a single operation.

# Multiple generation

Project configurations allow you to easily define the "target" platforms of the project. Multiple executables, components and libraries can be defined for the same project.

You can select each configuration individually to generate the corresponding program.

There is a faster method: the **multiple generation**. You can select all the configurations to be generated at the same time to obtain the corresponding programs immediately.



To start a multiple generation, go to the "Project" tab, "Generation" group and click "Multiple generation".

# Custom-Folders: Organize your project

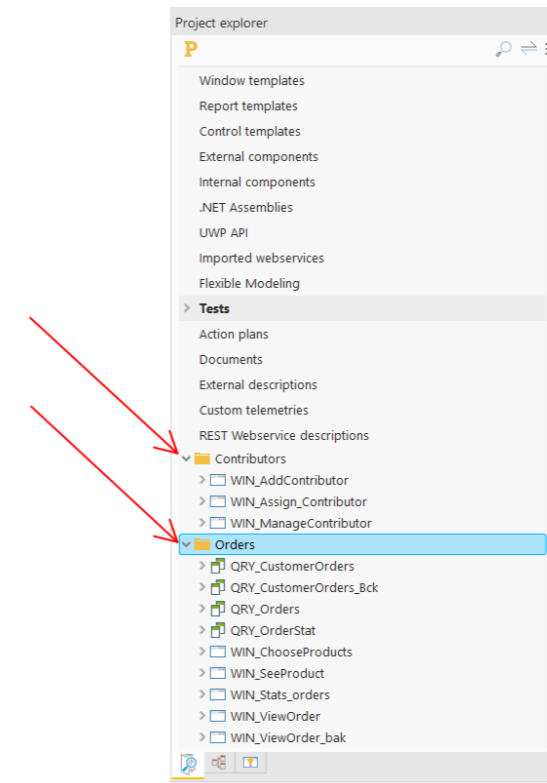
All elements included in a project are listed in the "Project explorer" pane. By default, the elements are organized according to their type: windows, reports, classes, etc.

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

To do so, create the "Custom-Folders" in the tree structure of the "Project explorer" pane and drag the different elements onto these folders.

Some elements can be common to several "Custom-Folders".

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



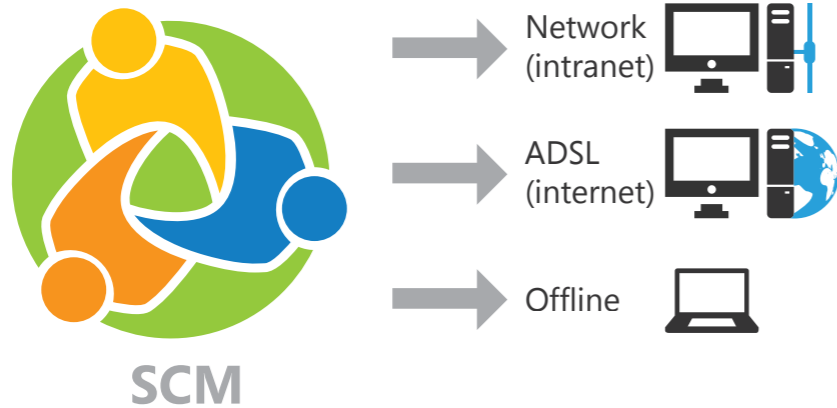
**Remark:** WINDEV also allows you to create internal components to organize your projects and share sets of elements between different projects.



# Source Code Manager (SCM)

## Overview

To simplify teamwork, WINDEV includes a Source Code Manager. The Source Code Manager allows developers to work together on the same project at the same time and share elements between multiple projects.



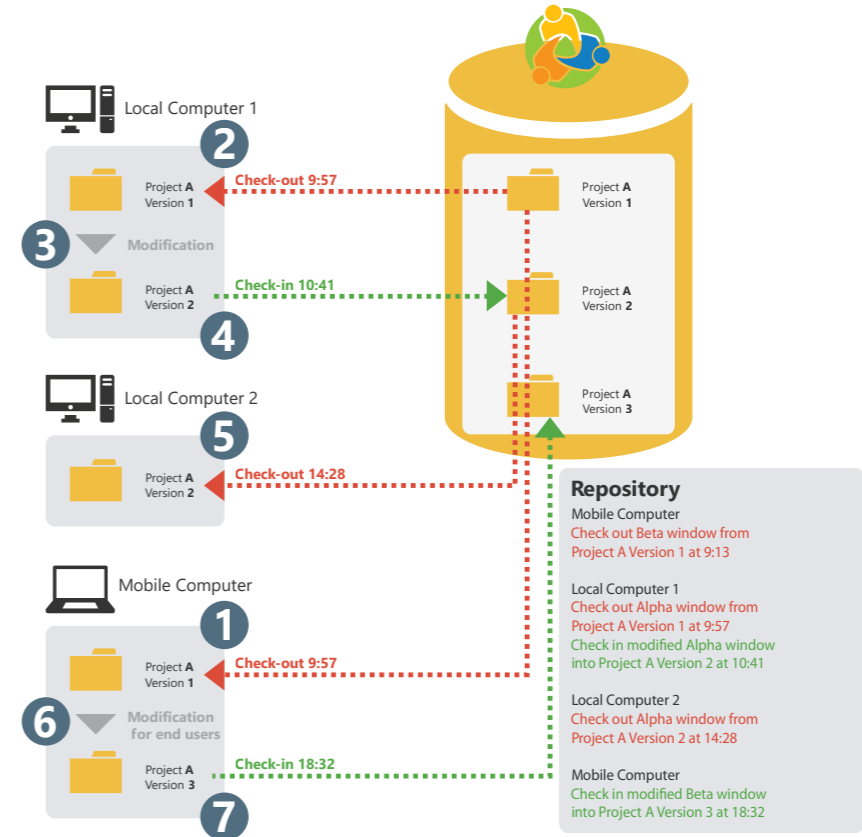
A repository contains a collection of project elements. Each computer has a local copy of the elements required for development.

Elements in the SCM can be shared:

- via PCSCloud (paid turnkey solution, on a dedicated server),
- via SCMDrive (paid turnkey solution, on a shared server).
- via a network,
- via the Internet,
- in offline mode. In this case, the necessary elements will be checked out from the SCM when the laptop is connected to the main system, for example.

## Operating mode of the Source Code Manager

The following example presents the Source Code Manager:



A project element (window, report, etc.) can only be checked out by one developer at a time. Once the checked out elements are modified, they must be checked back in so that the source project can take the changes into account. The repository stores a history of all the project elements since they were created.

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

# Sharing a project in practice

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

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

## 1 Source Code Manager

### 1.1 Overview

WINDEV boasts innovative features for teamwork management in the Source Code Manager (also called SCM).

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

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

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

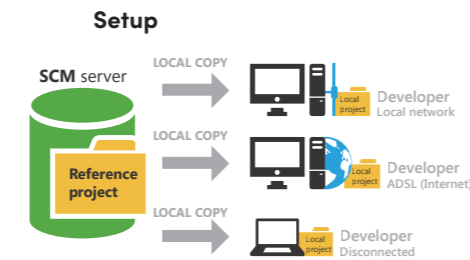
**The SCM uses a repository:** procedures, classes, pages, reports, components, analyses, etc.

You can choose between a repository:

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

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

### 1.2 Principle



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

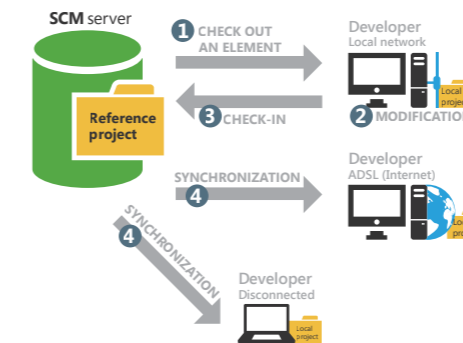
Each developer who uses the Source Code Manager retrieves a local copy of the project.

### Use

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

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

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



### Tips

The sources of your applications are essential. These sources must be handled with great care! Tips for configuring the server that will host your sources:

- Use a dedicated server with a disk with enough space (at least 200 GB).
- Use the Source Code Manager (SCM) in Client/Server mode.
- The hard disks may encounter physical problems: use a RAID I system on your server (several disks that store the same information).
- Use a UPS to protect the power supply of your server.
- Make backup copies of the repository on a regular basis (at least once a week).
- Place the server in a "secure" area and use a firewall.

## 2 Creating a repository for the SCM

### 2.1 Overview

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

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

The repository can be created:

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

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

### 2.2 When to create a repository?

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

WINDEV allows you to create this repository at different times:

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

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

### 2.3 Backups

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

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

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

#### 3.1 Adding a project to the SCM

To add a local project to the SCM:

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

- An existing repository. Simply select the desired repository.
4. Validate the wizard. The project is added to the SCM.

#### 3.2 Sharing resources

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

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

### 4 Working with the SCM

#### 4.1 Project options related to the SCM

Several options are available to configure a project handled by the Source Code Manager. These options can be found in the "SCM" tab of the project description ("Project" tab, "Description"). These options are checked by default.

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

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

#### 4.2 Checking out an element

##### The different check-out modes

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

- **standard mode:** if you open an element that has not been checked out from the SCM, a panel indicates that the element must be checked out to modify it. The element can be checked out immediately (check-out button found in the dialog box).

- **automatic mode:** if you try to modify an element that has not yet been checked out, the SCM automatically prompts you to check it out. Once the check-out is validated, the element can be modified.  
Remark: this mode is not recommended when using the SCM with a slow Internet connection.

To change the check-out mode:

1. On the "Home" tab, in the "Environment" group, expand "Options" and select "General options of WINDEV".
2. In the "General" tab, the "" option allows you to switch all subsequent projects to automatic mode.

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

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

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

#### 4.3 Checking an element back in

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

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

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

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

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

#### 4.4 Project management modes

Two management modes are available with the SCM:

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

##### Master and guest

The Source Code Manager distinguishes between 2 types of users:

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

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

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

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

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

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

**Caution: Changing the project options:**

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

**Automatic mode**

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

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

**5 Working in offline mode with the SCM**

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

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

The principle is simple:

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

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

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

For more details, see the online help.

**6 SCM administrator**

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

It allows you to:

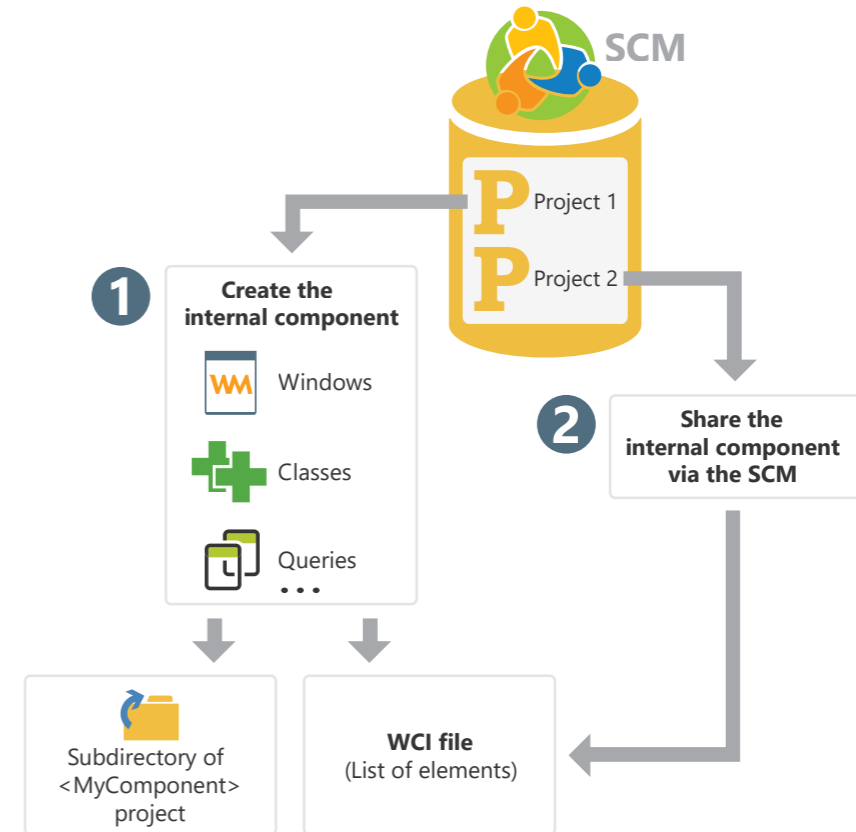
- manage repositories (create or connect to a repository).
- manage branches.
- manage files and directories located in a project in the repository (add, delete, rename files and directories).

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

# Internal component

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

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



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

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

# Internal components in practice

## 1 Overview

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

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

One of the advantages of an internal component compared to an external component is that it can be debugged directly from the project that uses it.

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

When using an external component, the "public" elements of the component cannot be handled directly. To modify the external component, you need to open the corresponding project.

## 2 Creating an internal component

### 2.1 Steps to follow

**To create an internal component:**

1. On the "Project" tab, in the "Project configuration" group, expand "New configuration" and select "Internal component". The internal component creation wizard starts.
2. Specify the characteristics of the internal component:
  - its name. The name of the internal component will be used for the WCI file that corresponds to the description of the internal component. This name will also be used to create a subdirectory in your project with all the elements of the internal component.
  - its caption.
  - its description.
3. Select the elements that will be included in the internal component. An internal component can contain all types of elements from a project: windows, reports, templates, sets of procedures, etc.

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

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

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

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

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

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

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

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

### 2.2 Internal component and analysis: total autonomy

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

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

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

## 3 Sharing internal components (via SCM)

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

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

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

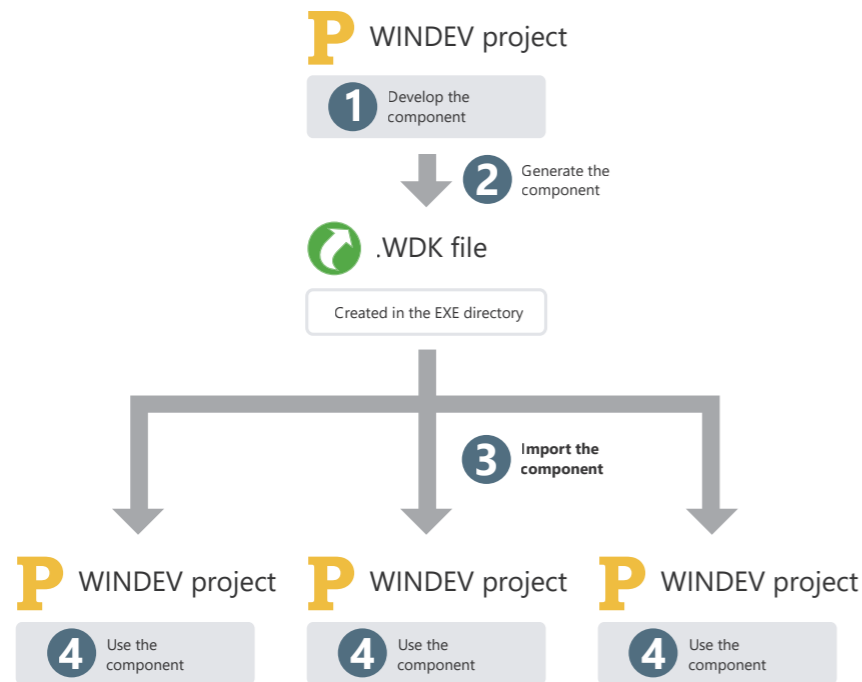


# External component

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

- Sending SMSs,
- Sending emails,
- etc.

A WINDEV external component can be distributed to other WINDEV developers (free or not). Developers will be able to easily include the functionalities of the external component in their applications. The external component will be included in the application and distributed with the application.



## External component in practice

### 1 Overview

#### 1.1 Definition

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

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

#### Remarks:

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

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

When creating the external component, the author specifies:

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

**Remark:** WINDEV includes several examples of components. These components can be imported via the "Project" tab of the editor: on the "Import" tab, in the "Project" group, expand "Project" and select "An External component .. Utility component or example". You can open the corresponding projects from the WINDEV home page (Ctrl + <).

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

An external component can be:

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

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

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

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

The external component consists of several files:

<Component name>.WDK

**Component file.** Contains all the elements of the external component.

This file is required to include the external component in a project. This file must also be provided in the client version of the application that uses the external component.

<Component name>.WDI

**Component description** (when developing applications only).

This file is required to include the external component in a project. This file **must not** be provided in the client version of the application that uses the external component.

<Component name>.WDZ

**File with the dependencies of the WDK file** (only for WEBDEV and WINDEV Mobile components)  
File containing the dependencies of the WDK file. This information is extracted in the project in which the component is integrated.

- For a WEBDEV component, this file contains the generated WEBDEV pages, images, etc.
- For a Mobile component (iOS or Android), this file contains the Android generation files, etc.

<Component name>.WDO

**Optional file**  
Text file containing the list of the additional elements provided with the external component (data files, .INI files, etc.). This file can be created manually or generated automatically from the items specified in the "Other" folder of the "Project explorer" pane. For more details, see WDO file.

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

An external component contains:

- the elements to be distributed (windows, reports, etc.). These elements can be accessible (or not) when importing the component into a WINDEV, WEBDEV or WINDEV Mobile project. These elements cannot be modified from the project that uses the component.
- a short description of the component.
- a help text to reuse the component. This help is generated from the code comments.

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

## 2 Creating and generating an external component

Below are the steps to create an external component:

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

### 4. Generating the external component.

The external component can be distributed and reused. For more details, see Distributing an external component and Using an external component in an application.

## 3 Distributing an external component

### 3.1 Overview

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

- **Direct use of the external component**  
Example: the external component is directly used on the computer of the developer who created it.

- **Standard distribution**, providing the necessary files.  
Example: the external component is intended to be used within the same company, by several developers. In this case, the necessary files can be copied to a network drive.

- **Distribution via the SCM:** This distribution mode allows all projects in the SCM to directly use a component published in the SCM. For more details, see SCM and components.  
Example: A company shares several projects via the SCM. The projects used to create the external components and the projects that use the components share the same SCM.
- **Distribution via a setup program** (with a setup procedure of the component similar to the one used for the applications).  
Example: This distribution mode is recommended if the external component is intended to be sold or distributed on a large scale, with regular updates.

### 3.2 Direct use of the component

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

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

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

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

### 3.3 Basic distribution

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

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

To distribute an external component, you must provide:

- The files automatically generated by WINDEV or WEBDEV (<Component name>.WDK and <Component name>.WDI files)

- If necessary, the specific files handled by the component (data files, initialization files, etc.) as well as the <Component name>.WDO file. The <Component name>.WDO file contains the list of files that must be provided with the component. For more details on this file, see Distributing an advanced component: WDO file.

### 3.4 Distribution via a setup procedure

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

This installation mode allows you to automatically manage:

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

To create a setup procedure for an external component:

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

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

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

## 4 Using an external component in an application

A component can be reused at any time in any WINDEV application: simply import the directory of the component into the project. The component can also be imported via the SCM.

When a new version of the component is available, you just have to install the new files in the setup directory of the component (depending on the setup mode used). For more details on the different modes to install a component, see the online help.

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

The executable (that uses the component) is created in the same way as a WINDEV application. The DLLs required for the component are automatically detected.

## 5 Deploying an application that uses an external component

### 5.1 Overview

An application that uses one or more components is deployed in the same way as a standard application ("Setup procedure" on the "Project" tab).

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

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

### 5.2 Updating components and deployed executables

When updating a component and/or an executable, the following elements can be installed on the end-user computer:

- the application executable, the component and the necessary files.
- the executable and the files required for the application.
- the component and any necessary files.

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

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

#### 1. Recompiling the host project

Recompile the application that uses 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 to the component and must be taken into account.
- The parameters of some procedures have been modified.
- Incompatibility between new and earlier versions of the component.
- ...

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

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

In most cases, this possibility applies when:

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

## 6 Modifying an external component

### 6.1 Overview

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

- add elements to the component.
- remove elements from the component.

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

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

## 6.2 Types of compatibility

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

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

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

## 7 Advanced component characteristics

### 7.1 Automatic documentation

External components must include a technical documentation.

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

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

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

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

- Comments at the beginning of the WLanguage procedures.

```
Global procedure gDefineChartSettings
// Procedure used to define the parameters
// gDefineChartSettings(sChartName, sName)
PROCEDURE gDefineChartSettings()
// These parameters are contained in a file
// Define the default parameters if the file
bHPattern is boolean = True
```

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

#### When is the documentation generated?

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

When generating the documentation:

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

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

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

#### How to access the component documentation?

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

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

- when a component is included in a project: on the "Project" tab, in the "Project" group, expand "Import" and select "An External component".
- in the description of the components included in the project. In the "Project explorer" pane:
  - In the "External components" folder, select the desired external component.

- Right-click to open the context menu and select "Description".
- In the "Details" tab, you can get the details of the generated documentation.

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

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

### 7.2 Component element visibility

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

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

**Remark:** The code of this element is not visible.
- If the element is not accessible, the user will not even know that it exists.

**Caution:** Depending on how the project elements are declared (classes, sets of procedures, etc.), the elements that are accessible may change.

### 7.3 WDO file

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

<Component name>.WDK

Contains all the WINDEV elements to be redistributed with the component (windows, reports, etc.).

<Component name>.WDI

Component interface. This file contains:

- a help text on how to use the component when it is reintegrated,
- the elements required to use the component in the project (compilation information, etc.).

<Component name>.WDZ

**File with the dependencies of the WDK file** (only for WEBDEV and WINDEV Mobile components)

File containing the dependencies of the WDK file. This information is extracted in the project in which the component is integrated.

- For a WEBDEV component, this file contains the generated WEBDEV pages, images, etc.
- For a Mobile component (iOS or Android), this file contains the Android generation files, for example.

These files must be distributed along with the component.

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

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

#### What is a WDO file?

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

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

This ".WDO" file can contain:

- **the full name of the file.**  
For example: C:\ComponentsWD\PickerComponent\InitialStatus.INI
- **the file name.** This file will be searched in the current directory of the external component.  
For example: InitialStatus.INI

- **a file name that uses a relative path.** The available syntaxes are as follows:
  - Directory\FileName.xxx to specify a subdirectory of the current directory.
  - \FileName.xxx to specify the current directory.
  - \FileName.xx to specify the parent directory.
 For example: \PickerComponent\InitialStatus.INI  
 This file will be used when including the external component in the project. The paths specified in the WDO file must correspond to the paths where the files are installed on the development computer of the external component.

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

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

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

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

#### Distributing a component with a WDO file

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

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



# Generation modes

In addition to Windows applications, WINDEV allows you to generate other types of projects.



## Applications

Applications are the most common generation mode. Applications developed with WINDEV can run on the following platforms: Windows (32 or 64-bit) and Linux.



## Services

WINDEV allows you to generate services for Windows (32 or 64-bit) and Linux (in Linux, services are usually called daemons). A service is a specific type of application, started at the same time as the operating system but without an interface.



## Libraries and patches

A library is a single file that contains multiple elements of a WINDEV project: windows, reports, etc. You can generate stand-alone libraries that can be used by other applications as well as patches for an application that is already deployed. This allows you to avoid reinstalling the entire application for a minor correction.



## External components

External components are application snippets that allow you to share one or more specific functionalities between multiple 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 or WEBDEV project. A Webservice uses a set of functions (one or more sets of procedures) and makes them accessible via the web (or a private network) using HTTP, SOAP and REST.

Remark: A Webservice must be deployed to a WEBDEV Application Server 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 shared between two WINDEV applications.



## Java applications

WINDEV allows you to generate Java applications. 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 run on Windows, Mac OS or Solaris, for example.



## Universal Windows Platform applications

WINDEV allows you to generate Universal Windows Platform applications. These applications can be used in tile mode or be distributed via the Windows Store.



## Creating the executable in practice

When your application is complete (windows, reports, etc.), you must create the corresponding executable file.

The ".EXE" file can be run directly : there is no need to start the application from WINDEV.

Once the executable file has been created, the application can be deployed on the end-user computers via the setup program

### 1 Creating the executable file of a WINDEV application

To create the executable file:

1. Select (if necessary) the project configuration that corresponds to the executable to be created:

- Project configuration for a 32-bit Windows executable file.
- Project configuration for a 64-bit Windows executable file.
- Project configuration for a Linux executable.

