



Express



DEVELOP 10 TIMES FASTER

PCSOFT

WinDev - Concepts
Express Version 17 - (1) 06-2012

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

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

This documentation is not contractually binding. Some 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 2012: This publication may not be reproduced in part or in whole in any form without the express permission of PC SOFT.

In which order should these guides be read?

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

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

- 1** Reading the "Concepts".
This guide presents the main concepts required to create a high-powered WinDev application. Some concepts are followed by a "Practical" section that presents some features of the editor.
- 2** "Tutorial" (guide + exercises)
The Tutorial provides a first "hands-on" approach to WinDev. It enables you to familiarize yourself with the main editors of WinDev.
- 3** Test of the examples
Run the test of the examples supplied with WinDev in the fields you are interested in (sales management, scheduling, ...)

The "WLanguage" book presents the programming in WLanguage. For each programming theme, you will find a description of the associated feature and the list of corresponding WLanguage functions.

The online help, accessible by Internet from <http://doc.pcsoft.fr> or installed with WinDev, allows you to easily find the syntax of a WLanguage function, to get help about the interface ...

Note: If there is a difference between the guide and the online help, follow the instructions given in the online help.

We hope you enjoy getting started with WINDEV.

Organization of the guide

This guide presents the main concepts required to create a high-powered WinDev application. Some concepts are followed by a "Practical" section that presents some features of the editor.

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

Internal component

An internal component is used to group several project elements. This grouping allows you to:

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

SCM

Project 1 | Project 2

1 Create the internal component

2 Share the internal component via the SCM

Windows
Classes
Queries
...

Sub-directory of development project | WCI file (List of elements)

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

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

Part 2: Development environment | 67

Concept page

Internal component in practice

1 Overview

An internal component is used to group several project elements. This grouping allows you to:

- Organize a project: you can create internal components to group the elements of a project by feature for instance.
- Share elements among different projects: elements found in an internal component can be shared among several projects via SCM. (Sharing the internal components via SCM): page 68 for more details.

One of the benefits of the internal component consists in an internal component is that the internal component can be displayed directly from the project that uses it.

Differences with an external component: an internal component allows you to include all the elements of the component in the interface of the project containing the internal component. 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, the corresponding project must be opened.

2 Creating an internal component

2.1 The different steps

To create an internal component:

- Select "Workshop - internal component - Creating an internal component". This wizard for creating an internal component starts.
- Specify the characteristics of the internal component:
 - Its name. The name of the internal component will be used for the WCI file corresponding to the description of the internal component. This name will also be used to create a sub-directory in your project containing all the elements of the internal component.
 - Its location.
 - Its description.
- Indicate the elements that must be included in the internal component. An internal component can contain all the types of elements found in a project: windows, queries, variables, and so on.
- Specify the elements to be included in the internal component that will be directly accessible in the code and in the greater scheme of the project that uses the internal component. Note: the elements or "public" elements of the internal component will be affected automatically by the compilation features. They can be used in the elements found in the project or from another internal component. The inaccessible elements or "private" elements can only be used by another element of the internal component (the sub-component).

This mode is reserved to specific cases.

- Full autonomy:** the internal component accesses its own data files. The internal component and the project can interact with runtime controls. This mode corresponds to the default mode if the component uses no archive.
- In this case, the archive used by the internal component must be specified. You also have the ability to create a new archive directly. This archive will be associated with the internal component.
- Validate the creation of the internal component.** You will then be able to:
- Modify the characteristics of the internal component on the description window of the internal component.

68 | Part 2: Development environment

Implementation page

Summary

PART 1 - BASIC CONCEPTS

Project and Analysis.....	9
Development cycle of an application.....	14
GUI: The windows.....	16
Modal, non-modal or MDI windows	17
Internal window	19
Window templates.....	20
The different types of standard controls.....	34
Control templates.....	48
Edit: The reports.....	49
Report templates	50

PART 2 - DEVELOPMENT ENVIRONMENT

The 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
The Control Centers	85
Managing the requirements.....	86
Project Monitoring Center.....	87
Managing the suggestions and incidents	88
Managing the business rules.....	89