2. On the "Project" tab, in the "Generation" group, click "Generate".

3. The executable creation wizard starts.

4. Specify:

- the **general options** of the executable (executable name, icon of the executable file, splash screen, etc.).
- the **operating mode** of the executable:
  - Multi-instance: The program can be started multiple times from the same computer: multiple instances will be opened at the same time.
  - Single-instance: The program can be started only once from the same computer: a single instance will be opened at a time. If the user tries to run another instance of the program, the current instance will be enabled.
- **how the library will be used.** The library is optional. It can be created in a separate file (".WDL" extension) or included in the executable. The advantage of including the library in the executable file is that you only need to install or update one file. However, this would increase the size of the ".EXE" file.

- the **error message** that will be displayed if a programming error occurs.
  - if the **telemetry** will be enabled. Telemetry allows you to get information on how your application is being used.
  - the use of **patches for updates**. In this case, you just have to copy the patch file to the executable directory.
  - the languages supported by the executable.
  - the **files to be included in the library** (if it is used).
  - whether the components used in the project are included in the executable.
  - **the location of the User Groupware data files** (if the application uses the User Groupware)
  - the **location of the data files** that will be used by the executable.
  - the **version information**. This information will be available in the "Version" tab of the executable properties window (accessible in the Windows explorer). The **version number** can be automatically **incremented** each time the executable is created.
  - the mode for using the WINDEV framework (framework specific to the application or common to all 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 created.
5. Validate the executable description. The executable is automatically generated based on this information. The executable can be directly run once it has been created.

## Java applications in practice

### 1 Overview

WINDEV allows you to generate Java applications without previous experience in Java development. 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 uses a database, this database must have a JDBC driver (MySQL, Oracle, etc.). This driver is specific to each database (it can be downloaded from the Web site of the database publisher).

HFSQL databases are accessed natively (in Windows and Linux only) via the SQL WLanguage functions. No additional driver is required.

A Java application corresponds to a ".jar" file. It requires the Java Runtime to be executed.

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

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

You have developed a WINDEV application that runs on Windows and you want to make it accessible on any operating system? Generate a Java application so that Mac OS users can use your application, for example.

This allows you to share network applications.

#### 1.3 Generating a Java application with WINDEV


As mentioned above, you don't need to have previous experience in Java development to develop a Java application with WINDEV. You can develop your applications in WLanguage like any other WINDEV application and, when compiling the project, the windows and WLanguage functions are automatically converted to Java.

**Remark:** 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 forthcoming versions.

### 2 Features of a Java project

#### 2.1 Creating a Java project

A Java project can be created like any WINDEV project:

1. Click  in the WINDEV quick access buttons. The new element window appears: click "Project".
2. The project creation wizard starts.
3. In one of the first steps, the wizard allows you to choose the project generation mode. Select "Java archive" and continue.

By default, a WINDEV project for Java corresponds to a specific project configuration. The type of project generation can be changed at any time. Simply create a new project configuration. Once the project has been configured as a Java project, all the tools available in WINDEV are automatically configured for the Java project:

- The window RAD or the full 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.
- The creation of the archive makes it possible 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.

**Remark:** it is recommended to create a new WINDEV project to develop a Java application rather than converting a WINDEV application to Java.

### 2.2 JAVA RAD


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

The RAD methodology 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 allows you to enter the type of connection to the database:

- HFSQL (in Windows or Linux): the standard window creation wizard appears.
- JDBC: you can specify the parameters for connecting to the database that uses the JDBC driver.

### 2.3 Test a Java project in the editor

When a Java project is tested (  ), the project is compiled and run in Windows. The project is not compiled in Java.

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

### 2.4 Compiling a Java project in WINDEV

A Java project is compiled to:

- identify programming errors in the WLanguage code
- see controls, events and WLanguage functions that are not available in Java. These errors appear in purple in the "Compilation errors" pane.  
For more details on WINDEV elements not available in a Java application, see the online help.

**Remark:** as mentioned in the What is a Java application? section, 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 Web Start application.

To create the Java archive, all you have to do is create the executable: on the "Project" tab, 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 Java applications. This setup procedure is available for Windows only. The setup program is an executable. It can be used to install the necessary files as well as the Java Runtime Environment (JRE). For other platforms, you need to deploy your files manually.

To start the setup creation wizard, go to the "Project" tab, "Generation" group and click "Setup procedure".

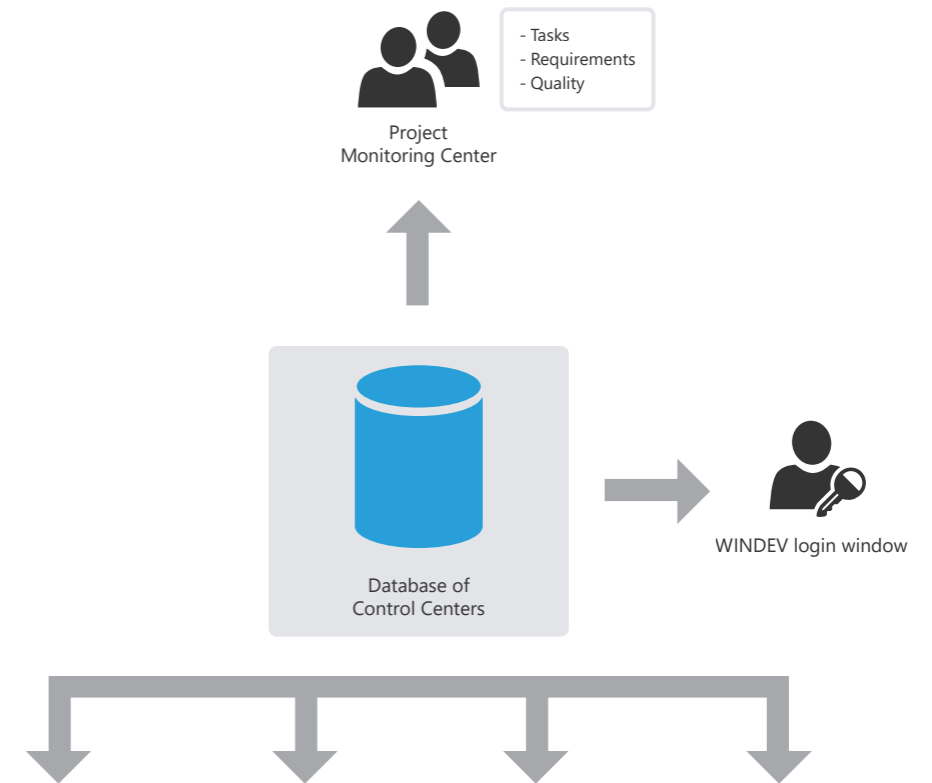
For more details, see the online help (keyword: "Java").

# Project Monitoring Center

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

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

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



**Remark:** When installing WINDEV, the setup program prompts you to:

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

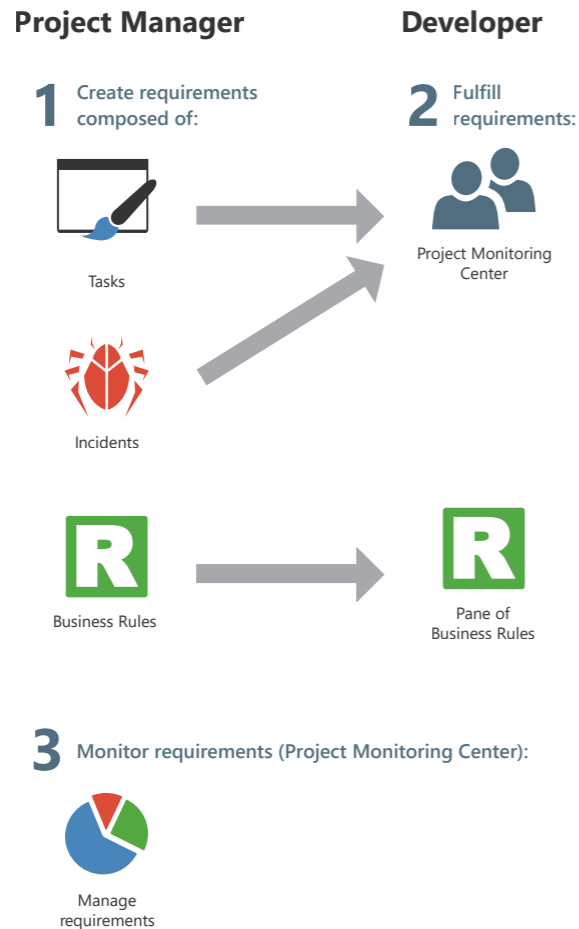
# Managing requirements

The Project Monitoring Center allows the project manager to manage a development project. To do so, you must:

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

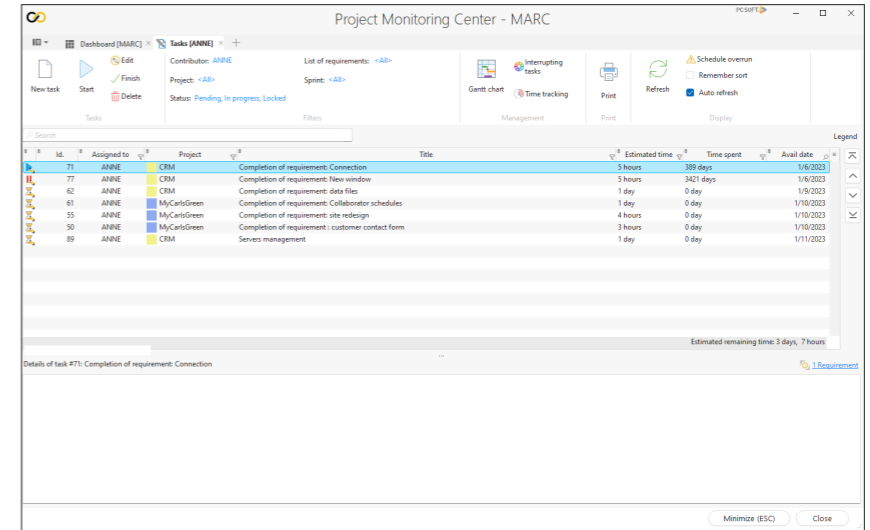
Each developer performs the different tasks assigned to them.

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



# Managing tasks

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



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

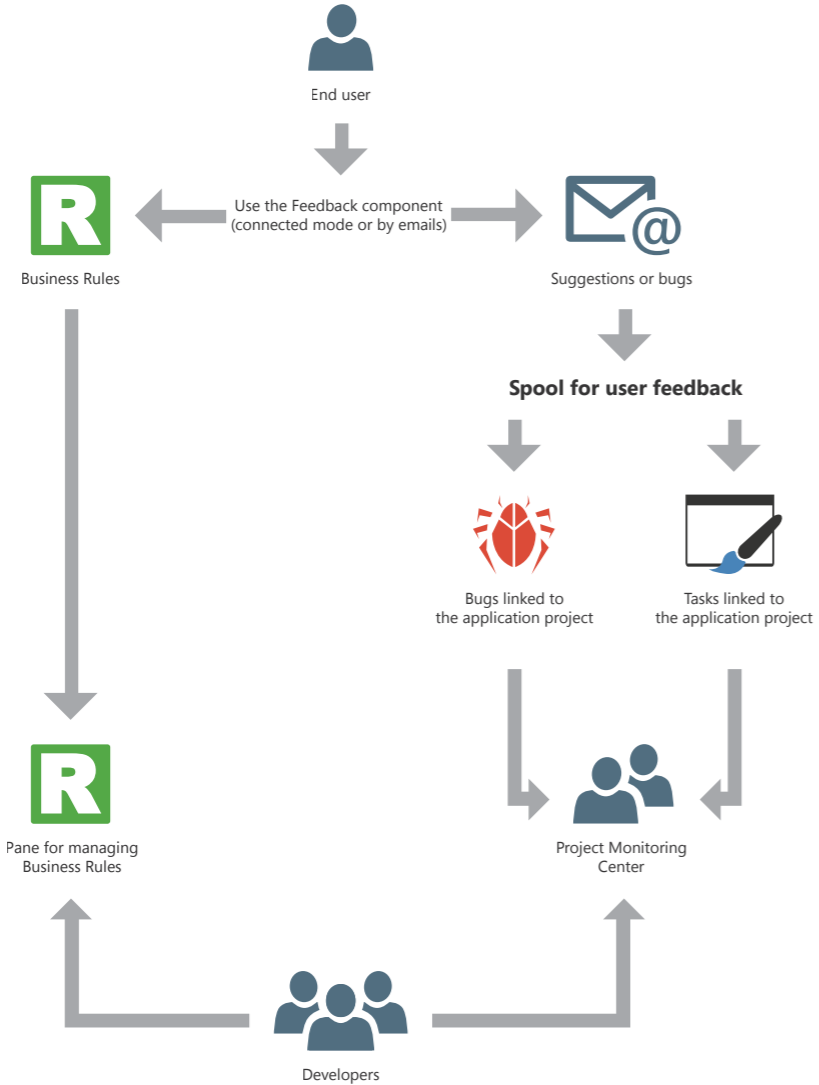
When the relevant project is opened, the Project Monitoring Center requests or indicates the current task. As soon as a task is completed, simply indicate that the task is finished and specify the new task.

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

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

# Managing suggestions and incidents

The Control Centers also allow you to manage user feedback on 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.



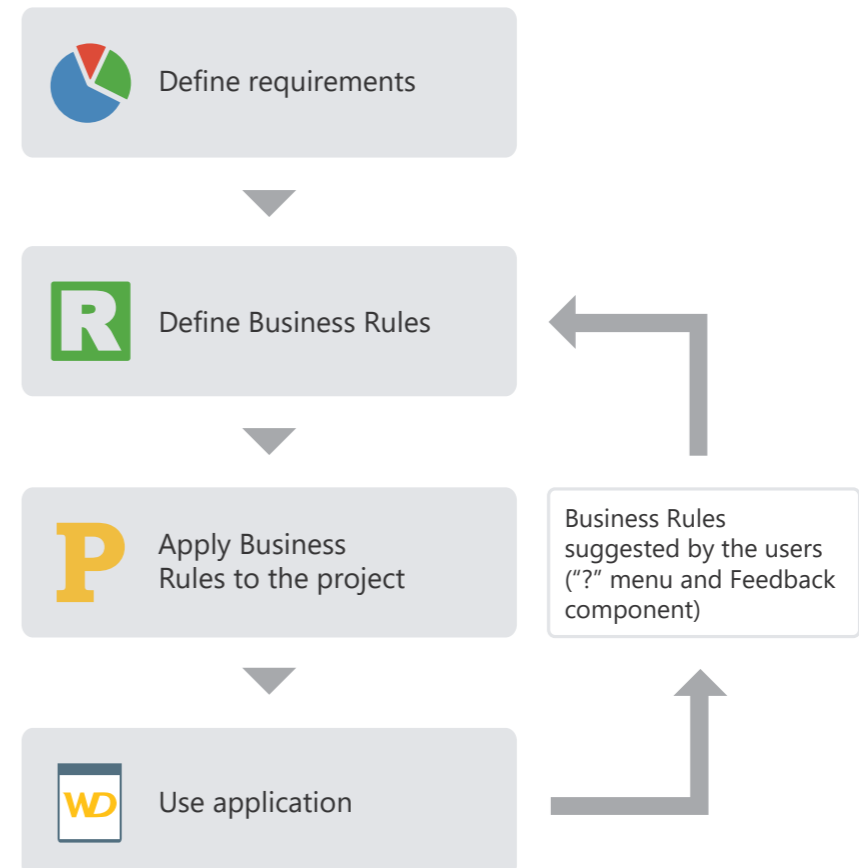
**Remark:** if a fatal error occurs when the application is running, an error report can be automatically generated and sent to the developer via the "Feedback" component provided with WINDEV.

# Managing business rules

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

A business rule can be simple or complex.

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



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



PART 3

Databases

28

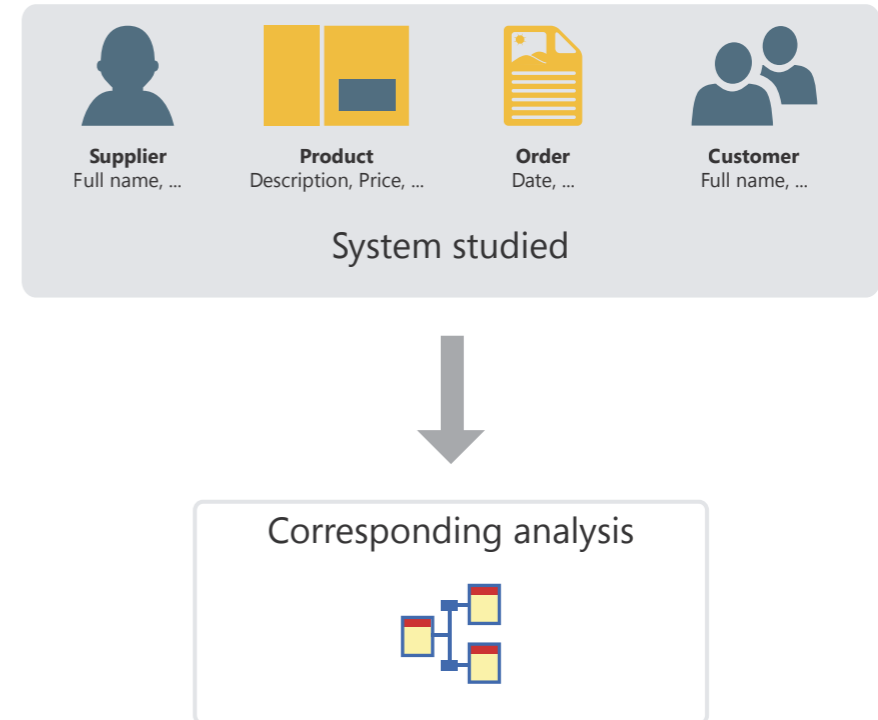
DEVELOP 10 TIMES FASTER



# Analysis: Database structure

When a WINDEV, WEBDEV or WINDEV Mobile project uses data files, it must be associated with an analysis. An analysis is used to describe the data structures (data files, items, etc.) used in your project.

The data model editor allows you to easily create an analysis.



The analysis of a WINDEV, WEBDEV or WINDEV Mobile project corresponds to the LDM (Logical Data Model). The analysis contains the entire structure and data organization: data is grouped into files (data files). Each file contains multiple data fields, called items.

In the analysis, the description of a data file can be linked to a type of file (HFSQL Classic, HFSQL Client/Server, SQLite, etc.).

# Analysis in practice

## 1 Overview

When a project uses data files, it must be associated with an analysis. An analysis describes the structure of the data (data files, items, etc.) used in your project.

There are two methods to describe an analysis:

- **1st method:** Creating the analysis directly (which means a Logical Data Model (LDM)).

- **2nd method:** Creating the Conceptual Data Model (CDM), then automatically creating the analysis from the CDM. For more details, see the online help.


This chapter presents the first method.

## 2 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  in the quick access buttons. The creation window appears: click "Data" then "Analysis". The analysis creation wizard starts.
2. Specify the general details of the analysis:
  - **the analysis name and directory.** The analysis corresponds to a ".WDA" file. By default, this file will be created in the directory of the project analysis (<Project name>.ANA directory). This directory must be accessible in read/write mode.
  - **the analysis description** to briefly explain what the analysis is about.
3. In the next step, select the type of database:
  - Database HFSQLDrive (paid turnkey solution, on a shared server).
  - HFSQL database in PCSCloud (paid turnkey solution, on a dedicated server).
  - HFSQL database (Client/Server or Classic).
  - Database accessible via a Native Connector.
  - Other types.
4. In the next step, specify if necessary the connection information.
5. Validate the analysis creation wizard. A wizard to create the first data file appears automatically.
6. Create all the elements of the analysis (data files, items and links).

### 2.2 Adding a data file to an analysis

The analysis defines the structure of the data files used by the project. In the analysis, a data file can be:

- A new data file.
- A preset data file, provided with WINDEV, WEBDEV or WINDEV Mobile.
- A data file imported from an existing database (using a specific format, for example).

The following paragraphs explain how to create a data file.

To create a new data file:

1. On the "Analysis" tab, in the "Creation" group, click "New file". The data file creation wizard starts.
2. Select "Create a new data file description".
3. Specify:
  - **the name of the data file.** This name is the logical name of the data file. It will be used to handle the data file.
  - **the caption of the data file** that briefly describes the file subject.
  - **the representation of a record** in the data file. This representation makes the questions asked when describing the links easier to understand. This option must be preceded by an indefinite article (A or AN).

- whether the data file includes an **"Automatic identifier" item**. The value of this item is unique for each record and is automatically calculated by WINDEV, WEBDEV or WINDEV Mobile.
- **the type of database associated with the data file:**
  - Database HFSQLDrive (paid turnkey solution, on a shared server).
  - HFSQL database in PCSCloud (paid turnkey solution, on a dedicated server).
  - HFSQL database (Client/Server or Classic).
  - Database accessible via a Native Connector.
  - Other types.

Depending on the type selected, this data file will be handled by the HFSQL engine, an OLE DB driver or one of WINDEV or WEBDEV's Native Connectors (SQL Server, Oracle, etc.).

4. The new data file becomes the current data file. The description window of the data file items opens automatically. It allows you to describe the file items.

To import a preset description of a data file:

1. On the "Analysis" tab, in the "Creation" group, click "New file". The data file creation wizard starts.
2. Select the option "Select a description among the preset data files".
3. Choose the preset data file to be created (you can select multiple data files). This data file will be imported into the current analysis. This data file can be modified later.
4. Select the items you want to keep. These items can be modified later.
5. The imported data file is automatically inserted into the current analysis.

**Remark:** To modify a data file or the items in a data file:

1. Select the data file.
2. Select "Data file description" or "Description of items" in the context menu.

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

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

1. On the "Analysis" tab, in the "Creation" group, click "New file". The data file creation wizard starts.
2. Select "Use the data files from an existing database" and select the type of database.
3. Specify the source database containing the descriptions to be imported and the type of database. Depending on the selected type, specify the requested information.
4. Select the tables or data files whose description must be imported and validate.
5. The imported data file is automatically inserted into the current analysis.

**Solution 2: From the Windows explorer**

Drag the data file description from the Windows explorer and drop it onto the data model editor.

### 2.3 Creating an item

To create an item:

1. Double-click the data file where the item will be created. The description window of the data file items opens.
2. Click the first empty row in the table.
3. Specify the name, caption and type of the item.
4. In the right section of the window, indicate the details of the new item (type, size, default value, iteration direction, etc.).
5. Specify the details about the shared information: click "Parameters of the control bound to the selected item (shared information)".
6. Validate the item description.

**Remark:** you can also create an item from the metatypes available in WINDEV. To do so, click the button. The list of available metatypes appears.

### 2.4 Creating a link

Different types of links can be created between data files. For more details on links, see Characteristics of the links defined in an analysis.

#### To create a link:

1. On the "Analysis" tab, in the "Creation" group, click "New link". The mouse cursor turns into a pen.
2. Select the two data files to be linked. The link description window opens automatically.
3. To define the cardinalities between the two data files:
  - select the desired cardinalities (0, 1; 1, 1; 0, N; 1, N).
  - answer the questions. The cardinalities will be automatically updated.

4. To define the advanced cardinalities, check "Display advanced cardinalities" and answer the questions.
5. Specify the link caption and briefly describe the purpose of the link.
6. Specify the keys to be linked.
7. Define the integrity rules. These rules ensure data integrity when one of the keys of the relationship is modified or deleted.
8. Validate. The link is automatically created.

## 3 Characteristics of the links defined in an analysis

### 3.1 Owner file and member file

When a link is defined between two data files, there is an **owner** data file and a **member** data file:

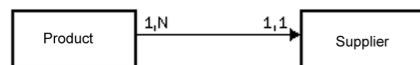
- the owner data file is the owner of the key.
- the member data file is a member of the analysis data files containing a copy of the key.

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

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

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

This type of link is represented as follows:



### 3.2 Cardinalities

Cardinalities are used to count the relationships between two data files.

Cardinalities are defined according to the answers to the two following questions:

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

The answer provides the first part of the cardinality (**minimum cardinality**):

- if the answer is "none", the cardinality is 0,X.
- if the answer is "a single one", the cardinality is 1,X.

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

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

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

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

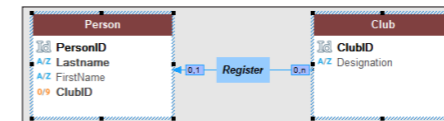
The description of cardinalities is essential: it is the basis for respecting the referential integrity of the database.

To put it simply, these two questions can be broken down into four questions. For example, to describe a link between the Club data file and the Person data file:

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

#### Example of cardinalities:

This example presents two different cardinalities:



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

### 3.3 Advanced cardinalities

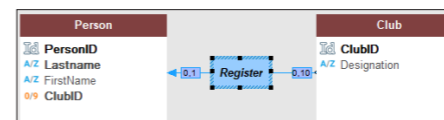
Cardinalities can be defined more precisely. This is referred to as **advanced cardinalities**.

These cardinalities allow you to specify exactly the minimum and maximum cardinality.

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

#### Example of advanced cardinalities:

This example presents two different cardinalities:



- **0,1 cardinality:** A person can be a member of a single sport club. This person does not have to belong to a sport club.
- **0,10 cardinality:** A club can have 0 to 10 members.

### 3.4 Referential integrity

Referential integrity in a database is a set of constraints that maintains consistency of data in a relationship between two data files.

Referential integrity checks that:

- if a record is deleted from the owner data file, the matching records in the member data files are also deleted,
- if a record is added to a member data file, there is a matching record in the owner data file,
- if a record is updated in the owner file, the unique key doesn't change, etc. The referential integrity is checked based on the nature of the link between the data files.

For more details on referential integrity, see the online help.

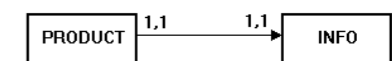
### 3.5 Types of links

There are different types of links between data files:

- parallel
- optional
- complement
- shared
- complex

#### Parallel link

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



In a parallel link, the identifier of the Product data file is copied to the Info data file. This identifier is also a unique key in the Info data file.

Product is the owner data file and Info is the member data file.

**Remark:** This type of link is rare because the two data files can be combined into one.

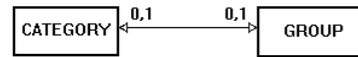
The parallelism of records in the data file is respected if the following operations are run simultaneously on the two data files:

- creation of a record.
- deletion of a record.
- reindexing with compression.

**Optional link**

For an optional link:

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

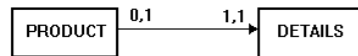


An optional link is established by copying the identifier from one data file to the other.

**Complement link**

For a complement link:

- **Each record** from a data file (Product) **has no matching records or has one matching record** in another data file (Details).
- **Each record** in the other data file (Details) **has a matching record** in the first data file (Product).



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

In a complement link, the key of the Product data file is copied to the Details data file. To ensure the maximum cardinality of 1, it remains a unique key. Since this key is unique, it is not possible to insert more than one record in the Details data file for a record in Product.

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

**Shared link**

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

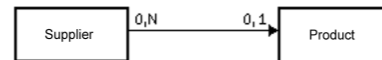
In a shared link, the key of the Supplier data file is copied to the Product data file. It becomes a multiple key to optimize the speed of the integrity check.

Supplier is the owner data file and Product is the member data file.

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

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

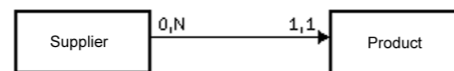
**Shared link with a 0,n - 0,1 cardinality**



With this type of link:

- an owner may have no member (a supplier does not necessarily have a product).
- a member may have no owner (a product does not necessarily have a supplier).

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



With this type of link:

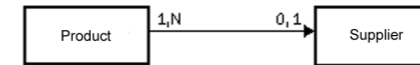
- an owner may have no member (a supplier may have no product).
- each member has a single owner (each product has a single supplier).

**Remark:** This type of link is quite common. WINDEV allows you to automatically create windows that manage data files with a 0,N - 1,1 relationship.

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

With this type of link:

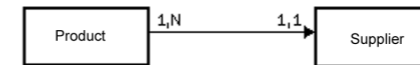
- each owner has at least one member (a supplier has at least one product).
- a member may have no owner (a product may have no supplier).



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

With this type of link:

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



**Complex link**

For a "complex" link, you need to define a **relationship file**.

The relationship file will be automatically created. It will contain a unique key that includes the two keys of the related data files.

The relationship file can also contain information specific to the link.

A complex link includes two shared links.

**Example of complex link**

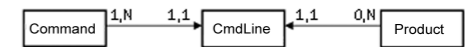
An order (Orders data file) can contain one or more products. A product (Product data file) can be used in several orders.

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

The OrderLine data file contains:

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

The links between the data files are as follows:



**4 Possible operations in an analysis**

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 a LDM in the editor.
- Move a LDM in the editor.

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

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

For more details, see the online help.

## 5 Generating the analysis (LDM)

The analysis is generated before the programming cycle and after describing the data files. This generation is used to:

- validate the changes made on the analysis (LDM).
- create the modules required for programming.
- automatically update the data files if necessary.

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

To generate the analysis, go to the "Analysis" tab, "Analysis" group and click "Generation".

The generation includes three steps:

- Verification of the changes made and generation of the physical files corresponding to the analysis description.
- Automatic modification of accessible data files (files in the "EXE" directory of the project).
- Project synchronization.

## 6 Managing the analysis versions

To manage the different versions of the analysis:

1. On the "Analysis" tab, in the "Analysis" group, click "Generation".
2. Select "Manage versions".
3. In the wizard, specify whether you want to work on the current analysis or on a specific analysis.

**Remark:** when an analysis is damaged, you can select the damaged analysis to restore one of the earlier versions, for example.

All the versions of the analysis are displayed as a graph. The yellow box indicates the version number of the analysis and the text on the right shows the generation date.

The "small rectangles" displayed between two versions indicate:

- one rectangle: few changes were made between the two versions.
- several rectangles: several changes were made between the two versions.

**To see the details of the changes made between two versions:** Double-click the row containing the "small rectangles" or click [Changes].

The options available in the version manager are as follows:

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

### 6.1 Restoring an analysis

To restore an analysis from the version manager:

1. Select the version to restore.
2. You can:
  - restore the version in the specified directory. An independent analysis corresponding to the selected version is created. This option is selected by default.
  - overwrite the current analysis.
3. Validate. The restore operation is performed.

**Caution:** The data files of an analysis whose version number is more recent than the restored version cannot be reopened. In this case, you must also restore the data files of the restored analysis version or delete the existing data files and create them again.

### 6.2 Undoing the last generation

Undoing the last generation of the analysis allows you to restore the analysis to its state before the last generation. All changes made since then are not applied.

### 6.3 Resetting the version number to 1

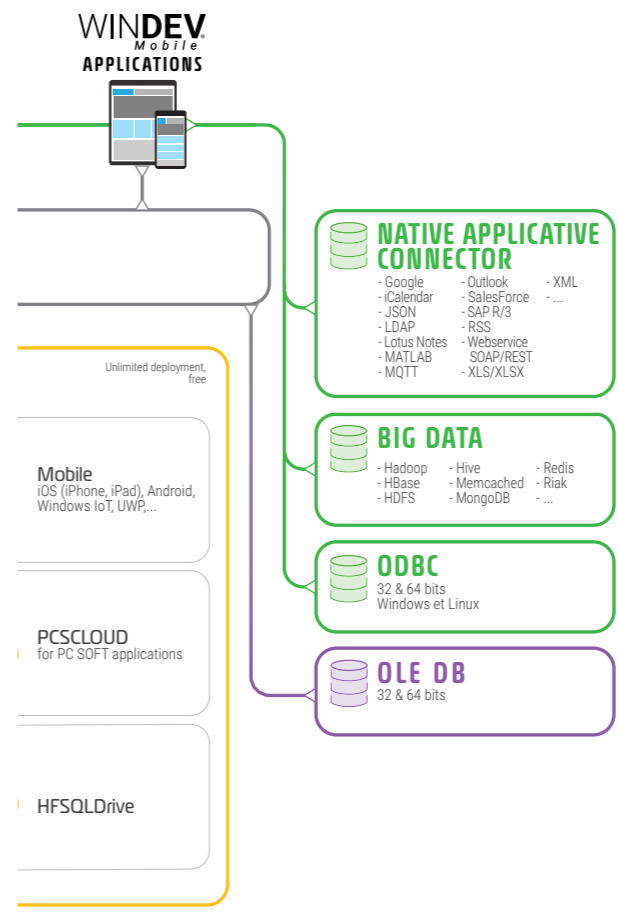
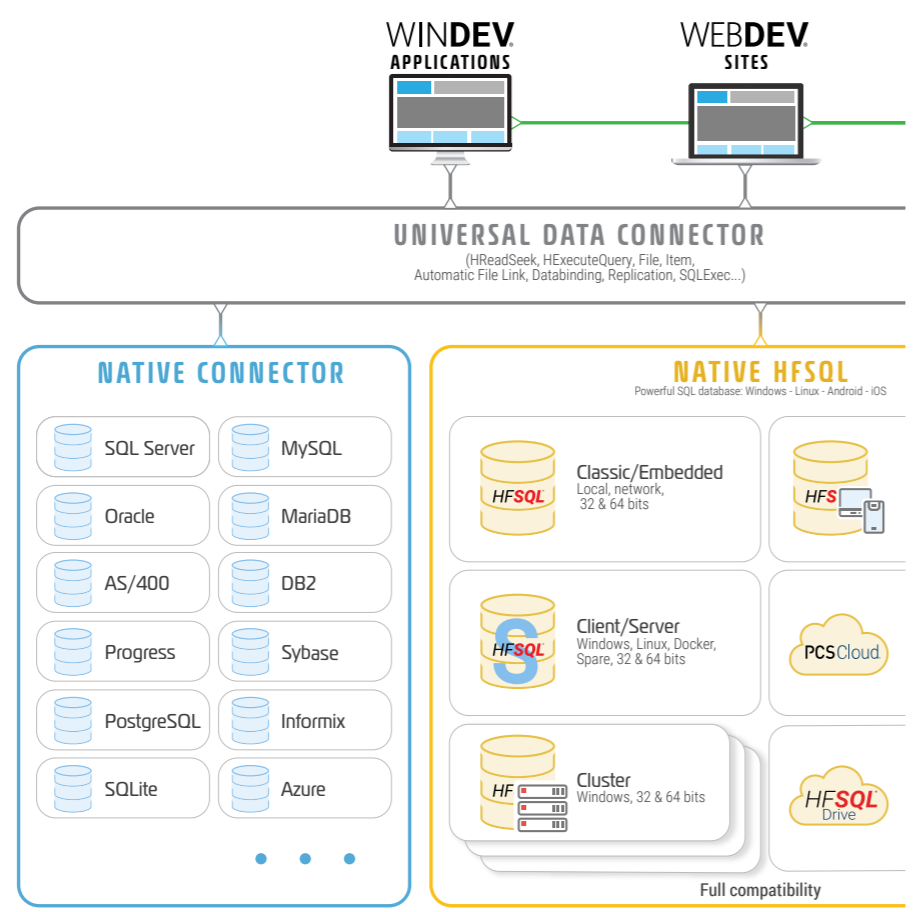
The analysis generation number can be reset to "1": on the "Analysis" tab, in the "Analysis" group, click "Generation" and select "Reset analysis version to 1".

In this case, the analysis version number is reset to one. No specific action is performed on the data files.



# Types of accessible files

WEBDEV, WINDEV and WINDEV Mobile offer simple access to the majority of databases on the market.

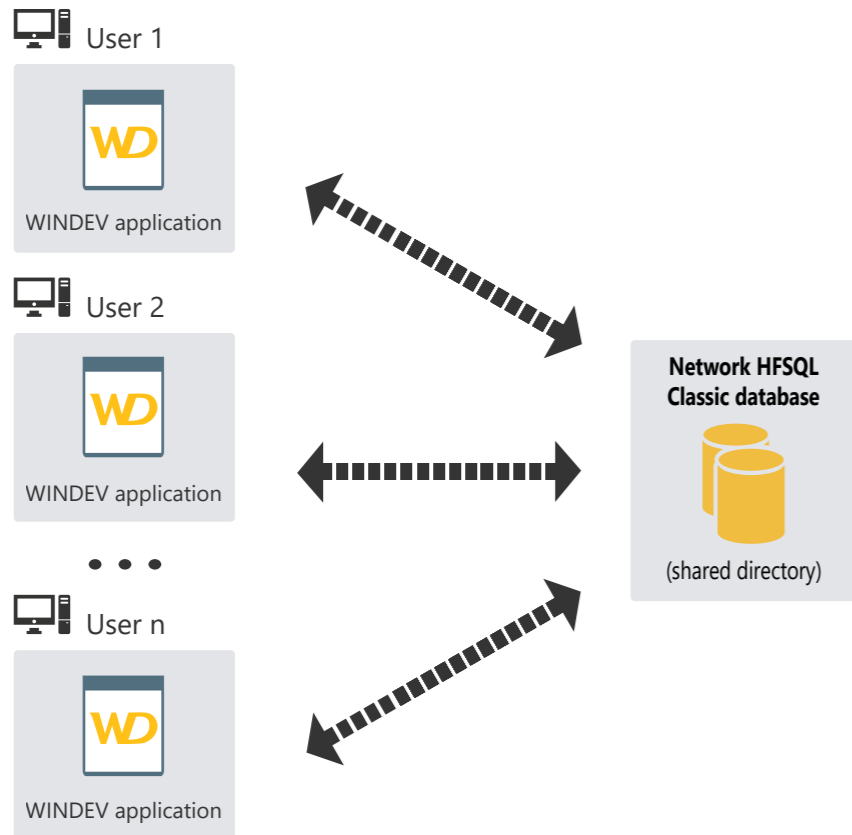


# HFSQL Classic

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:

- An HFSQL Classic application can be run on different client computers distributed across a local network.
- The data files are located in a directory that can be accessed by the client computers (a shared directory on the network, for example). Each client computer physically accesses the data files.
- All processes (requests, read from and write records to a data file, etc.) are performed on each client computer.

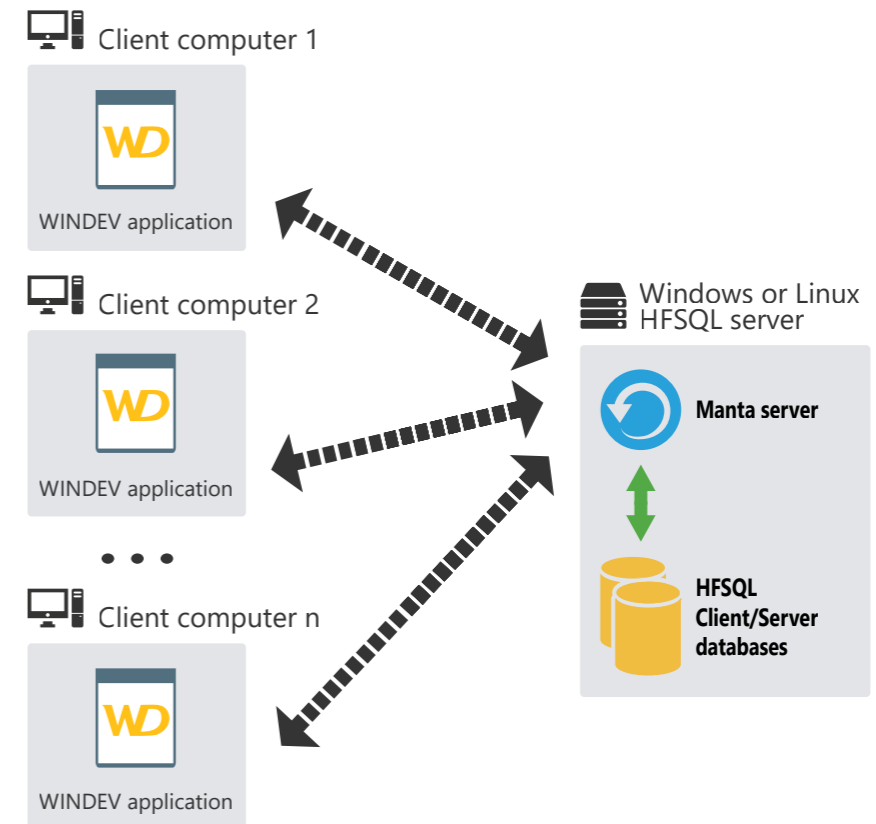


# HFSQL Client/Server

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:

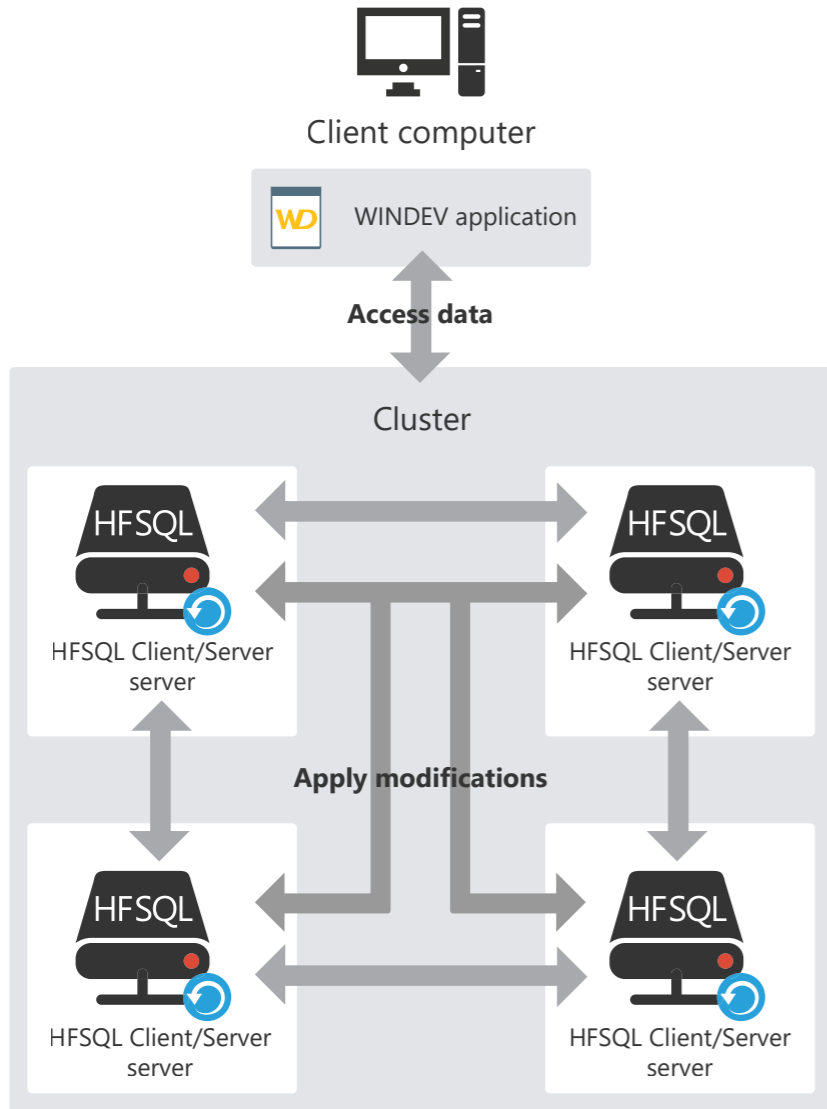
- An HFSQL Client/Server application can be run on different client computers that communicate with the database server over the network.
- Data files are located on a server. Only the server physically accesses the data files.
- All processes (queries, read from and write records to a data file, etc.) are performed on the server.



# HFSQL Client/Server cluster

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

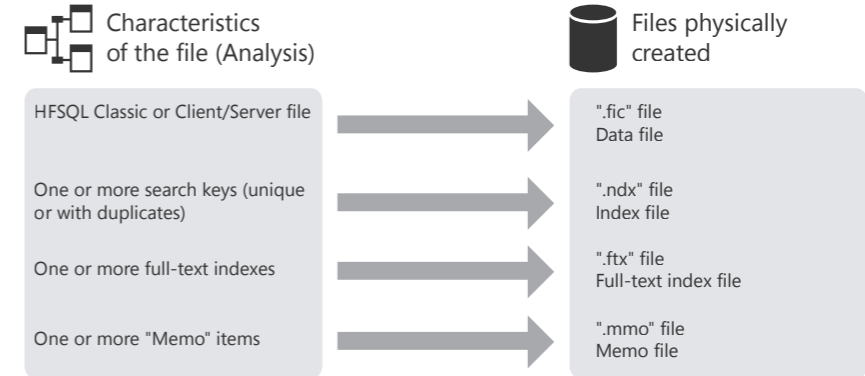
- The read load can be balanced across multiple servers.
- You can make changes to the physical configuration without interruptions on client computers.
- If one of the servers crashes, the client is automatically redirected to an operating server.



# HFSQL: files physically created

The data model editor allows you to define the structure of the data files.

Files are physically created depending on the information entered in the data model editor.



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

# Binding controls to data

A window can display information from:

- a database: controls are directly bound to the items in the data files or queries available in the database.
- variables in the application's WLanguage code (variables global to the window or to the project or parameters passed to the window).

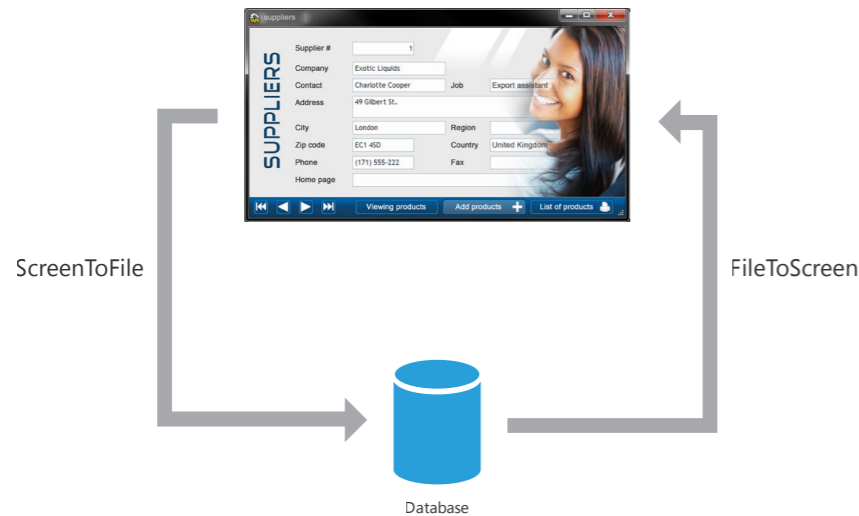
To display information in a window, the controls in the window must be bound to:

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

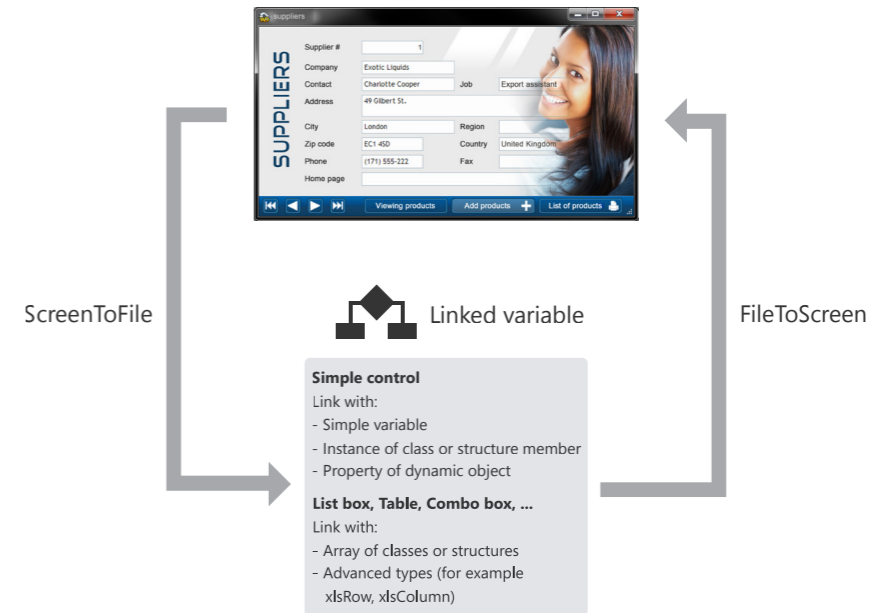
There are multiple methods to display and update values from a data source:

- The binding between a control and an item or variable can be defined in the window editor, in the control description ("Binding" tab).
- **ScreenToFile** updates the record or the variable with the values from the UI.
- **FileToScreen** updates the data in the UI with the values from the data file or variable.