PART 3 - DATABASES

Analysis: Structure of the database	93
The different types of accessible files	104
HyperFileSQL Classic	106
HyperFileSQL Client/Server.....	107
HyperFileSQL Cluster	108
HyperFileSQL: the files physically created.....	109
Associating controls and data.....	110
Queries.....	112
Embedded queries.....	113
The Table/Looper controls	114
Universal replication	116
3-tier architecture	117

PART 4 - DEVELOPING APPLICATIONS: ADVANCED CONCEPTS

RAD RID121
The UML model 122
User groupware 129
Multilingual applications139
Optimizing an application.....144
Distributing "Reports and Queries"146
UMC: User Macro Code..... 147
The different available tests.....149
The debugger150
Unit tests Automatic tests152
Unit tests on the executable153

PART 5 - SETUP

The WinDev framework.....163
The different modes for installing an application.....164
Monitor your applications.....173

PART 6 - COMMUNICATION

Managing the emails 177
Transferring files by FTP.....179
Managing the sockets181
WinDev and telephony.....183
The Webservices 184

PART 7 - APPENDICES

Components supplied with WinDev187
Examples supplied with WinDev191



PART 1

Basic concepts



Express



DEVELOP 10 TIMES FASTER

PCSOFT

Project and Analysis

The development of a **Windows application** with WinDev relies on two main elements: the Project and the Analysis.

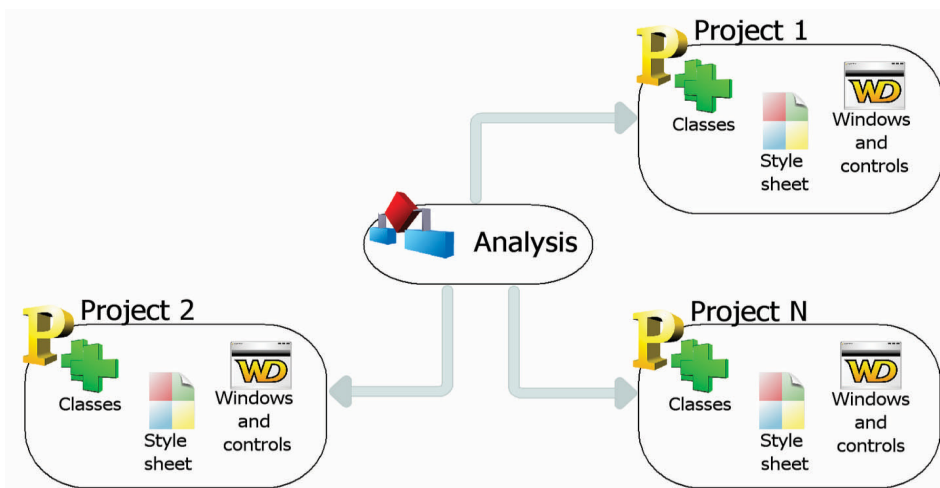
A **WinDev Project** is a set of elements: windows, reports, controls, classes, components, ... whose combination allows you to build a Windows application.

A **WinDev analysis** groups the description of the data files found in the application.

An application is built from a project.

In most cases, a project is associated with an analysis.

An analysis can be associated with one or more projects.



The project in practice

1 Overview

The first step when describing an application consists in creating a project.

When creating a project, several questions will be asked to you in order for the project to correspond to your expectations. All the characteristics spe-

cified when creating the project can be modified later.

Note: The analyses are presented in detail in the chapter "Analysis: Structure of the database", page 93.

2 Creating the project

To create a project:

1. Select "File .. New .. Project". The project creation wizard comes up.

2. Specify the different options of the project:

- **the name and the location.** These options cannot be modified. The project corresponds to a ".WDP" file. All the elements associated with the project will be created in the specified directory.
- **the type of generation.** Several choices are offered: executable, component, Java archive, These options will be presented in details later. In most cases, a project is used to generate an executable.
- **whether the project will be handled by several developers.** You can use the developer groupware or the source code manager (SCM) to share your project.
- **the programming charter.** This charter is used to automatically prefix the variables, the names of controls and windows, ...
- **the style book.**

• **the supported languages.** These languages are offered by default as soon as a control option, a window option, a report option, and so on, can be translated.