## Control - item binding



## Control - variable binding

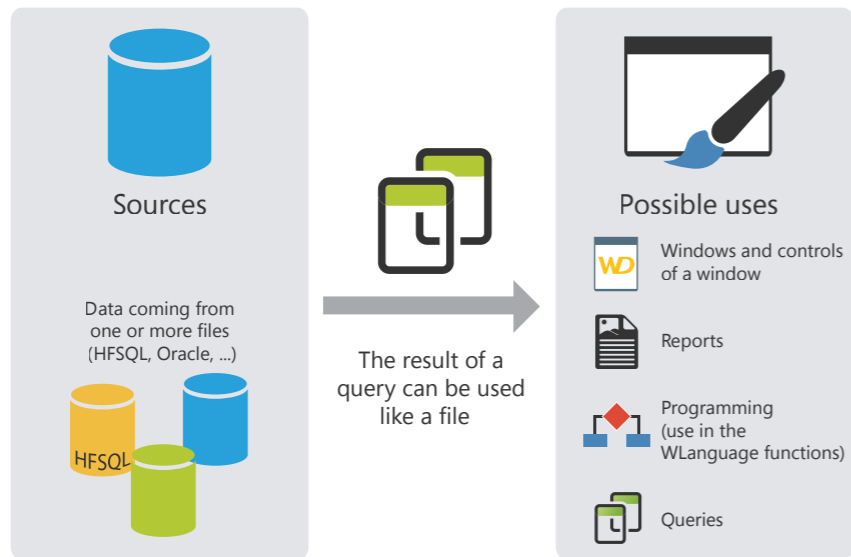


# Queries: concepts

A query is used to request information from a database to view, insert, change or delete data. The query structure defines the data used.

A query can request information from one or more data files.

The query editor allows you to easily create queries without programming.



**Remark:** In the code, you can handle queries and data files in the same way. Queries can be associated with controls in the UI (a Table control, for example) to display the desired data.

# Embedded queries

The controls from a window can be linked to a data file or existing query, ... These controls can also be linked to a query created when building the control.

This block contains three screenshots. The top right screenshot shows a 'Req. Modeler' window with a flowchart diagram. Below it is the caption 'Embedded query: MyWindow\_1\$Query'. The middle screenshot shows a 'Req. Component' window with a table of computer components. Below it is the caption 'Embedded query: MyWindow\_2\$Query'. The bottom screenshot shows a 'WD HC DOC Concepts - WINDEV | test in progress | [WIN\_The\_Embedded\_Queries]' window with a table of computer components. Arrows point from the table in the bottom screenshot to the table in the middle screenshot, and from the table in the middle screenshot to the flowchart in the top right screenshot.

Type	Last name	Make	Comment
Bolider/Sup.	10860	FUJITSU	
Bolider/Sup.	10860	LPA	ATX
Network card	1211TX	SMC	
Screen	12XE	DELL	
Video card	258 BUSINESS	MATROX	

Type	Make	Model	Supplier Name	Min. lease Date	Guarantee
***	***	HP	622811848	October 12	
***	***	DELL	515142456	December 12	
***	***	IBM	EDG505DD	November 20	
***	***	COMPAQ	EDG505DD	December 01	
***	***	HEWLETT	BQ505DD	December 05	

In this case, the query is embedded in a window. It is located in the WDW file that corresponds to the window. If the WDW file is copied (to another project, for example), the embedded queries used by the window will also be copied.



# Table/Looper controls

Table/Looper controls can be used to display a set of information (e.g., the content of a data file). The content of these controls can come from three different sources:

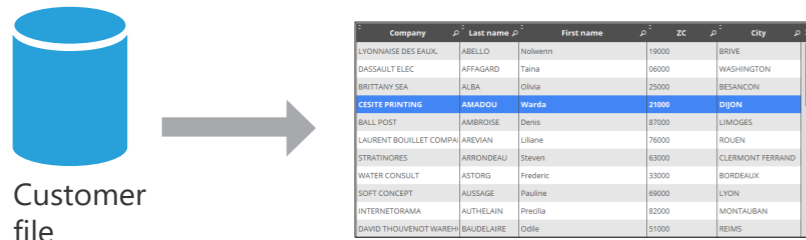
- Table/Looper controls with direct access to the data source,
- Table/Looper controls populated programmatically,
- Table/Looper controls with in-memory data source.

**Remark:** These three modes will be explained in detail for the Table control. However, the same concepts apply to the Looper control.

## Table control with direct access to the data source

A Table control with direct access to the data source directly displays data from a data file or query. The structure the data file is described in the data model editor and the data is entered in the application, for example.

When you loop through the data file, the data is displayed in the Table control. The data file is read for each row displayed: the record read is displayed in a row of the Table control.



The data displayed that is not linked to the data file is lost when the Table control is refreshed (when scrolling, for example).

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

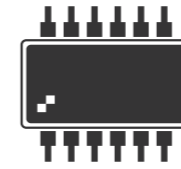
Several WLanguage functions can be used to handle Table controls based on a data file. These functions start with "Table".

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

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

## Table control populated programmatically

In this case, the Table control directly displays the data loaded in memory. Data is added to the Table control programmatically, via different WLanguage functions (**TableAddLine**, for example).



Data in memory

Company	Last name	First name	ZC	City
LYONNAISE DES EAUX	ABELLO	Nolwenn	19000	BRIVE
DASSAULT ELEC	AFFAGARD	Taina	06000	WASHINGTON
BRITTANY SEA	ALBA	Olivia	25000	BESANCON
<b>CEDITE PRINTING</b>	<b>AMADOU</b>	<b>Warda</b>	<b>21000</b>	<b>DIJON</b>
BALL POST	AMBROISE	Denis	87000	LIMOGES
LAURENT BOUILLET COMPA	AREVIAN	Lilane	76000	ROUEN
STRATINORES	ARRONDEAU	Steven	63000	CLERMONT FERRAND
WATER CONSULT	ASTORG	Frederic	33000	BORDEAUX
SOFT CONCEPT	AUSSAGE	Pauline	69000	LYON
INTERNETORAMA	AUTHELAIN	Precilia	82000	MONTAUBAN
DAVID THOUVENOT WAREH	BAUDELAIRE	Odie	51000	REIMS

When data is in memory, you can perform different actions in the Table control (sort columns, search in columns, etc.).

## Table control with in-memory data source

Table controls with in-memory data source combine the advantages of Table controls with direct access to the data source and Table controls populated programmatically.

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

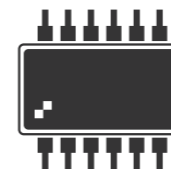
The data that is not linked to the data file is not lost when moving the scrollbar (in Check Box columns, for example).

Since the data file records are loaded in memory, this type of control is recommended for data files with less than 100 000 records (to avoid memory overflow).



Customer file

File loaded in memory



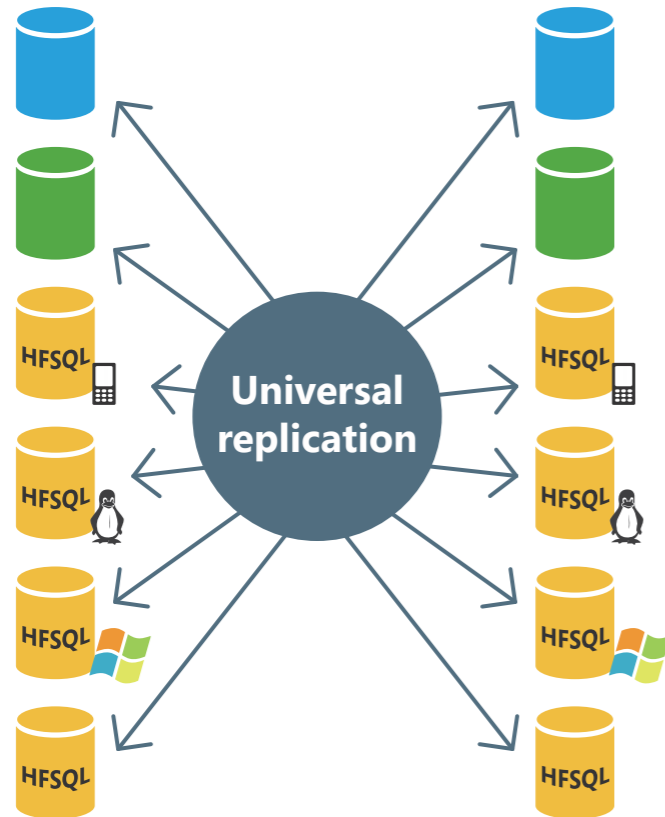
Company	Last name	First name	ZC	City
LYONNAISE DES EAUX	ABELLO	Nolwenn	19000	BRIVE
DASSAULT ELEC	AFFAGARD	Taina	06000	WASHINGTON
BRITTANY SEA	ALBA	Olivia	25000	BESANCON
<b>CEDITE PRINTING</b>	<b>AMADOU</b>	<b>Warda</b>	<b>21000</b>	<b>DIJON</b>
BALL POST	AMBROISE	Denis	87000	LIMOGES
LAURENT BOUILLET COMPA	AREVIAN	Lilane	76000	ROUEN
STRATINORES	ARRONDEAU	Steven	63000	CLERMONT FERRAND
WATER CONSULT	ASTORG	Frederic	33000	BORDEAUX
SOFT CONCEPT	AUSSAGE	Pauline	69000	LYON
INTERNETORAMA	AUTHELAIN	Precilia	82000	MONTAUBAN
DAVID THOUVENOT WAREH	BAUDELAIRE	Odie	51000	REIMS

**Remark:** The different modes to populate the controls (programmatically, with direct access to the data source and with in-memory data source) are available for different types of controls: List Box, Combo Box, Table, Looper, etc.

# Universal replication

Universal replication makes it possible to keep databases of the same or different formats updated (HFSQL, Oracle, SQL Server, etc.). For example, you can synchronize an HFSQL Classic database with an Oracle database.

Universal replication uses a centralized model: all databases are synchronized with a master database. Then, the master database reflect the changes in the other databases.



The synchronization can be adapted to special cases. For example, you can retrieve the records related to a specific product or created on a given date, manage conflicts, display a configuration window, etc.

These adaptations must be done through programming with **HRpIFilterProcedure**.

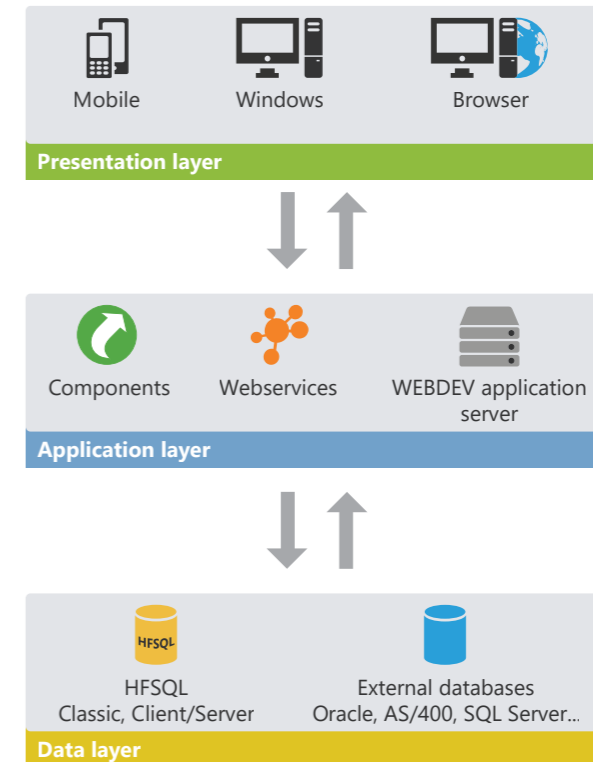
# 3-Tier architecture

3-tier architecture is an application architecture model.

It consists in dividing an application into 3 tiers:

- Presentation tier: the User Interface,
- Application (or processes) tier: business logic of the application,
- Data tier: to access persistent data.

The objective with this separation is to make each tier independent, in order to facilitate maintenance and future developments of the application. This provides better security because only the Application tier allows access to the database. It also optimizes teamwork and cross-platform development.



The 3-tier architecture is fully compatible with the development of applications or sites using WINDEV, WEBDEV or WINDEV Mobile.

PART 4

Advanced  
concept

28

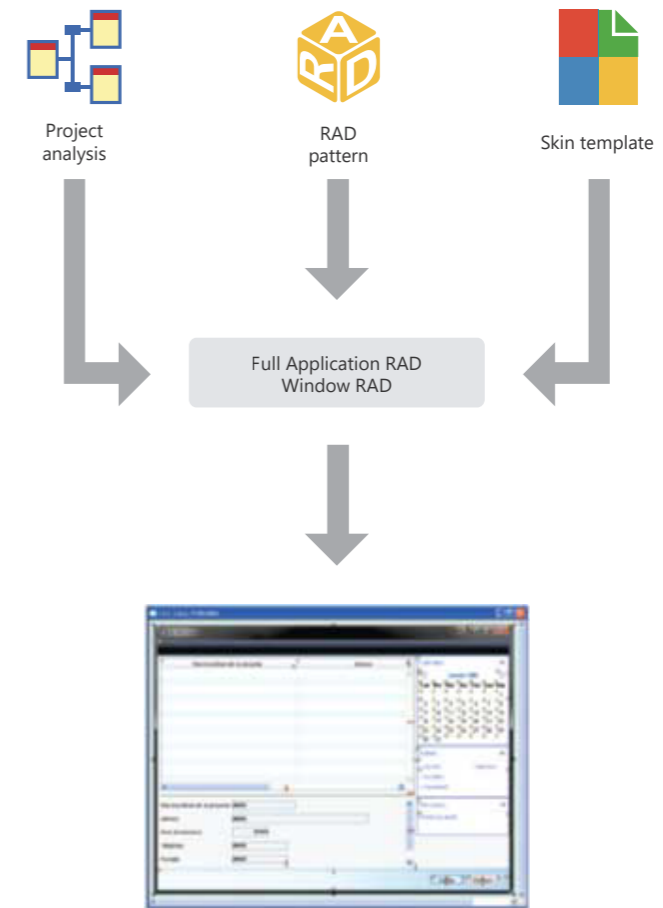
DEVELOP 10 TIMES FASTER

# RAD (Rapid Application Development)

The RAD (Rapid Application Development) is used to create windows from:

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

With RAD, windows are generated with all the code they require to work properly. These windows can be tested immediately with the data present on the development computer.



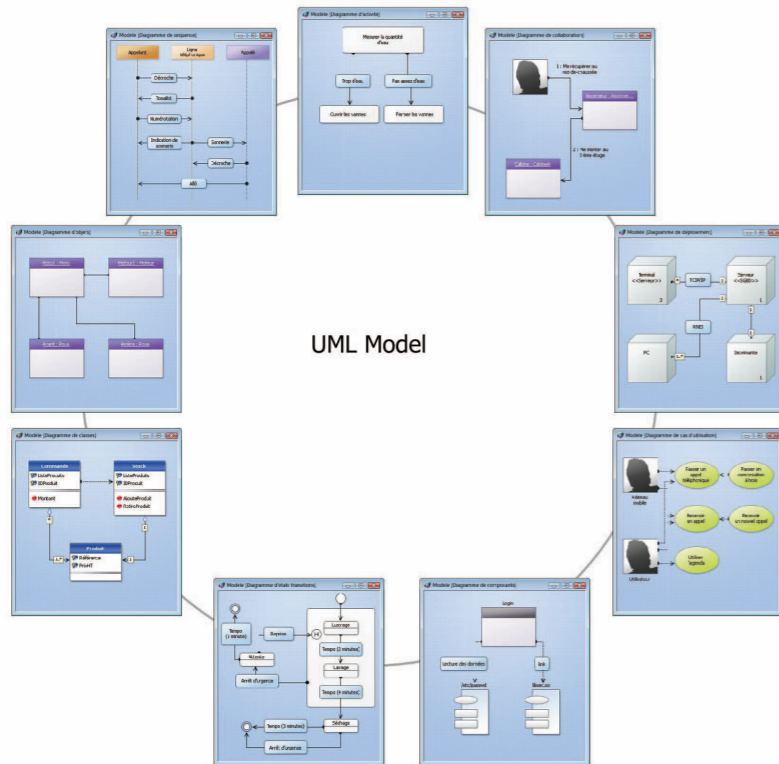
Example of window generated by RAD



# UML models

WINDEV allows you to create nine types of UML models:

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



## UML models in practice

### 1 Overview

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

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

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

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

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

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

This chapter only provides an overview of UML. For more information about UML, see the specific documentation.

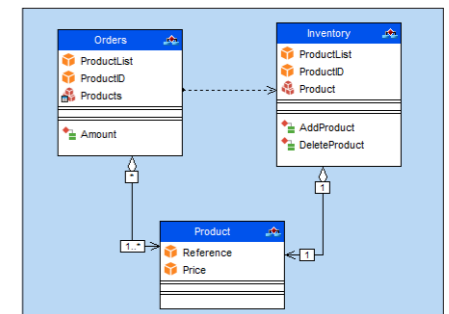
### 2 Types of UML diagrams

#### 2.1 Class diagram

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

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

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





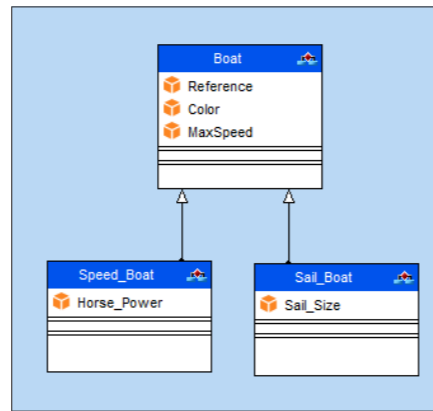
A class diagram includes the following elements:

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

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

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

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



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

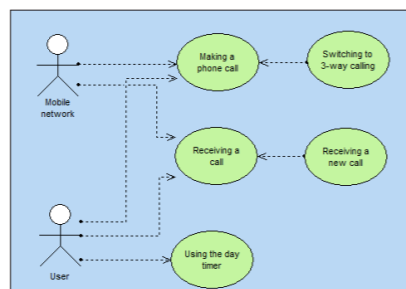
The generation of a class diagram allows you to create the structure of the WINDEV, WEBDEV and WINDEV Mobile classes used in your application.

### 2.2 Use case diagram

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

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

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



A use case diagram includes the following elements:

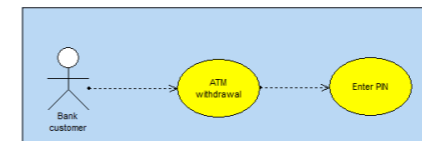
- **Actor:** represents the role of the users that interact with the application. For example, a person who works in a bank as a loan manager. If they have an account in the same bank, they will also play the role of customer.

- **Use case:** describes a sequence of actions performed by the application. For example, place an order, enter an invoice, create a new customer entry, etc, ...

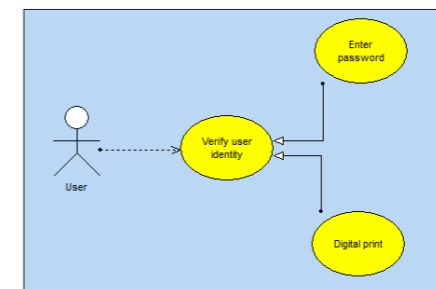
A use case describes the actions performed by an application but it does not specify how the application performs these actions.

- **Relationship:** describes the behavior of an actor with a use case. There are three types of relationships:

- **Association:** Structural relationship between two linked elements.
- **Dependency:** Indicates that one element uses or depends on another element. For example, a bank customer may get cash from an ATM. In this case, the Get Cash action depends on the Customer. To withdraw money, the Customer must enter a PIN number. In this case, the Get Cash action depends on the Password.



- **Generalization:** Relationship in which elements are organized based on a hierarchy. For example:
  - the Customer actor can be of two types: Individual customer or Company customer.
  - there are two ways to verify the user's identity: a password or a fingerprint.



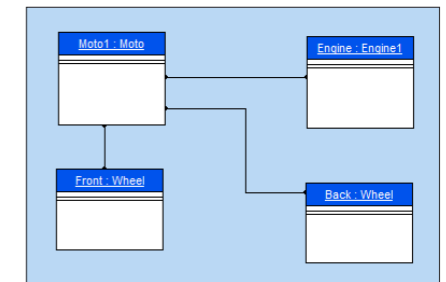
- **Package:** divides and organizes the diagram representation (in the same way that directories organize files). Each package can contain actors and use cases.

### 2.3 Object diagram

An object diagram represents a set of objects and how they relate to each other at a certain moment in time.

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

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



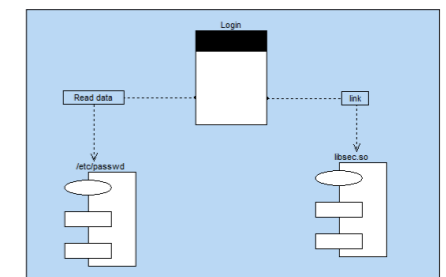
An object diagram includes the following elements:

- **object:** represents a class instance. **Remark:** If a class diagram is opened, you can create an object from a class in this diagram (Drag and drop it from the "UML analysis" pane).
- **composite object:** visually represents an object made of other objects. For example: a window that contains scrollbars, buttons, etc.
- **link:** represents the relationship between different objects.

### 2.4 Component diagram

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

For example, the diagram below presents the steps of a program that allows users to log in in text mode in Unix.



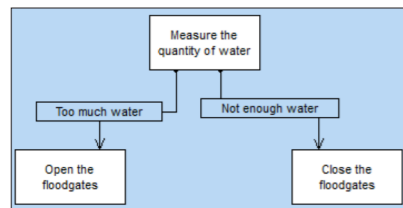
A component diagram includes the following elements:

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

### 2.5 Activity diagram

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

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



An activity diagram includes the following elements:

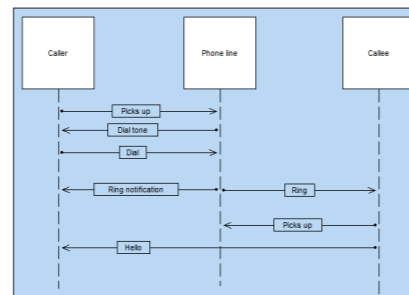
- **activity**: represents a specific step in a workflow. For example: "Print an estimate", "Open the window", etc.
- **synchronization bar**: synchronizes different activities:
  - by indicating the activities that must be performed before a given activity. For example: "Press clutch" and "Change gear" before "Release clutch".
  - by indicating the activities that will happen in parallel.

- **object**: relates activities to the object that performs them. For example, the "Order" and "Pay" activities are related to the "Customer" object; the "Teach" and "Evaluate" activities are related to the "Teacher" object.
- **send signal**: represents a signal sent to an object.
- **accept event**: represents the acceptance of an event coming from an object.
- **transition**: represents the transition from one completed activity to another. For example: "Too much water", "Enough money", etc.

### 2.6 Sequence diagram

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

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



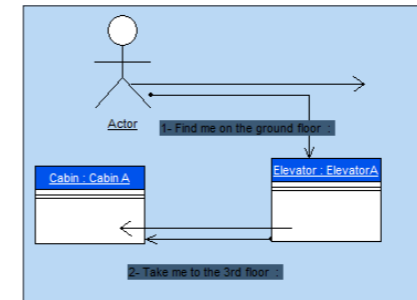
A sequence diagram includes the following elements:

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

### 2.7 Collaboration diagram

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

For example, the diagram below represents a person that uses an elevator:



A collaboration diagram includes the following elements:

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

### 2.8 State-transition diagram

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

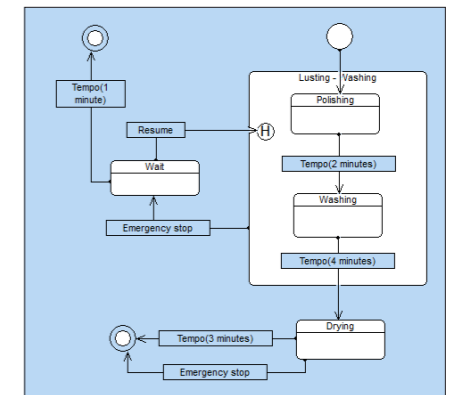
A state is defined by its duration and stability.

A transition represents the change from one state to another.

A transition is triggered:

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

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



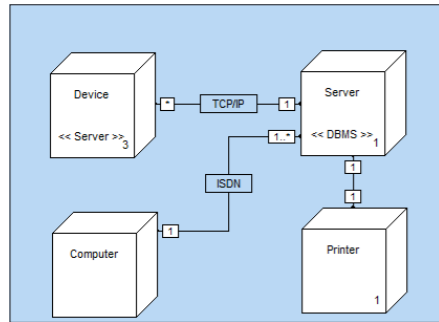
A state-transition diagram includes the following elements:

- **state**: represents the value of the object attributes at a given time.
- **initial state**: represents the state when the system is started.
- **final state**: represents the state of the system at the end of the operation.
- **superstate**: used to structure the diagram by specifying several distinction levels between the states.
- **history**: represents the last active state of a superstate (or composite state).
- **entry/exit points**: represent the states in a superstate. This allows you to link these states to other states that do not belong to the superstate.
- **transition**: represents the change from one state to another.

### 2.9 Deployment diagram

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

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



A deployment diagram includes the following elements:

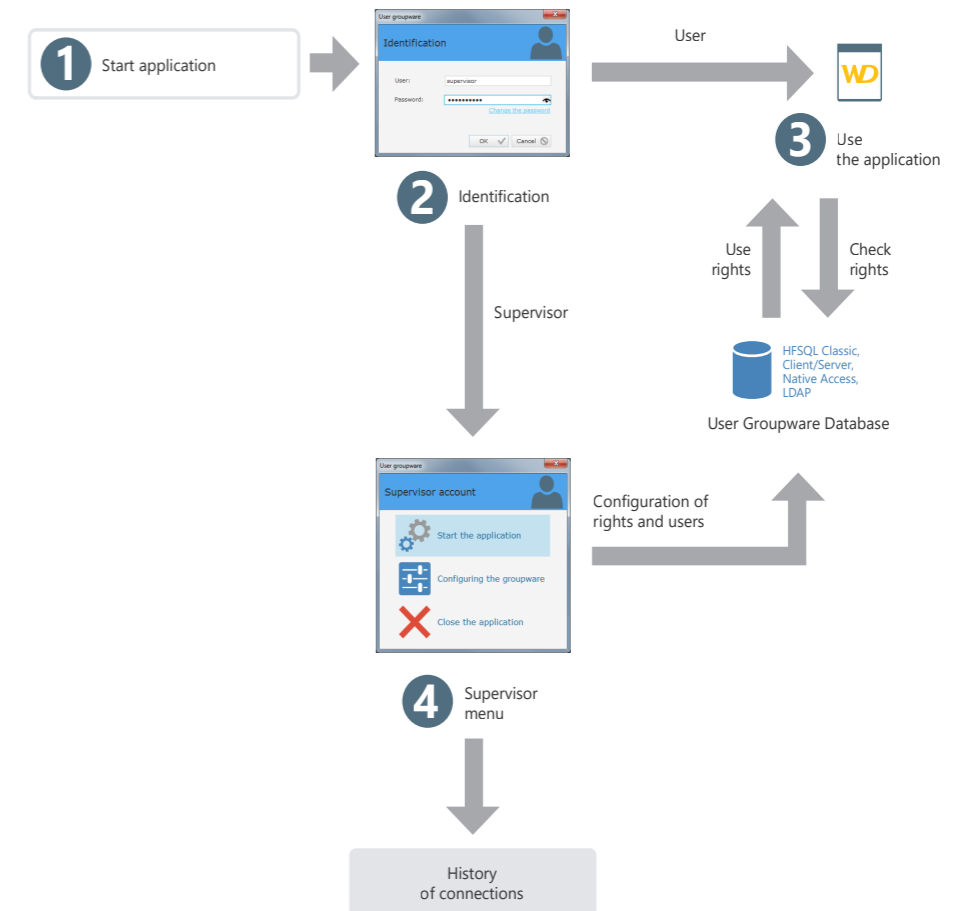
- **node class:** represents a class of a physical resource. For example: server, PC, printer, etc.
- **node instance:** represents a physical resource. For example: server 3, printer 70, etc.
- **connection:** depicts the communication path between two nodes. For example: ISDN or TCP/IP link.

# User Groupware

The User Groupware allows easily including user and rights management in a WINDEV application.

There are two types of users:

- simple user: uses the application directly.
- supervisor: configures the users and their rights.



# User Groupware in practice

## 1 Overview

An application must define the role of different contributors. Several access levels can be defined according to the user.

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

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

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

Access may vary according to the user. For example:

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

To manage these access levels in your WINDEV applications, simply integrate the User Groupware. In just a few clicks, a standard application can be transformed into an application that manages several access levels. This feature can be easily implemented.

When starting the application, the manager will be able to create users (identified by their username and password) and give them access to some application features.

## 2 How does User Groupware work?

An application that uses the User Groupware has two levels:

- user level.
- supervisor level.

### 2.1 User level

The user logs in to the application and accesses the available features.

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

### 2.2 Supervisor level

The supervisor logs in to the application and accesses an advanced menu to configure 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 and associating users and groups, and managing the rights granted to each user (or group) on each window of the application. Rights can be defined for menu options, groups of controls and controls. The supervisor can gray out these elements, make them inactive or invisible. These configurations take precedence over any configuration defined in the program.

For more information on the groupware configuration defined by the supervisor, see [Configuring the application that uses the User Groupware](#).

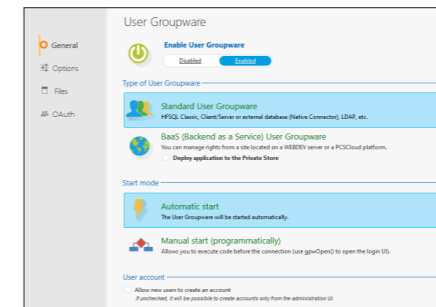
**Remark:** The information regarding the users and their rights is stored in data files in HFSQL format (Classic or Client/Server).

# 3 Implementing the User Groupware

## 3.1 Adding the User Groupware to an application

To implement the User Groupware in a WINDEV application:

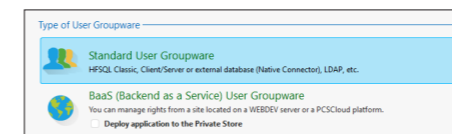
1. On the "Project" tab, in the "Project" group, click "User Groupware".
2. In the description window of the current project configuration, in "User Groupware: Disabled", click "Change".
3. The User Groupware settings window allows you to enable the User Groupware and define:



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

### Type of User Groupware

There are various types of groupware:



- **Standard User Groupware:** The User Groupware will be integrated in its "Standard" version: the User Groupware will be associated only to your application, and the rights will be defined from the application. If this option is selected, two specific tabs are displayed:

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

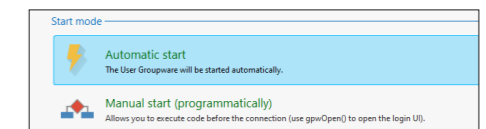
### BaaS User Groupware: Required configuration:

Two cases may occur:

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

### Starting the User Groupware

Several start modes are available:

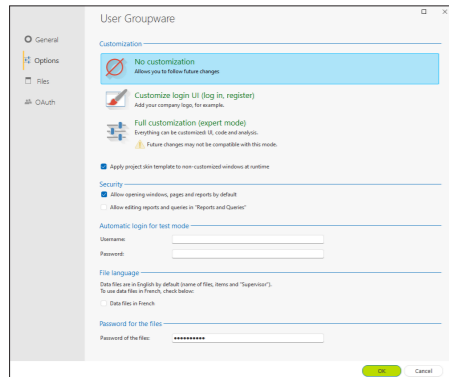


- The "Automatic start" option indicates that the User Groupware is started at the same time as the application. The User Groupware login window is started before any other window of the application. The project initialization code is executed once the login window is opened and validated.
- The "Manual start (programmatically)" option allows you to define when the User Groupware should be started.

The login window will only be opened if **gpwOpen** (or **gpwLogin**) is used. This option can be used to execute the project initialization code before opening the login window. This option is useful for a multilingual application for example: a window can be displayed to ask for the application runtime language.

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

There are multiple User Groupware customization modes:



- **No customization:** WINDEV takes care of everything. You just need to start the application to configure user access. The windows of the User Groupware can automatically use the skin template of the application. Simply check "Apply project skin template to non-customized windows at runtime". To get the same appearance for all the windows of the application, it is recommended to always check this option if you use the User Groupware in non-custom mode.
- **Customize login UI:** The login/register window is automatically integrated into your application. The internal component named **WDGPU** is included in your project. This component contains the login window and the associated procedure. You can customize the connection to the User Groupware.

- **Full customization (expert mode)** - Standard User Groupware only: The programming of the User Groupware is automatically included in your application. The internal component named **WDGPU** is included in your project. This component contains an analysis as well as all the windows, queries, classes and procedures required to manage the User Groupware. This allows you to customize all the windows used by the User Groupware.

The option "Apply project skin template to non-customized windows at runtime" applies the skin template of your project to the windows specific to the User Groupware if they have not been customized.

The security options are as follows:

- **Allow opening windows and reports by default:** This option allows all users to open windows and reports by default. In this case, simply specify which elements will not be accessible when configuring the rights.
- **Allow editing reports and queries in "Reports and Queries":** If this option is checked, the application will use "Reports and Queries" for all the reports and queries. Remark: This option is identical to "Enable 'Reports and Queries'", in the "Advanced" tab of the project description.

The option "**Automatic login for test mode**" allows you to specify the username and password to be used in the "Automated test" mode. This information will be used if an automated test is run on the application.

By default:

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

For more details, see the online help.

### 3.3 Options specific to the standard User Groupware

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

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

#### Database type

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

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

#### Type of authentication

The standard User Groupware includes several user authentication modes:

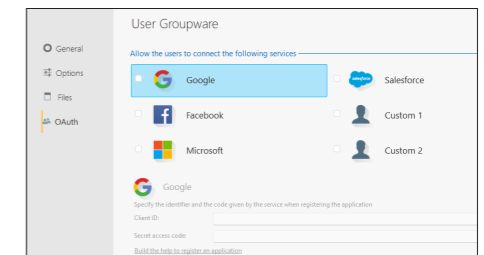
- **Authentication entirely managed by the User Groupware:** The username and password are saved in the User Groupware data files.
- **Authentication from an LDAP directory or Active Directory:** If the company of your customer uses an LDAP directory, you can enable the User Groupware on this directory. Simply click the link "Click here to enter the server parameters" to specify the characteristics of the LDAP directory (server, port, base DN, etc.). Therefore, the account and the password in the LDAP directory will be automatically requested to the user. For more details, see the online help.

- **Windows authentication:** In this case, the username and password correspond to those defined for Windows. Remark: You can also use a unique authentication. In this case, the username and password will be requested only once per session.

Depending on the option chosen, additional options may be available:

- **Login window only (no management of rights):** This option is only available for LDAP and Windows authentication. If this option is checked, no groupware data file will be created. If the application is started in automatic mode, the user will need to log in. If the user is saved in the LDAP directory, the application will start; otherwise, it will be closed. A supervisor cannot configure the rights on the windows. If this option is not checked, only the users in the LDAP directory will be able to log in. You can set the rights on the windows.
- **Single authentication (SSO):** This option allows you to enter the password only once per Windows session.

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



### 3.4 Options specific to the BaaS User Groupware

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

The "BaaS" tab allows you to define:

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

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



### 3.5 Testing the application

When an application that uses the standard User Groupware is tested, the first window that appears is the login window (regardless of the first window defined in your application).

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

- Name : SUPERVISOR
- Password: SUPERVISOR

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

**Remarks:**

- To set a first window other than the login window in your application, select "Manual start (programmatically)" in the "General" tab of the User Groupware options. Simply use `gpwOpen` to open the login window.
- The User Groupware management windows are executed before the project initialization code.
- The window of your application defined in the project will be opened after the login window (if the user is not the supervisor).
- In the "Options" tab of the User Groupware configuration window, you can define the username and password to be used for the test mode. The password will no longer be requested when you click GO to test the window or project.

### 3.6 Creating the WINDEV executable

When creating an executable that uses the standard User Groupware ("Generate" option on the "Project" tab), the wizard will show a screen specific to the standard User Groupware.

This allows you to define the location of the User Groupware data files.

### 3.7 Installing an application that uses the User Groupware

An application that uses the User Groupware is installed like any other application: on the "Project" tab, in the "Generation" group, click "Setup procedure".

The wizard displays a screen specific to the standard User Groupware to set the location of the User Groupware data files.

**Remarks:**

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

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

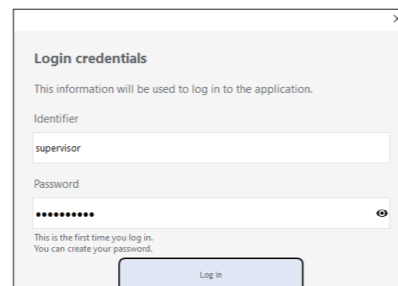
## 4 Configuring the application that uses the standard User Groupware

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

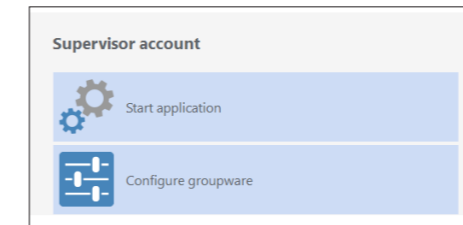
This configuration consists in creating users and groups of users, and granting them specific rights for each control in each window of the application. These rights are used to gray controls, groups of controls or menu options, and to make them visible or invisible.

**To configure the standard User Groupware, simply:**

1. Start the application (executable or test from WINDEV) and log in as supervisor:
  - Name : supervisor
  - Password: supervisor



2. Select "Configure groupware".

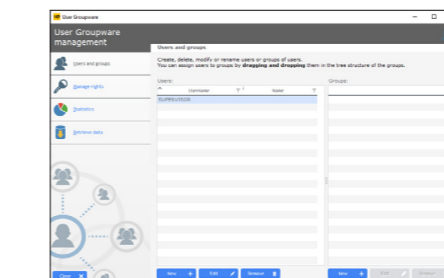


3. The User Groupware management window appears. This window is used to:
  - manage users and groups,
  - manage rights,
  - display statistics,
  - migrate data coming from a User Groupware version 17 and earlier.

### 4.1 Managing users

The user management consists in:

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



To create a new user, you must specify:

- the user's last name.
- the user's first name.
- the username (mandatory). This username is the identifier used to log in to the application.
- the user's password. This password is optional and can be entered by the user the first time they log in (check the corresponding option).

The user can be defined as supervisor of the application.

You can also modify or delete users. When you delete a user, you can delete the entire user or their rights only.

**Users can be grouped.**

To associate a user with a group, simply drag the user and drop it onto the selected group.

A user can be associated with several groups.

**Remarks:**

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

### 4.2 Managing rights

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

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

The elements in the windows, internal windows and window templates are as follows:

- controls.
- groups of controls.
- menu options.

The following states are available for each element:

- **Default:** the element behavior corresponds to the default behavior, defined in the application.
- **Inactive:** the element is displayed but no input can be performed.
- **Grayed:** the element is grayed. No input can be performed.
- **Invisible:** the element is not displayed.

### 4.3 Displaying statistics

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

### 4.4 Migrating data

From version 18, the User Groupware has been optimized. The data files used by the User Groupware have changed.

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

## 5 Tips for an application that uses the User Groupware

### 5.1 Using groups of controls

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

You can create groups of controls in your windows according to the controls to be displayed for a specific user.

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

These groups of controls can be created in your 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 of the elements in your windows:

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

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

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

### 5.3 Defining rights

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

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

# Multilingual applications

A multilingual application can be distributed in multiple languages. WINDEV Mobile takes into account the different languages of the application during the entire development cycle.

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



# Multilingual applications in practice

## 1 Overview

A multilingual application can be distributed in multiple languages. The different languages of the application 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. Writing the different elements (window, code, etc.) in the different languages of the project.
3. Defining the project language programmatically.
4. Managing specific character sets in the data files.
5. Creating the executable and the setup program.

**Remarks:**

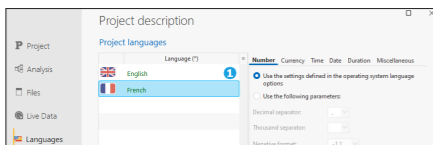
- If the operating system supports several languages (Hebrew, Arabic, Greek, etc.), the corresponding character set will be automatically used when translating to these languages.
- If your application is multilingual, this feature must also be supported in the User Groupware and the automatic HFSQL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

## 2 Choosing the languages supported by the project and the analysis

### 2.1 Languages supported by the project

The different languages supported by the project are defined in the project description:

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



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

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

- for 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 runtime.

### Language options

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

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

**When an Edit control or a Table control column** that shows numeric data (number, currency, date, time, etc.) is created, it will automatically use the input mask defined in the project language options. This option is available in the windows.

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

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

### Executable program

When creating the executable file, you can specify the languages that will be included in it. You can create a different executable file 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 uses an analysis, the languages supported by the analysis are selected in the data model editor. The same analysis can be shared between different projects that do not use the same languages. Therefore, the analysis might use more languages than the project.

To configure the languages supported by the analysis:

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

The different languages set in the analysis will be available:

- when configuring the shared information of the items. The description of the controls bound to the items (options, captions, etc.) can be entered in the different languages supported by the analysis. When generating a "full RAD" or window RAD, this information is automatically used for all languages common to the analysis and the project.
- for the information printed in the technical documentation (notes of data files or items).
- for the information managed by "Reports and Queries". If "Reports and Queries" is provided with your WINDEV application, you can translate the names of the data files and the items. Users will be able to use the names in the desired language when creating their reports and queries. This information can be entered in the "Reports and Queries" tab of the data file and item description.

### 2.3 Languages supported by the different project elements

By default, the different elements of the project (windows, reports, code, classes, sets of procedures, etc.) support the same languages as the project in which they were 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 Translating the interface into multiple 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.

Several elements must be taken into account to create the UI in different languages:

- The selected translation mode.
- The supported languages (with or without a specific character set).
- The use of specific windows (User Groupware, automatic HFSQL error handling, report viewer, etc.).
- The messages displayed through programming.

#### 3.1 Translation method

WINDEV includes different translation modes:

- Automatic translation of the UI via a specific translation tool, installed on the development computer. This translation is performed in the WINDEV editor.
- Export all the messages via WDMMSG and reintegrate them once they have been translated.

#### Translating the UI

The description window of each element has different tabs with multiple multilingual areas. These multilingual areas allow you to enter information in the different languages supported by the project.

To translate this information from WINDEV:

1. Go to the WINDEV options to configure the translation options of the software used: on the "Home" tab, in the "Environment" group, expand "Options" and select "General options of WINDEV".
2. In the "Translation" tab, you can define:
  - the default translation tool.
  - 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. Select the text to be translated and click "Translate".

#### Exporting and reintegrating the information to translate

WDMMSG is a stand-alone tool used to extract all the messages of a project (control captions, tooltips, etc.) and reintegrate them once they have been translated.

Contact PC SOFT Sales Department for more details about the conditions for using this product.

#### 3.2 Languages that use a specific character set

If your application supports languages that use specific character sets (Greek, Russian, etc.), the different messages must be translated using these character sets.

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

When the text cursor is located in the input area of a language that uses a specific character set, the corresponding input language (character set used by the keyboard) is automatically enabled.

Therefore, if a caption is entered in the Russian section of the caption description, the keyboard will automatically switch to the Russian character set.

**Reminder:** To use specific character sets, the files corresponding to the desired character sets must be installed in the Windows region settings (control panel).

#### 3.3 Translating specific windows

##### Handling HFSQL errors

By default, HFSQL error handling windows are available in English and French. To translate them into another language, you must:

1. Include the default error windows in your project. You will find examples of these windows (with their WLanguage code) in \Programs\Data\Preset windows\HFSQL - Automatic help windows.
2. Customize error handling to use HFSQL error handling windows (**HOnError**).
3. Translate the different messages (see previous paragraph).

#### Automatic error handling

By default, automatic error handling windows are available in English and French. To translate them into another language, you must:

1. Include the default error windows in your application. You will find examples of these windows (with 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 available in English and French.

To translate an application that uses the User Groupware, you must choose a custom integration when configuring the User Groupware in your application.

**Reminder:** To configure the User Groupware, go to the "Project" tab, "Project" group and click "User Groupware".

During the custom integration, the different elements of the User Groupware are included via an internal component.

You will only need to translate these elements when you translate the project.

**Remark:** If necessary, when configuring the User Groupware, in the "General", tab, select "Manual start": this allows you to display a window to select the language of the application before the login window.

#### Report viewer

By default, the report viewer window is available in English and in French.

To translate this window, you must integrate the windows corresponding to the report viewer into your application. You will find examples of these windows (with their WLanguage code) in \Programs\Data\Preset windows\Preview. For more details, see the online help.

Then, all you have to do is translate these windows.

#### System menus

WINDEV includes a default context menu for Edit and Table controls. This menu allows users to easily perform several operations.

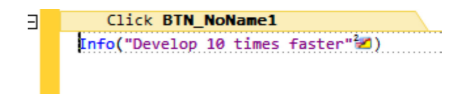
To translate these menus, you must have WDMMSG.

#### 3.4 Translating WLanguage code messages

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.

To translate a character string typed in the code editor:

1. Select the character string to be translated.
2. Select "Translate" in the context menu of the selection. The window for translating messages is displayed.
3. Type the translation and validate.
4. The line of code is displayed in the editor. An icon appears to the right of the translated message. This icon indicates that the current message has multiple languages. The icon allows you to quickly access the translation window.



When the icon is hovered over, a tooltip shows the different translations found for the string in all the project languages.

**Remark:** To convert all the messages in your code into multilingual messages, go to the "Code" tab, "Languages" group, expand "Translate strings" and select "Convert simple strings into multilingual messages".

## 4 Choosing the language through programming

A multilingual application can be distributed in multiple languages. The user will be able to choose the language used by the application at runtime. You can for example:

- prompt the user to choose the runtime language the first time the application is started

- include an option (menu option or button, for example) to allow users to change the language while the application is running.

## 5 Defining specific character sets in HFSQL data files

When an application displays or saves data in HFSQL data files, this data is typed in the current language. If the application is used in a country that uses a Latin character set, no specific configuration is required.

If the application is used in a country that uses a specific character set, this character set must also be supported by the data files. This means that search, sort and other operations will be performed according to the selected character set.

## 6 Unicode and specific character sets

WINDEV includes two modes:

- **single charset mode** ("Use ANSI strings at runtime"): In this mode, if your application includes languages that use specific character sets (Greek, Korean, etc.), change the following elements through programming:
  - the character set used by all the fonts used on the screen (**ChangeCharset**). This character set can also be used for the data saved in the HFSQL files. For more details, see the online help (keyword: Multilingual).
  - the language of the keyboard used by the application users (**ChangeKeyboard**).
- **full Unicode support** ("Use UNICODE strings at runtime"): This mode can be used for non-Latin character sets. In this mode, **ChangeCharset** and **ChangeKeyboard** don't have any effect. The character set is automatically changed based on the language options defined in the project description. You can also combine different character sets.

The Unicode options can be set in the current configuration ("Unicode" tab in the configuration description window).

### Important

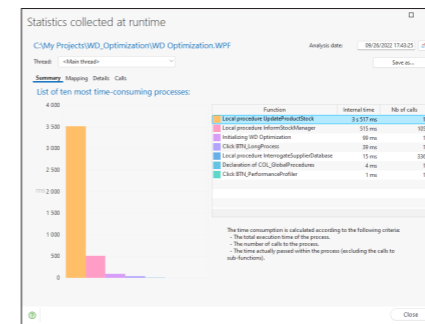
If you change your projects to Unicode, the ANSI/Unicode combination generates some constraints: Data must be converted when changing from one encoding system to another. Any data exchange must be adapted depending on how strings are represented in memory (in Unicode, 1 character is encoded on 2 bytes, while in ANSI, it is encoded on 1 byte); for example, calls to APIs, read from or write to the disk to exchange data, read from or write to sockets containing strings, etc, ... If you change your projects to UNICODE, a wizard starts automatically to measure the impact of this change.

# Optimizing an application

WINDEV includes multiple methods to optimize your applications:

- **The performance profiler**, which allows you to quickly identify time-consuming processes. You can then optimize the desired process.

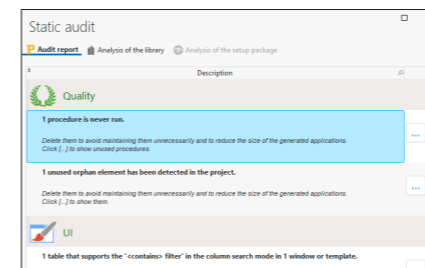
## Application test



Analyze the performance of the application during a test and view the longest processes to optimize the source code. To start the performance profiler, go to the "Project" pane, "Audit and performance" group, expand "Analyze performance" and select "Analyze performance".

- **The static audit of the project**, which points out potential programming issues (empty groups, orphaned elements, etc.) as well as large files that can cause abnormally long loading times.

## Static audit

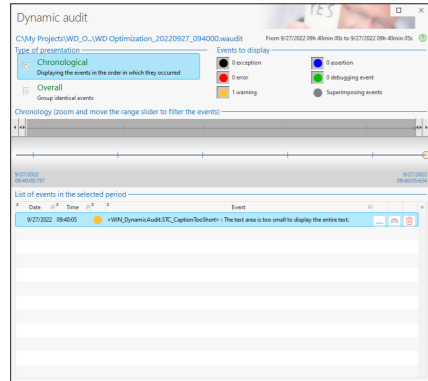


Audit report, analysis of the library (to detect large files) and analysis of the setup package. To start the static audit, go to the "Project" pane, "Audit and performance" group, expand "Edition audit" and select "Trigger edition audit".



- The **dynamic audit of the project**, which detects loss of resources, unprocessed errors and slowness of the application in its production environment.

### Dynamic audit

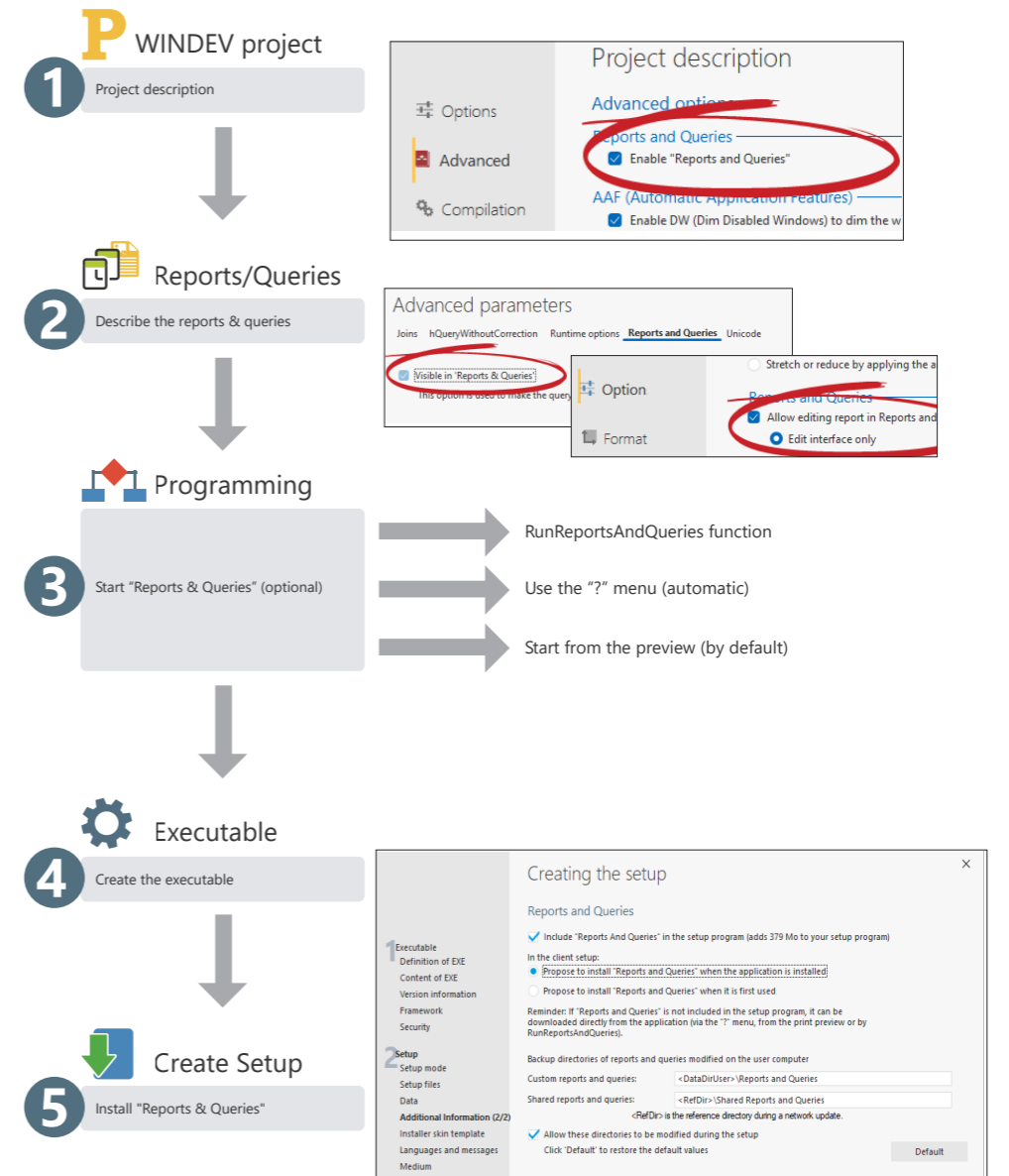


Analyze the performance of the application during a test, view processes containing exceptions and errors that are not properly handled.

To start the dynamic audit, on the "Project" pane, expand "Test mode" and select "Debug the project while the audit is enabled".

# Distributing "Reports and Queries"

With "Reports and Queries", which can be freely distributed with your applications, users will be able to freely customize the reports you provide. Of course, only the reports you have selected can be customized.

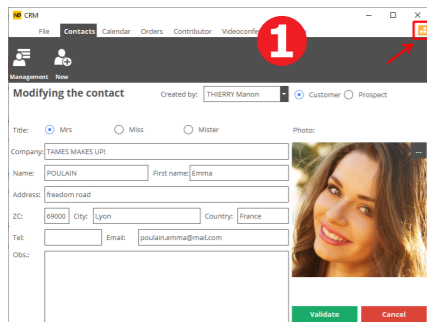


# UMC: User Macro-Code

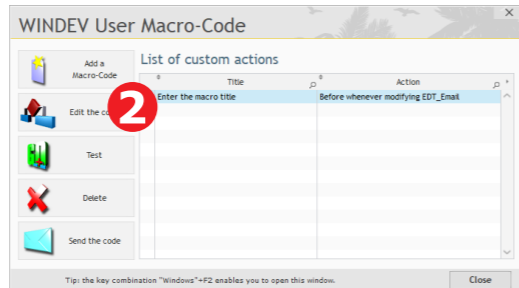
The User Macro-Code (UMC) allows end users to enter their own WLanguage code and include it in an existing application.

Let's consider the following case:

A window for entering a customer form requires an email address but it does not check its format. The user wants to check if the "@" character is in the address.



An icon at the top of the window allows the user to customize the processes.



The UMC configuration window 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 the UMC:

- in the project description,
- when creating the executable.

## User Macro-Code: UMC in practice

### 1 What is the UMC?

UMC is an option that allows advanced users to create their own processes in an application. End users can type WLanguage code and include it in an existing application.

The new processes can be sent to the developer so that they can be added to the source code of the application.

Multiple WLanguage functions are available to the users. However, the developer can set certain restrictions. For example, do not allow the use of file management functions. To maintain data integrity, the commands to update or add records are not available to the users.

### 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 set the UMC on the project:

1. Open the project description window: on the "Project" tab, in the "Project" group, click "Description".
2. Go to the "Advanced" tab. Click the "User Macro-Code" button to allow or prevent users from entering User Macro-Code in the entire application.



Two options are available if you allow it:

- Allow the use of HFSQL commands (HReadFirst, HFilter, HAdd, etc.).
- Accept the suggestions made by the users.

The second option allows the application to send an email when the user creates a macro code. The "\*.MCU" file corresponding to the macro code is sent as an attachment. This allows you to easily retrieve the suggestions made by the users and automatically include them in your project! WINDEV includes an option for importing "\*.MCU" files (on the "Project" tab, in the "Project" group, expand "Import" and select "User Macro-Codes...").

# Available tests

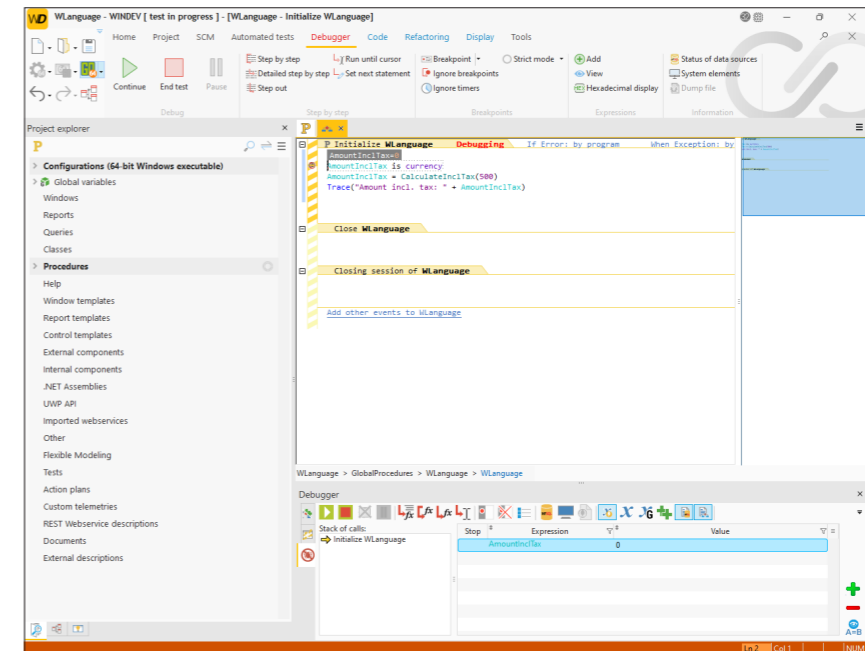
WINDEV includes multiple methods to test applications. The table below presents the different modes available.

Type of test	Description
Test of the entire project 	Runs the application like the executable file would. Tests the entire application, even if it is not fully developed. When a problem occurs during the execution of the project, you can start the debugger to identify and fix the problem. Remark: If the number "64" appears on the "Go" icon, it means that the test will be run in 64-bit mode in a 32-bit editor.
Test of the current element 	Tests an element once it has been developed. You can start the debugger when a problem occurs in reports and windows. Remark: If the number "64" appears on the "Go" icon, it means that the test will be run in 64-bit mode in a 32-bit editor.
Test of a project step by step (via the debugger)	Allows you to start the debugger when the application is run. This solution is used to monitor the operating mode of the application via the debugger. You can also debug a specific executable file, a running executable or an executable file running on another computer.
Unit test	Tests windows, procedures and classes in an application at different stages of development. The test scenarios are automatically generated in WLanguage from the application test or from the element test. The scenarios can be directly modified in the code editor.
Regression test	Used to make sure that an element did not cause any regressions between different versions of an application. To do so, select (or create) the test scenario that will be run.

# Debugger: concepts

WINDEV includes a debugger to help you optimize your code.

With the debugger, you can run your applications step by step, view the WLanguage code being executed, the content of variables, etc.



## Debugging the current project

### 1. Using breakpoints in the code editor.

The debugger will be automatically started when the line preceded by a breakpoint is run.

To insert a breakpoint, simply click to the left of the line of code: a red dot appears.

```

● AmountInclTax is currency
AmountInclTax = CalculateInclTax(500)
Trace("Amount incl. tax: " + AmountInclTax);
  
```

**2. Using the STOP keyword in the WLanguage code**

The debugger will be automatically started when this line of code is executed.

**3. Using the "Debug project" option**

The debugger is launched at the same time as the application test. The option "Debug project" can be found on the "Project" tab, under "Test mode".

**4. Pressing Ctrl + Pause during the window or project test**

The debugger starts with the action performed after using Ctrl + Pause.

**Debugging an executable**

WINDEV also allows you to debug the executable file associated with the current project. This executable file 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:

- Test the component from the component project.
- Test the component by running another project located on the development computer. This allows you to test the component from the project that uses it.
- Test the component by running an executable file located on the development computer. This allows you to test the component directly from the executable file that uses it.

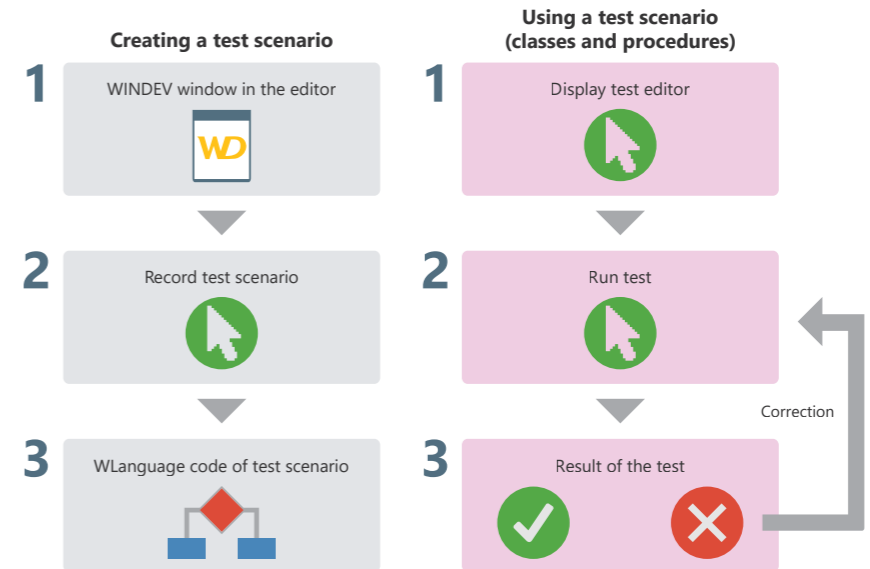
# Unit tests/Automated tests

Unit tests (also called automated tests) are used to test windows, procedures and classes in an application at different levels of development.

To do so, select (or create) the test scenario that will be run.

Test scenarios are generated in WLanguage and can be directly modified.

These scenarios can be found in the test editor. The test editor analyzes the result of the unit tests and calculates the validation rate of the application.

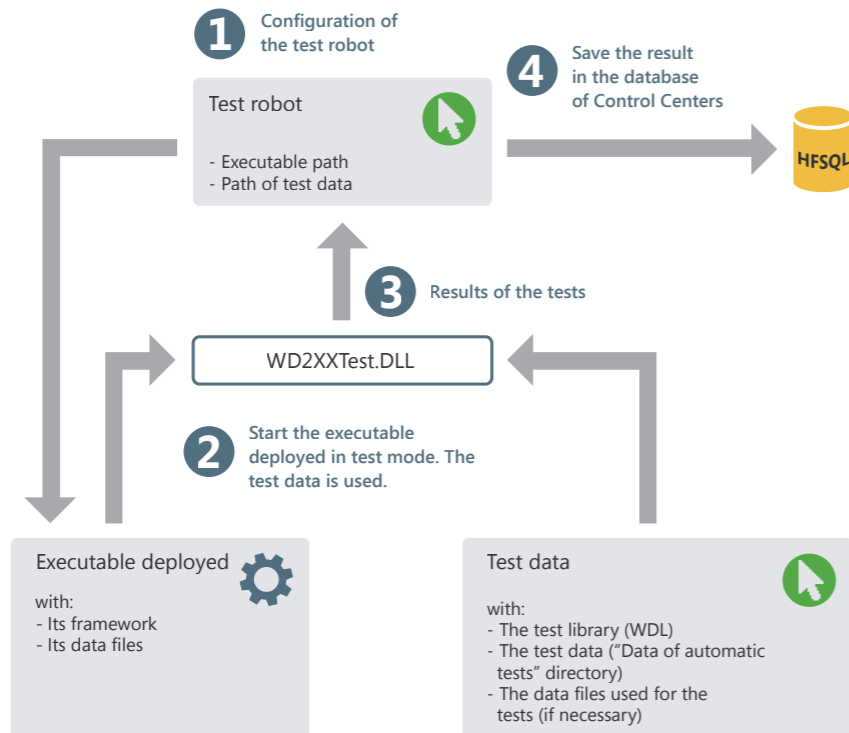


To create a unit test on the current window, go to the "Automated tests" tab, "Tests" group, expand "New" and select "Record a new scenario".

To create a unit test on a procedure or on a class method, select the procedure or class method in the "Project explorer" pane and select "Unit tests .. Create a unit test" in the context menu.

# Unit tests on the executable

The unit tests created with WINDEV can be run on a deployed executable, via the test robot.



## Testing an application

### 1 Overview

WINDEV includes multiple methods to test applications:

- test of the entire project,
- test of a single window,
- test of a single query (see the online help for more details),
- test of a single report (see the online help for more details),
- step-by-step project execution,
- application performance test,
- regression test/automatic test.

**By testing the project** you can simulate how the application will run when launched from an executable. This allows you to test the entire application, even if it is still being developed. If a problem occurs, you can start the debugger to identify and fix the problem.

**The window test** is used to run the current window only. This allows you to test your project from a given window or to test a window once it has been finished. Like for the project test, the debugger can be started if a problem occurs.

**The test of a single query** runs the current query only. This allows you to check the operating mode of a query once it has been developed.

**The test of a single report** runs the current report only. This allows you to test a report once it has been developed. Like for the project test, the debugger can be started if a problem occurs.

**The step-by-step project execution** allows you to start the debugger when starting the application. This solution allows you to monitor the application progress.

**The performance test** checks and optimizes the execution time of your application.

**The regression test** (or automated test) is based on the execution of scripts. It checks that the existing features are still supported when running the applications.

In addition to these methods, WINDEV also provides a "Code coverage" tool to measure how many tests have been executed on your application. For more details, see the online help.



## 2 Testing and debugging a WINDEV project (development computer)

### 2.1 Overview

The test of the entire project allows you to simulate the start of the application with the executable. This allows you to test the entire application, even if it is still being developed. When a problem occurs during the execution of the project, you can start the debugger to identify and fix the problem.

WINDEV includes an extensive set of application tests that can be run from the development computer:

- Test and debugging of the current project in the editor.
- Execution and debugging of the executable corresponding to the current project.
- Debugging of an executable already run on the development computer.

**Remark:** The test of a project can be run regardless of the current element in the editor.

### 2.2 Testing and debugging the current project

To test a project from the editor, go to the "Project" tab, expand "Test mode" and select "Debug project" (Ctrl + F9). The editor is automatically minimized and the project is run.

### 2.3 Testing and debugging the executable on the development computer

Some problems can only be reproduced when testing 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 executed comes from the libraries (and not from the code in the elements on the computer). Therefore, the code from the current project and the code from the executable may be 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.

For more details, see the online help.

### 2.4 Debugging an executable already run on the development computer

Some problems can only be reproduced when testing the executable in real-use conditions.

This test mode allows you to find a problem that only occurs at runtime (and not in test mode).

The principle is simple:

- The developer starts the executable on their computer and directly goes to the desired location in the application.
- In WINDEV, the developer opens the project that correspond to the executable and establishes a connection with it.
- The WINDEV debugger monitors the execution of the program step by step.

In this case, the code executed comes from the libraries (and not from the code in the elements on the computer). Therefore, the code from the current project and the code from the executable may be out-of-sync: the out-of-sync code is highlighted in pink during the debug operation.

For more details, see the online help.

## 3 Testing and debugging a WINDEV project on the user computer

WINDEV includes multiple methods to test and debug a project on the development computer. However, in some cases, you may have to debug directly on the end-user computer (connected over the network or via the Internet).

You can, for example, debug an application executed in Taiwan from your office in Paris. The debugging is performed remotely, directly on the user configuration.

Two features are available:

- Running and debugging the executable on a remote computer,
- Debugging an application currently running on a remote computer.

For these two features, a specific configuration of the remote computer is required.


**Remark:** to debug a WINDEV application remotely, the application must include the WD280CPL.DLL library in its framework.

For more details, see the online help.

## 4 Testing a window

### 4.1 Testing a window from the editor

To test a window from the editor:

1. Open the window to be tested.
2. Click  in the quick access buttons (or press F9). The editor is automatically minimized and the window is run.

During the test, all window features can be run. You will be able to open other windows, for example.

### 4.2 Stopping the window test

There are multiple methods to stop the test:

- 1st method:  
Close the application being tested. WINDEV goes back to editor that was open when the test was started.
- 2nd method:
  - Go back to the editor via the taskbar or press Alt + Tab.
  - Stop the test. WINDEV goes back to editor that was open when the test was started.

## 5 Tracing a project

### 5.1 Debugging principles

Debugging an application consists in:

- checking the operating mode of a process,
- understanding the operating mode of an existing program,
- checking the values of variables,
- checking the operating mode of special cases in an application.

The debugger is used to perform these operations.

**Remark:** WINDEV also includes multiple trace tools (trace window, dialog box, etc.). For more details, see the online help.

### 5.2 Debugger overview

The debugger monitors WLanguage programs so they can be improved.

The executed source code appears on the screen. The processes run are sorted in hierarchical order in the "Debugger" pane.

The value of the variables is displayed:

- individually in the on-hover tooltip of each variable.
- in the "Debugger" pane.

### 5.3 Debugger features

The debugger is used to:

- view the call stack,
- view the content of variables or expressions,
- execute the code step by step with the possibility to skip code blocks,
- use conditional breakpoints,
- modify the code while continuing the execution,
- etc.

### 5.4 Debugging without debugger

In some cases, running a program with or without debugger may be different. The debugger sets pauses in the execution of the processes, during which Windows performs multiple tasks. For example, the debugger cannot be used in a procedure called by a timer, nor in the code of a Scrollbar control.

**Remark:** For more details on the limitation of the debugger, see the online help.

To debug this type of applications, you may want to monitor the changes of a value, how different procedures are called, etc.

This information can be:

- displayed on the screen.
- stored in a trace file.

**Caution:** If the information is displayed on the screen, it must be displayed during the application tests only.

#### Displaying information

Two tools can be used to display information:

- **the dialog boxes:** WLanguage **Info** function.  
Caution: When displayed, dialog boxes block the application.
- **the trace window:** WLanguage **Trace** function.  
The trace window appears in the upper-left corner of the screen, without interrupting the program.

#### Displaying the debug information

Displaying the debug information on the screen is useful in test mode only.

Any unsuitable display must be removed before distributing an application.

To avoid any oversight, it is recommended to manage how the debug information is displayed via a global procedure. For example:

```
PROCEDURE MyTrace(StringToTrace)
IF InTestMode() = True THEN
    Trace(StringToTrace)
END
```

In this code, depending on the result of **InTestMode**, the trace window appears during the application test only.

This procedure prevents trace windows from being displayed on end-user computers, even if they are called in the code of the application.

The call to this trace procedure is identical to the use of **Trace**:

```
MyTrace("Customer: "+ ...
Customer.CustomerNum)
```

### Creating a trace file

During long processes (batch processing, etc.), to check the operating mode of the program, you must keep a physical trace of the processes run (a text file, for example).

The following procedure allows you to define how the trace will be displayed:

- on the screen (/DEBUG parameter in command line).
- in a text file (default mode).

```
PROCEDURE MyTrace(StringToTrace)
FilePath is 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, foCrea
teIfNotFound + 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
```

**Remarks:**

- The trace file is created by default in the user data directory. The name of this file is the same as the project name. This file contains the information to trace during the program execution.  
The information is completed by the date and time of each "Trace". This allows you to detect a potential problem during the process.
- Example of trace file content:

```
01/12/2015 - 10:53:25:20
Customer name: Montgomery
```

## 6 Performance test

### 6.1 Overview

The performance profiler checks and optimizes the execution time of the application.

The principle is straightforward: When you test your application, the performance profiler keeps track of all the actions performed and the corresponding processes executed.

At the end of the test, the performance profiler displays:

- the 10 most time-consuming operations
- all actions performed in the application that was tested, sorted by duration (from the longest to the shortest action).

You can select a process to analyze its processing time and optimize it.

### 6.2 Starting the performance profiler

To start the performance profiler, go to the "Project" tab, "Audit and performance" group, expand "Analyze performance" and select "Analyze performance".

The project is automatically run in test mode. The process to optimize can be run in your application.

To go back to the editor, simply close the application.

The performance profiler displays the result of the analysis.

**Remark:** It is recommended to use the performance profiler to optimize the application (before it is distributed, for example).



### 6.3 Reading the performance profiler results

The performance profiler presents the result of the analysis in several tabs:

- the "Summary" tab shows the ten longest processes.
- the "Mapping" tab shows a graphical view of the main processes.
- the "Details" tab shows all the processes run during the test of the application (from the slowest to the fastest).
- the "Calls" tab shows the details of the operations performed in a process.

The following information is displayed for each process:

Function	Function, event, process or procedure run.
Total time	Function execution time.
Nb of calls	Number of calls made to the function (procedure, event or process).
Time 1 call	Execution time of a call to the function (procedure, event 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.

**Remarks:**

- "Full execution" represents the total execution time of the application test in the performance profiler.
- "Total for XXX Window" represents the total execution time of a given window (from the moment it is opened until closed).

### 6.4 Choosing a process to optimize

The processes to be optimized are chosen based on several criteria:


- **the process execution time.** The longest processes must be optimized.
- **the percentage of time used by the function or procedure.** The higher this percentage is, the greater the number of processes that can be optimized in the code.

**Remark:** If the process corresponds to a WLanguage function, it is fairly hard to optimize it.

## 7 Regression tests

### 7.1 Overview

To test a window from the editor:

1. Open the window to be tested.
2. Click  in the quick access buttons (or press F9). The editor is automatically minimized and the window is run.

During the test, all window features can be run. You will be able to open other windows, for example.

### 7.2 Automated tests

Each test contains a scenario that can be directly edited in the development environment. This scenario is written in WLanguage and can be modified at any time.

These tests can be run before creating the executable to make sure the application runs properly.

The following elements can be tested:

- WINDEV windows.
- sets of procedures.
- classes.
- WINDEV executables.

Each test is associated with a WLanguage code: the test scenario. This scenario can be viewed in the code editor. The code of the tests can be modified.

The tests and their code are not included in the executable and are not provided to the end users. Therefore, the number of tests has no impact on the size of the application available to end users.

For more details, see the online help (keyword: "Automated test").



# 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:

- a Framework-dependent executable,
- a self-contained executable,
- the common WINDEV Framework.

## Using a Framework-dependent executable

When using a Framework-dependent executable, the libraries are installed in the directory of each application. Therefore, each WINDEV application uses its own version of the libraries. Libraries can be renamed when you use a Framework-dependent executable.

### Advantages of using a Framework-dependent executable:

- Each application can use a different version of the Framework DLLs.
- The Framework of a given application can be updated without affecting other applications.
- The Framework can be renamed.

## Using a self-contained executable

When the Framework is included in the executable file, it is a self-contained executable. The Framework DLLs are not extracted when the executable file is run.

### Advantages of using a self-contained executable:

- Self-contained executable.
- The Framework DLLs are not extracted.

**Remark :** Depending on the features used by the application, it may not be possible to use a self-contained executable.

### Using the common WINDEV Framework

When you use the common Framework, the libraries are installed in a common directory. These libraries are shared between all the WINDEV applications installed on the same computer.

**Advantages of the common Framework:**

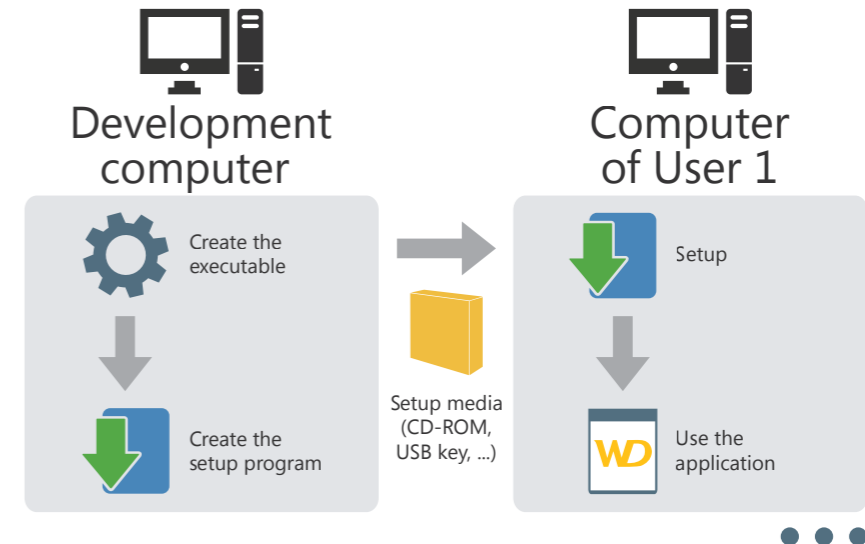
- The disk space used by the Framework (about 30 MB) is shared between all the applications.
- The Framework is updated only once for all the applications installed.
- The common Framework can be renamed.

**Remark :** When you use the common Framework, you can replace one or more DLLs (if a specific version is required, for example).

# Modes to install an application

## Individual setup

The individual setup consists in creating a setup program that will be installed on each user computer via a setup media.



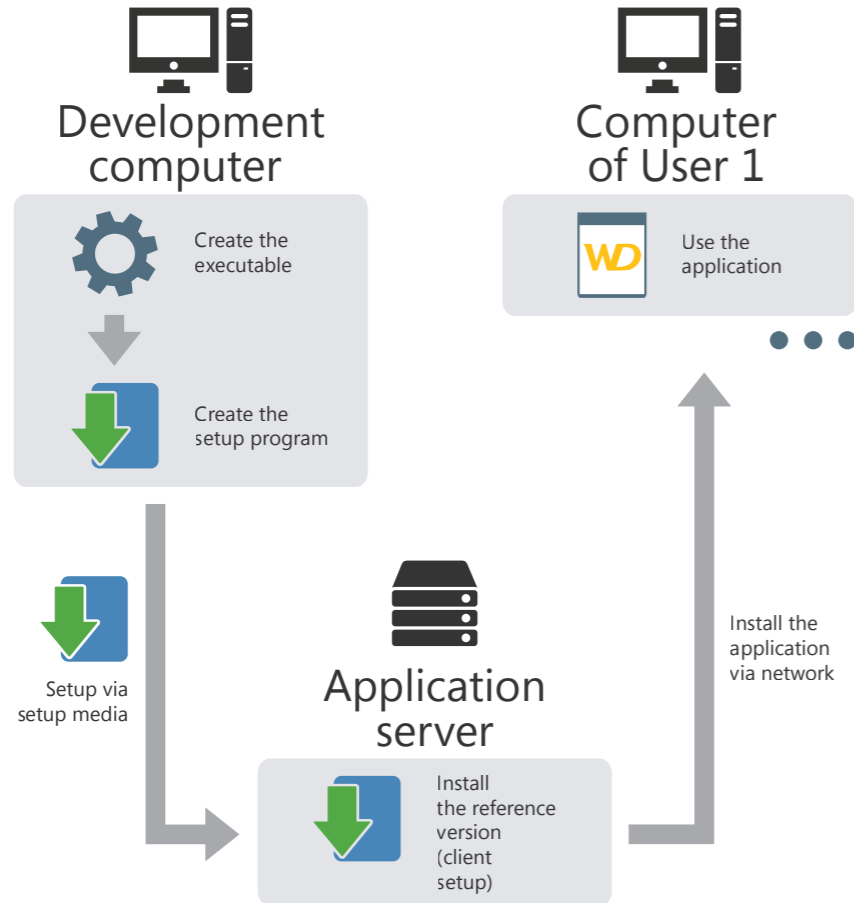
This setup mode is compatible with:

- data files in HFSQL Classic format installed on the user computer,
- data files in HFSQL Classic format installed on a 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 will install the application must start the application setup via the network.



If an update is available on the application server, users will be prompted to update the application on startup.

A network setup allows for better management (versions deployed, mandatory updates, disconnection of users, etc.).

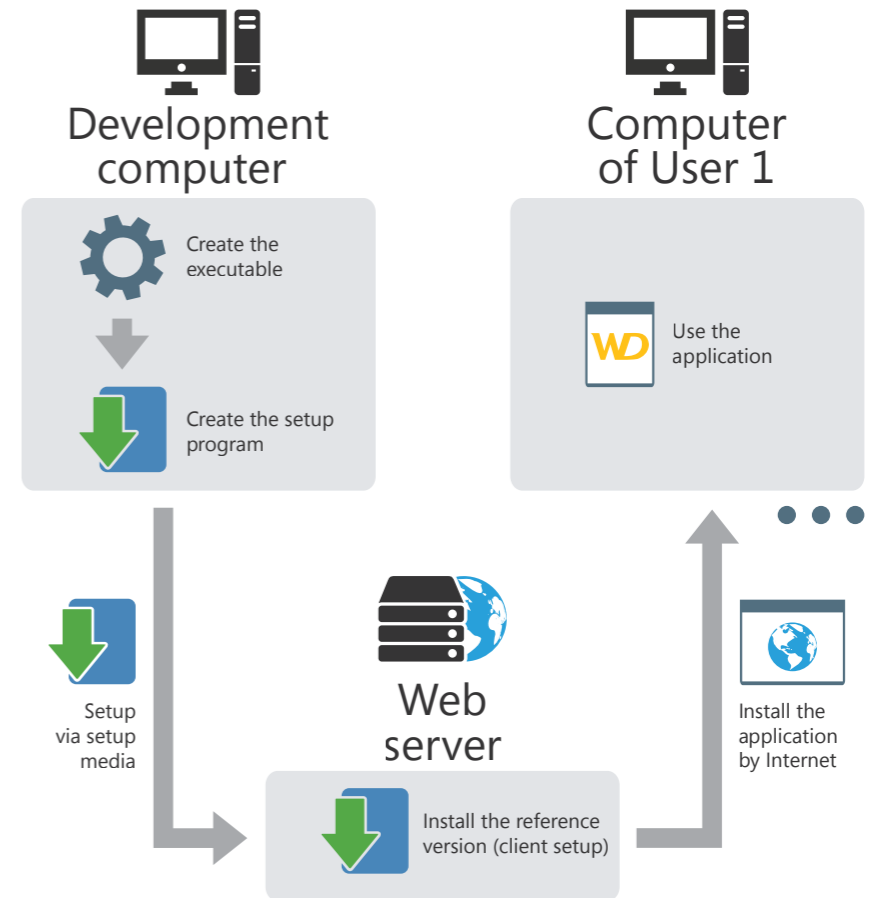
This setup mode is compatible with:

- data files in HFSQL Classic format installed on the user computer,
- data files in HFSQL Classic format installed on a 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 will install the application must start the application setup via an HTTP URL.

If an update is available on the Web server, users will be prompted to update the application on startup.



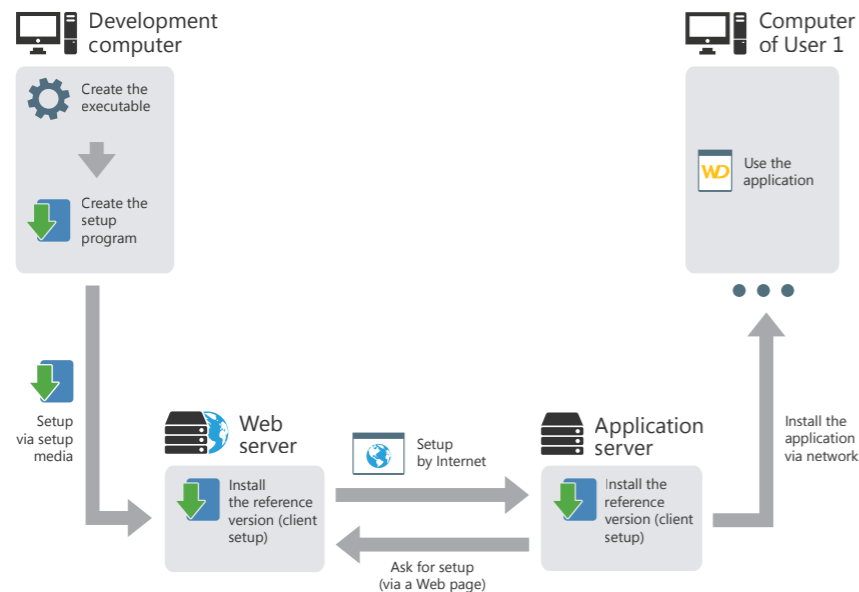
### Multisite setup (with automatic update)

The multisite setup allows you to install an application on different sites (a site can represent a subsidiary, for example).

The multisite setup consists in creating a setup program that will be installed on a Web server. All the sites that will install the application must 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 will install the application must start the setup via the network.

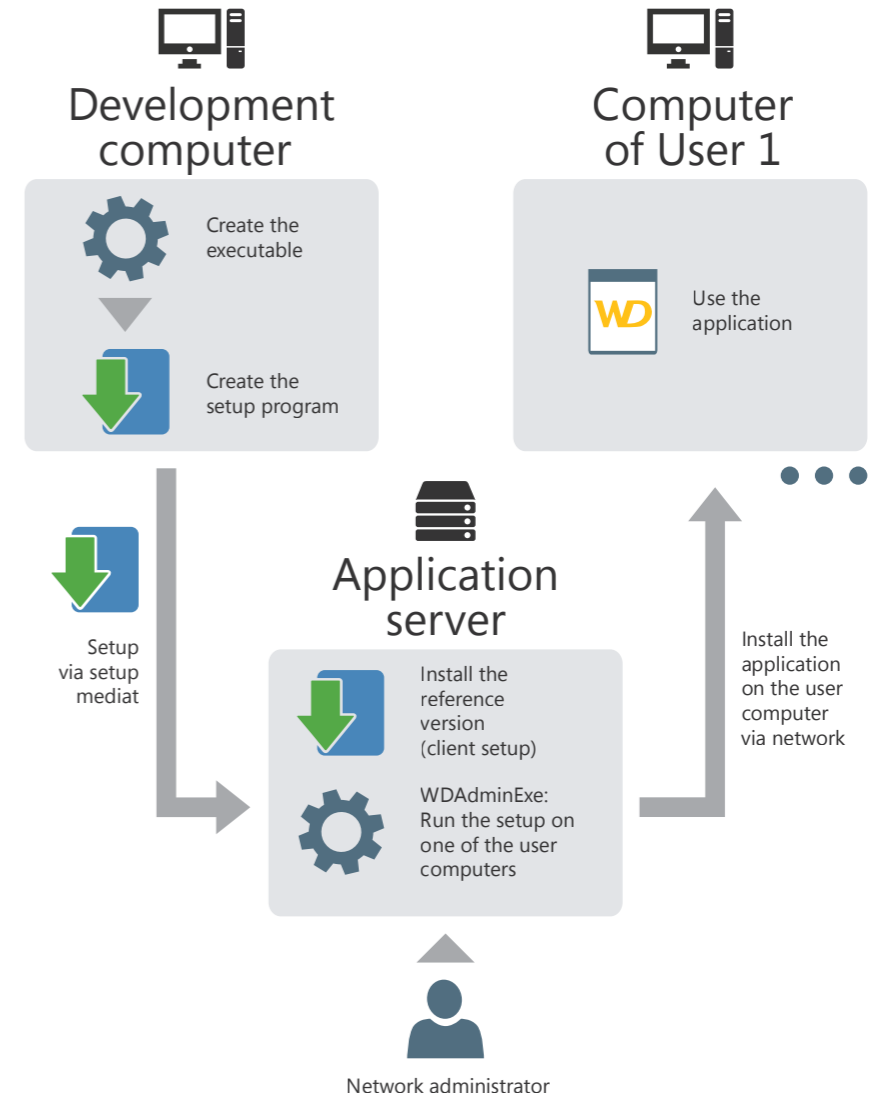
This setup combines the advantages of the network setup and the HTTP setup by allowing a large-scale deployment as well as a better control of local setups.



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. Using WD1xxAdminExe, the network administrator installs the application from the server on the client computers.



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 proposing 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.

For more details, see the online help.

## 1 Setup: Summary table

### 1.1 Setup media

Depending on the setup media used (Internet, in-house network, etc.), WINDEV offers multiple setup generation modes.

To optimize your setup, choose the recommended generation mode.

		Setup generation modes offered by WINDEV			
		Self-extracting executable	Multiple directories	Single directory	CD ROM/DVD ROM
Supports	Internet	X (recommended)			
	CD ROM/DVD ROM		X	X	X (recommended) Autorun possible
	In-house network	X	X	X (recommended)	X

## 1.2 Tools for creating the setup program

Different tools are available to create, maintain and customize the setup program.

	Wizard	WDInst setup editor	WDSETUP customization
Create a standard setup program (in French and in English by default)	X (recommended)	X	
Maintain a setup program	X	X	
Customize a setup program (the program that is started on the user computer)		X	X (multilingual setup for example)
Start mode	On the "Project" tab, in the "Generation" group, click "Setup procedure".	On the "Tools" tab, in the "Utilities" group, click "WDInst".	WDSETUP project available in the "Examples" subdirectory of the WINDEV setup directory.

### 1.3 Different types of setup

Two types of setup are available in WINDEV:

- **Individual setup:** recommended for single-user applications. The setup program installs the application on the user computer.
- **Setup with automatic update:** recommended for 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 the 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 changes 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 the application installed on the network.
  - get the characteristics of the 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 the files required to run the application (".EXE", ".WDL", ".DLL", ...) on each user computer.
    - Installing part of the database on the server and another part on each user computer. The files required to run the application (".EXE", ".WDL", ".DLL", etc.) 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 via HTTP (on a proprietary server or in the CLOUD).** A first setup of the application will be performed on an HTTP server. The users will connect to a Web page to download the application setup program. This setup is used to:
  - manage the automatic update of the application installed on an HTTP server.
  - 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 the files required to run the application (".EXE", ".WDL", ".DLL", ...) on each user computer.
    - Installing part of the database on the server and another part on each user computer. The files required to run the application (".EXE", ".WDL", ".DLL", etc.) are also installed on each user computer.
- **Multisite setup.** This setup is the combination between a local network setup and 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 setup program creation wizard directly from WINDEV (on the "Project" tab, in the "Generation" group, click "Setup procedure"). This wizard allows you to create the different types of setup programs.
  - use the WDIInst setup editor (on the "Tools" tab, in the "Utilities" group, click "WDIInst"). This editor is used to customize the setup program.
- **provide other files**, you must use the setup program creation wizard, started from the setup editor WDIInst ("File .. New" from WDIInst). Reminder: To start WDIInst, go to the "Tools" tab, "Utilities" group and click "WDIInst".

### 2.2 Installing a WINDEV application

#### Standard, compact or MSI setup

To install your application, simply start the application setup program on the user computer. The files required by the application are installed in the specified setup directory.

To install an update of your application, simply run the update setup program on the user computer. The application files modified by the update are replaced.

#### Setup with automatic update

To install your application:

1. Start your reference version setup program from a user computer. 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 reference version setup update from a user computer. 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 reference version setup program from a user computer. 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 reference version setup update from a user computer. 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.



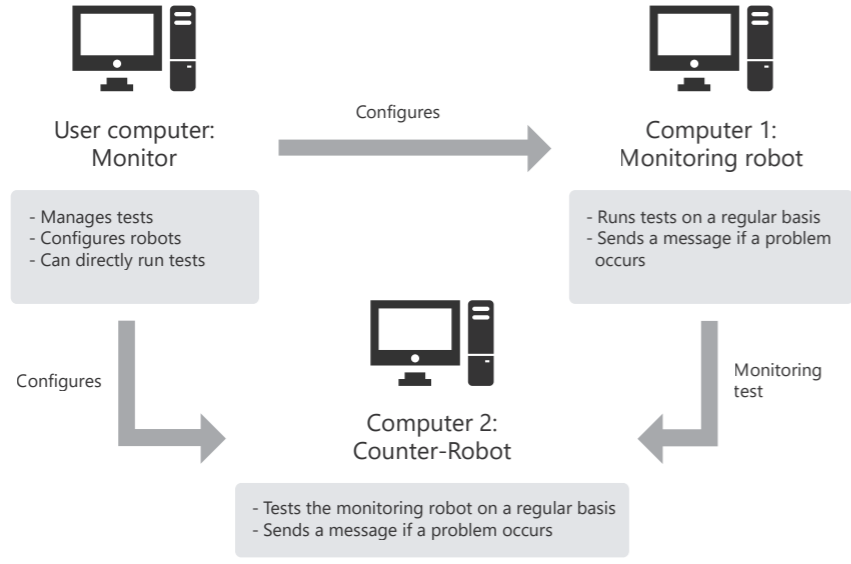
# Monitoring your applications

An application depends on several external parameters: Website, HFSQL server, network, etc. To optimize the management of incidents, WINDEV allows you to use a monitoring robot. The monitoring robot, which is composed of three executables running on different computers, is used to run different tests: Internet connectivity check, network tests, etc.

If a problem occurs during a test, the monitoring robot sends you a message using one of the following methods:

- Message sent via PC SOFT's messaging tool (WDBAL).
- Email.
- Via a third-party program.

Furthermore, a sound alert can be implemented on the monitor.



### Available types of tests

Internet test (test of a Web site)	Test of HFSQL Client/Server servers	Network test (SNMP)
Test of FTP server	Test of news server	Test of WINDEV application
Custom test (in WLanguage)	...	

## Communication





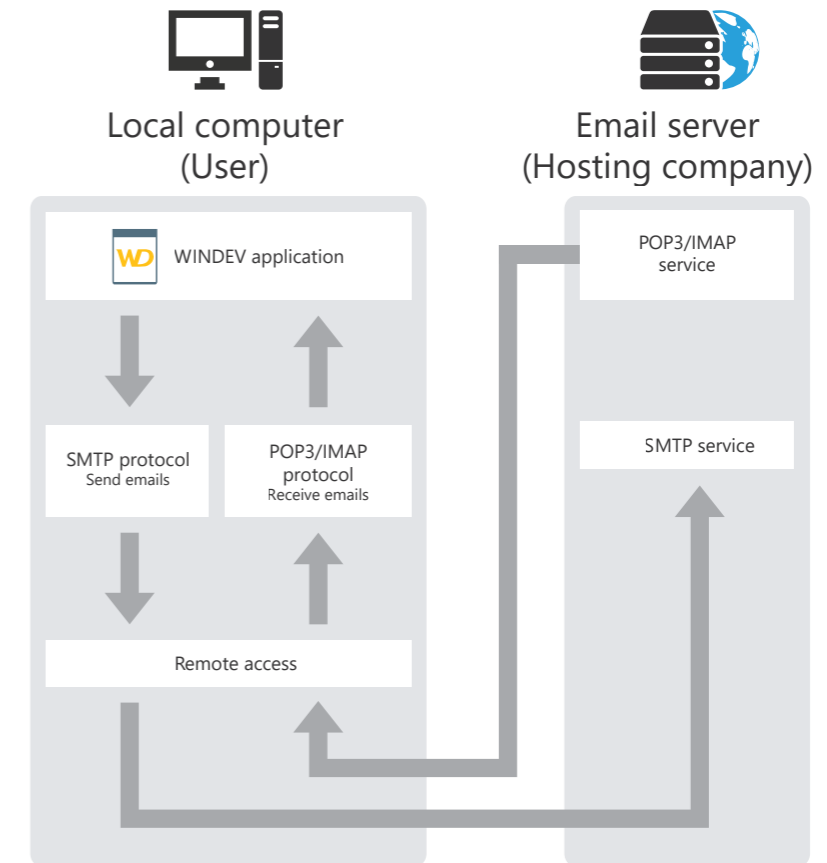
# Managing emails

WINDEV includes multiple methods to manage emails:

- POP3 or IMAP and SMTP (most common method).
- the "Simple Mail API (also called SMAPI or Simple MAPI)".

## POP3 or IMAP and SMTP

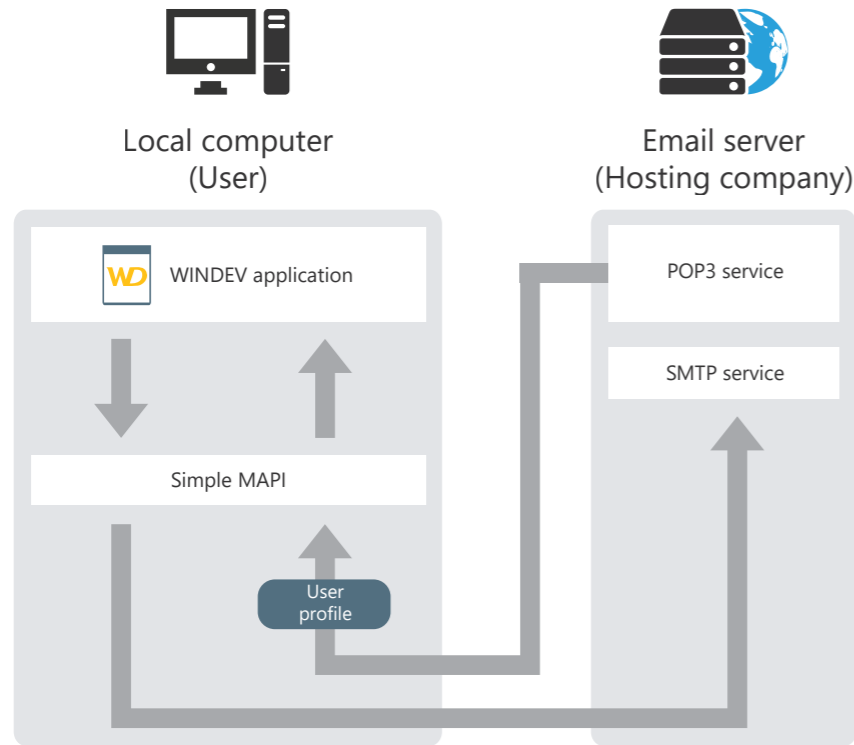
These are the standard Internet protocols used to manage emails. They are standardized by the RFC 1939, 3501 and 5321 documents. These protocols allow you to directly communicate with the server, available at your ISP.



- 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 email management mode is used by most Microsoft applications, especially MS Exchange.



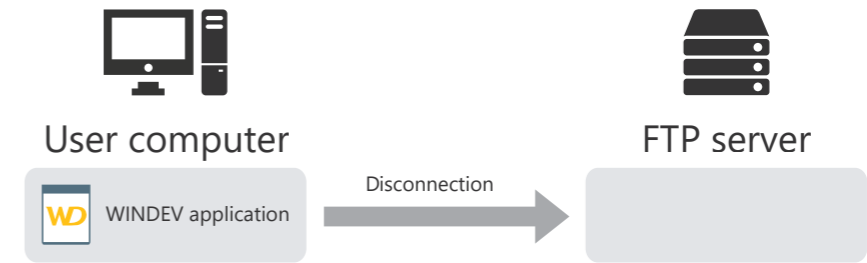
## Transferring files via FTP

FTP (File Transfer Protocol) is a protocol for transferring files from one site to another. This protocol allows you to exchange files via TCP/IP or the Internet.

### Principle

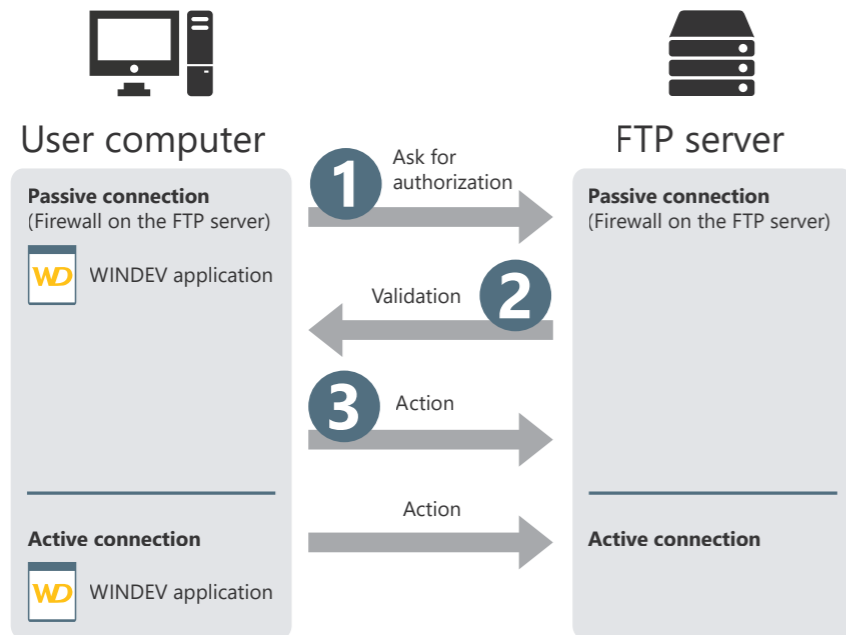
To handle files on an FTP server from a WINDEV application, you must follow these rules:

1. Connect to an FTP server (FTPConnect).

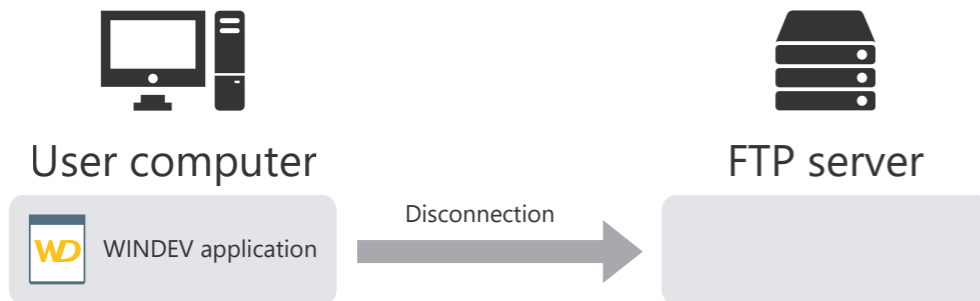


2. Send and retrieve files (according to the connection mode). For a passive connection, the WINDEV application must request the authorization from the FTP server before any operation is performed on the server files. You can also:

- get the characteristics of the files located on an FTP server: attributes, size, etc.
- handle files on an FTP server: create, delete...
- list files from a directory located on the FTP server by running a procedure that starts a process for each file found.



3. Close the connection to the server (**FTPDisconnect**).



# Managing sockets

WINDEV includes advanced socket management functions.

A socket is a communication resource used by applications to communicate between two computers regardless of the type of network.

This communication mode can be used, for example, to establish a communication between computers connected by Internet.

## Different modes

WINDEV applications can manage sockets according to different modes:

- **WINDEV Client application:** the application connects to any server and exchanges data via a socket.
- **WINDEV "Simplified server" application:** the WINDEV application is a server that exchanges information via a socket with a single client computer.
- **WINDEV "Standard server" application:** the WINDEV application is a server that exchanges information via a socket with multiple client computers.

## How a client application works

A client application of a socket server connects to a standard server to exchange information via a socket.

Example: A WINDEV client application can connect to a standard news server on the Internet.

### Step 1: Connecting to the server

To connect to a server socket, use **SocketConnect**. This function makes a connection request to the server. The socket is identified by its port and address.

### Step 2: Exchanging data

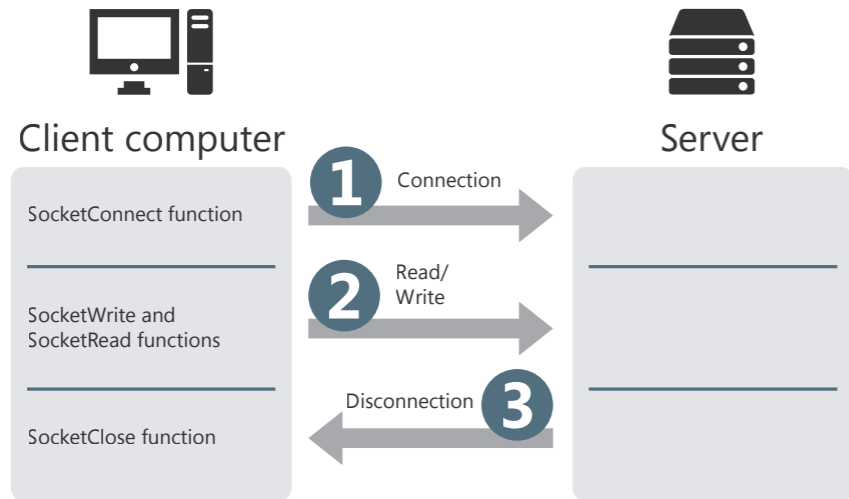
Once two machines have connected their sockets, a communication channel is established between them. These two machines can read from and write character strings to the socket.

To read from and write to the socket of the server, the WINDEV client application must use **SocketRead** and **SocketWrite**.

### Step 3: End of communication

To end the communication, close the socket from the client machine with **SocketClose**.

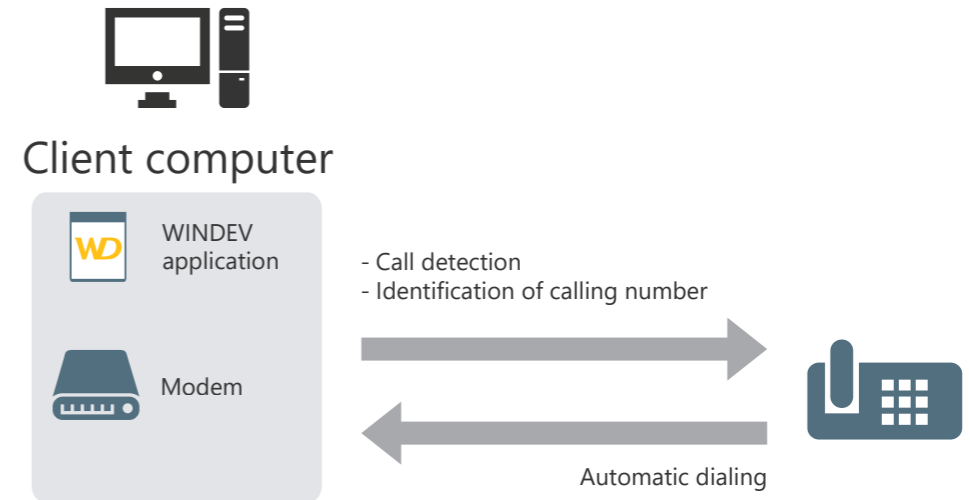
The different steps can be represented as follows:



**Remark:** the SocketXXX functions can also be used to manage secured sockets according to the SSL protocol. To do so, use **SocketCreateSSL** and **SocketConnectSSL**.

# WINDEV and telephony

WINDEV allows you to easily manage incoming and outgoing phone calls via the WLanguage phone functions. These functions allow you to manage a voicemail, an answering machine (or other) directly in a WINDEV application.



**Required configuration:**

To use the telephony features, you must have:

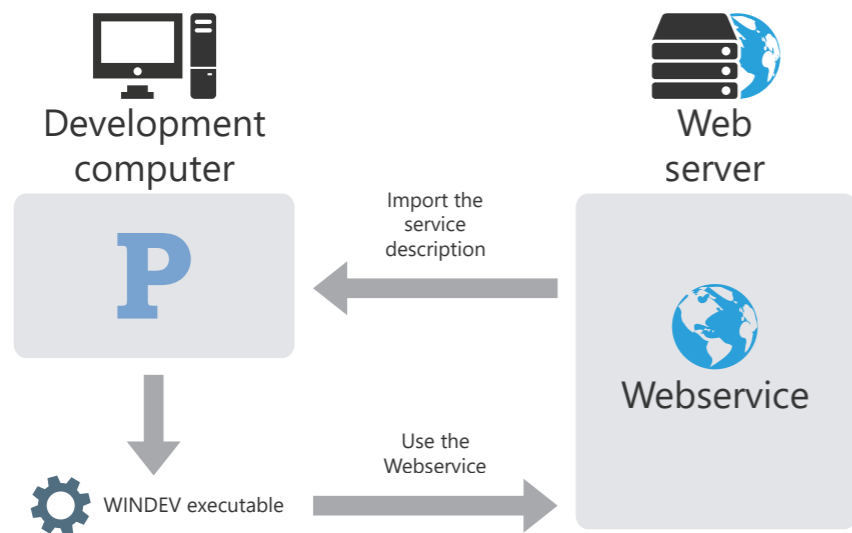
- a modem.  
To save and play recordings, this modem must include voice support.  
To identify the caller, the modem must support Caller ID.
- the TAPI 3.1 protocol.

# Webservices: concepts

WINDEV allows you to easily use Webservices (or XML Web services). A Webservice is an application available on the Web. WINDEV allows you to access both SOAP (Simple Object Access Protocol) and REST (Representational State Transfer) Webservices.

Any WINDEV application can use an existing Webservice.

To use a SOAP Webservice, you just need to get the Webservice address and import its description into the application.



To use a REST Webservice, you only need to get the Webservice characteristics and use **RESTSend**.

**Remark:** you can also provide one or more sets of procedures of a WINDEV application as a SOAP or REST Webservice. To do so, you must:

- create a specific project configuration,
- for a REST Webservice, create a description of the REST Webservice,
- generate the Webservice,
- deploy it on a WEBDEV Application Server.



# Components provided with WINDEV

WINDEV is provided with projects that contain components. These projects include:

- a project configuration corresponding to the component.
- a project configuration corresponding to the use of the component.
- an internal component.

The list of the main components provided with WINDEV is as follows:

WD BGInfo	This example explains how to write information on the background screen of a computer (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: <ul style="list-style-type: none"> <li>• On remote computers. The communication is performed via sockets.</li> <li>• On the same computer. The communication is performed via the Windows messages.</li> </ul>
WD Email address autocomplete	Used to simplify the input of emails.
WD EventLog	Used to: <ul style="list-style-type: none"> <li>• List and read the events saved in the event log of Windows.</li> <li>• Add an event into the event log of Windows.</li> </ul>
WD Explorer Bar	Used to manage an address bar similar to the address bar of the Windows Vista explorer.
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	Used to display temporary information to the user without interrupting his work.
WD Glossary	Used 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 customer himself All the glossary entries can be organized in tree-like categories.
WD HTML Page 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 Limitation	Used to limit an application by different methods: <ul style="list-style-type: none"> <li>• Time limitation (the application can be used during a 30-day period).</li> <li>• Limitation to a single computer (the application can be run only on a single computer).</li> <li>• Limitation on the number of users (the application cannot be run by more than n users).</li> </ul>
WD Managing Logs	Used to define, for an application, a configuration for the runtime logs: <ul style="list-style-type: none"> <li>• number of log files that must be stored,</li> <li>• maximum size for each log file,</li> <li>• information that must be logged.</li> </ul> This example simulates different processes to observe the management of log files (.wlog files).
WD Photocopier	Used to photocopy documents via a scanner and a printer.
WD PrintModel	Used to manage print profiles.
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: <ul style="list-style-type: none"> <li>• The paper orientation,</li> <li>• The scale,</li> <li>• The management of TrueType fonts,</li> <li>• The number of copies,</li> <li>• The management of color (color or black/white print),</li> <li>• The double-sided,</li> <li>• The output style (sheets grouped by copy or by page number).</li> </ul>
WD PrintSpooler	Defines the list of less used printers in order to perform one or more prints.
WD Progress bars	Used to automatically manage a simple or dual progress bar that displays the remaining time or the number of process steps.
WD Select RTF	Used to perform all types of searches in an RTF control. You have the ability to search for: <ul style="list-style-type: none"> <li>• the words in bold,</li> <li>• the words in italic,</li> <li>• the words of a specific color,</li> <li>• a character string containing specific words.</li> </ul>

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	Used to perform calculations from the Simplexe algorithm.
WD Sockets	Used to establish a communication between two applications by using sockets.
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.
WD Touch Screen	Used for interfacing a tactile screen that can replace the standard keyboard. This keyboard allows you to use a WINDEV application on a computer not equipped with a keyboard.
WD Trace Socket	Allows you to trace a communication via socket, on a given address, on a given port. This allows you to understand the principle of dialog via socket between two applications. In the applications that dialog via socket, you will be able to see all of the exchanges performed on the socket even if no trace option was specified by the developers. This example also allows you to better understand the operating mode of network protocols based on sockets: POP3, SMTP, FTP, HTTP, etc.

# Examples provided with WINDEV

The examples provided with WINDEV are intended to help you learn the features of WINDEV. Their source code is clear and includes comments.

WINDEV is provided with multiple examples:

- Full examples: these examples correspond to full applications that can be used without any adaptation.
- Training examples: these examples illustrate a specific feature.

These examples can be opened directly from the WINDEV home window:

- If the example is not installed on the computer, it will be automatically downloaded (from a PC SOFT server) and opened.
- If the example is installed on the computer, a copy is automatically created in "My projects\My examples".

The open example corresponds to the copy of the original example. This allows you to work or make changes to this copy. The original example can be kept unchanged. Every time you open the example (via the "Open an example" option), you can work on the copy or on the original version.

Let's see the features of some examples provided with WINDEV.

## Full examples

WBSSetup	Used to create customize the setup by physical media of a WEBDEV site.
WD Burner Lite	Used to burn data CDs, DVDs or Blue-Ray disks.
WD Burner	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 FTP File Transfer	This example is a full FTP client allowing you to store several FTP servers. Then, you have the ability to perform multi-file transfers from the local computer to the server or from the FTP server to the local computer. To do so, we are using the standard functions of WLanguage (FTPConnect, etc).
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 Live Messaging	Presents the functions for managing the sockets.
WD Loan	Enables you to simulate loans and to: <ul style="list-style-type: none"> <li>• calculate the amount of monthly payments according to the amount of the loan.</li> <li>• calculate the amount that can be borrowed according to the amount of a monthly payment.</li> <li>• calculate the interest on investment from a monthly deposit.</li> </ul> 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 Manager of task DotNet	Used to list the applications and processes via the .NET functions. Used to perform different processes on the applications and on the processes.
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: <ul style="list-style-type: none"> <li>• videos (AVI, MPEG, ...),</li> <li>• sound files (MP3, WAV),</li> <li>• audio CDs,</li> <li>• Flash animations.</li> </ul>
WD Office Image Editor	This example presents the operating mode of Image Editor control of WINDEV.
WD Office PDFReader	This example presents the operating mode of PDF Reader control of WINDEV.

WD Office Spreadsheet	This example presents the operating mode of Spreadsheet control of WINDEV.
WD Office Word Processing	This example illustrates the operation of the Word Processing control of WINDEV.
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: <ul style="list-style-type: none"> <li>• Messaging,</li> <li>• Calendar,</li> <li>• Contacts,</li> <li>• Tasks.</li> </ul> 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 Schedule	Used to create a schedule containing click areas.
WD Simple Internet Browser	This example presents the features of WINDEV’s HTML control and includes a simplified web browser with browsing history, favorites, etc.
WD Telephony	Uses the telephony functions of WINDEV: <ul style="list-style-type: none"> <li>• To dial a phone number,</li> <li>• To detect and identify the incoming calls.</li> </ul>
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 WebCam	Presents the integration of a Web Camera in a WINDEV window. You can: <ul style="list-style-type: none"> <li>• View the image broadcasted by the camera,</li> <li>• Perform an immediate capture,</li> <li>• Capture the scene broadcasted as a Windows animation (“AVI” file).</li> </ul>
WD Zip	Used to create and read compressed archives in “WDZ” format (WINDEV ZIP).
WDSETUP	This example explains how to create a custom setup procedure for your application with WINDEV.

### 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 the Bluetooth functions: <ul style="list-style-type: none"> <li>• to list the accessible Bluetooth devices,</li> <li>• to list the services proposed by a device,</li> <li>• to connect to a Bluetooth device,</li> <li>• to send a file to a device via the “OBEX” functions of WLanguage.</li> </ul>
WD ChainedList	Supplies the tools for building chained lists with simple chaining. A list with simple chaining is used to store and browse in organized way a set of ordinary values.
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 LibreOffice	Used to control LibreOffice via OLE Automation. All the LibreOffice functions can be used.
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 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: <ul style="list-style-type: none"> <li>• the default operating mode,</li> <li>• the customization options.</li> </ul>
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 double-sided printing that can be disabled, which means that the back sides may not be printed.
WD Duplication	This educational 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: <ul style="list-style-type: none"> <li>• the test of a window.</li> <li>• the test of a global procedure.</li> </ul>
WD Extended Map	This example shows the different features of the Map control in desktop (Windows) and mobile (Android, iOS) devices
WD Flowchart	This example is used to create organization charts.
WD Gallery Control	This example demonstrates the use of the "Image Gallery" control.
WD Graphic Objects	This example is an editor of graphic objects. You can: <ul style="list-style-type: none"> <li>• move, resize the objects,</li> <li>• perform selections with the lasso or with the mouse,</li> <li>• cut/copy/paste the objects,</li> <li>• delete/duplicate the objects,</li> <li>• save the created documents and print them..</li> </ul>
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	Visually represents the details of the execution plane of a SQL query via SQL EXPLAIN.
WD HotKey	Used to define your own keyboard shortcuts.



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	Presents the use of WinEdXXX function 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 that can be used in an Internet page. This applet is used to read and send emails.
WD JSON	This example shows how to use the JSON format with WINDEV: type of variable, type of item, import/export, ...
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 training example explains the operating mode and the implementation of the mechanism for managing the WINDEV errors.
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 Multi-Windowing	Used to understand the principle of multi-windowing in WINDEV.
WD Network frames	The analysis of the frames that circulate on the network allows the administrators to audit the network and to detect inconsistencies. To listen to the network, the application is based on the Winsock 2 API and on the RAW Socket.
WD OAuth	The example proposes to connect to Google and Microsoft via the AuthIdentify function and the OAuth2Parameter type.
WD Object Detection	This example shows how to use AIDetect.
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 UA	This example shows how to communicate with an OPC UA server.
WD OPC	Presents the implementation of OPC protocol in WINDEV.
WD OpenGL	Allows you to interface OpenGL and WINDEV. OpenGL is a software interface used to draw three-dimensional objects (3D).
WD OSD	OSD means: On-Screen Display. It is the name given to the interfaces that appear on the computer or television screens for example. They are often used to perform settings. These types of menus are displayed above all the 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: <ul style="list-style-type: none"> <li>• the data found in a WINDEV application such as the data found in tables, charts.</li> <li>• and from a base PowerPoint document (.pptx).</li> </ul>
WD PowerShell	This example allows you to run a powershell script from a WINDEV application and retrieve the result and the report.
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 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: <ul style="list-style-type: none"> <li>• prints based on different data sources (queries, variables, ...).</li> <li>• prints based on controls (Table, Spreadsheet, PVT, ...).</li> <li>• printing composite reports.</li> <li>• specific prints (portrait/landscape, report with watermark, report with bar code, ...).</li> </ul>
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 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 Toasts	This example shows how to use the ToastDisplay and ToastDisplayIW functions as well as Toast variables. A Toast is a temporary message used to inform the user without interrupting the application.
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: <ul style="list-style-type: none"> <li>• View with ability to perform modifications (read/write),</li> <li>• Read-only,</li> </ul> The triggers are used to check the access to the files according to the selected mode.
WD Two-factor Authentication	This example shows how to use two-factor authentication to secure access to user accounts. It uses TOTP authentication, which generates a 6-digit code every 30 seconds.
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: <ul style="list-style-type: none"> <li>• analyze a character string,</li> <li>• configure the dialog boxes displayed.</li> </ul>
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 WdSETUP Plugin	This project contains the elements required to create a setup plugin (or “extension”) for WdSETUP.
WD Webservice Client	This example illustrates the use of WebServices.
WD Webservice Server	This example illustrates the use of the WebServices.
WD Website automation	This example shows how to automate a website using WINDEV’s HTML Control and the WLanguage HTMLxxx functions
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 on our website ([www.windev.com](http://www.windev.com)).