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

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

Notes:

- **Caution:** The information specified when creating a project can change according to the use mode of WinDev. The current mode is displayed in the toolbar, below the logo of the product. To modify the use mode, all you have to do is click the current mode ("Full mode" for example).
- If you have requested the creation of a database, the corresponding wizard is automatically started.
- The specified information can be modified in the project description ("Project .. Project description").

3 Project dashboard and project graph

3.1 Dashboard

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

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

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

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

The dashboard gives an answer to all these wishes. The dashboard includes several lights that provide a global view of the different indicators. The lights switch from green to red whenever a section of the project requires special attention.

To display your project's dashboard, all you have to do is select "Project .. Project Dashboard".

For more details, see "Project dashboard", page 57.

3.2 Project graph

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

To display the project graph, all you have to do is select "Project .. Project graph".

3.3 The project elements

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

To find out the list of project elements, select "Project .. List of project elements".

This option enables you to:

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

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

4 Operations performed on a project

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

- Archiving a project.
- Restoring a project.

- Duplicating a project.
- Copying or deleting a project.
- Renaming a project.

See the online help for more details.

5 Project documentation

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

- "Project .. Print the project documentation". The documentation can contain all the characteristics of the project.
- "File .. Print the documentation". In this case, the documentation contains the characteristics of the current element (window, report, query ...).

5.1 Type of documentation

Several types of documentation are proposed:

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

- Full documentation: Contains all the information about the project. This documentation combines all the other types of documentation.

Edit mode of the documentation

The documentation can be:

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

5.2 Print areas

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

To specify the print areas of the current graphic representation:

1. Select "Display .. Print areas". Borders representing the areas that can be printed in the documentation are displayed in the current editor.
2. Reduce the display of the graphic representation ("Display .. Zoom" or [Ctrl] + Mouse wheel) to view the entire graph.

3. With the mouse:

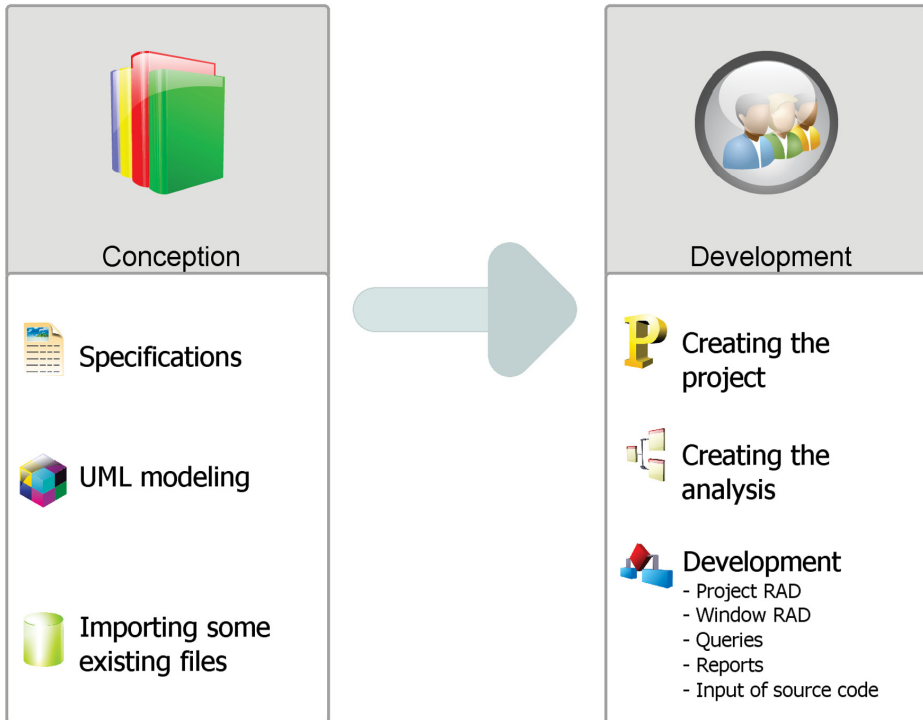
- moves these borders to the requested location by keeping the mouse button down (the mouse cursor is black).
- choose the number of pages on which the graphic representation must be printed (the mouse cursor turns into a North-West/South-East double arrow).

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

1. Define the print areas.
2. Display the options for documentation layout ("File .. Documentation layout").
3. Select "Resize the print areas of the opened documents".
4. Print the documentation.

Development cycle of an application

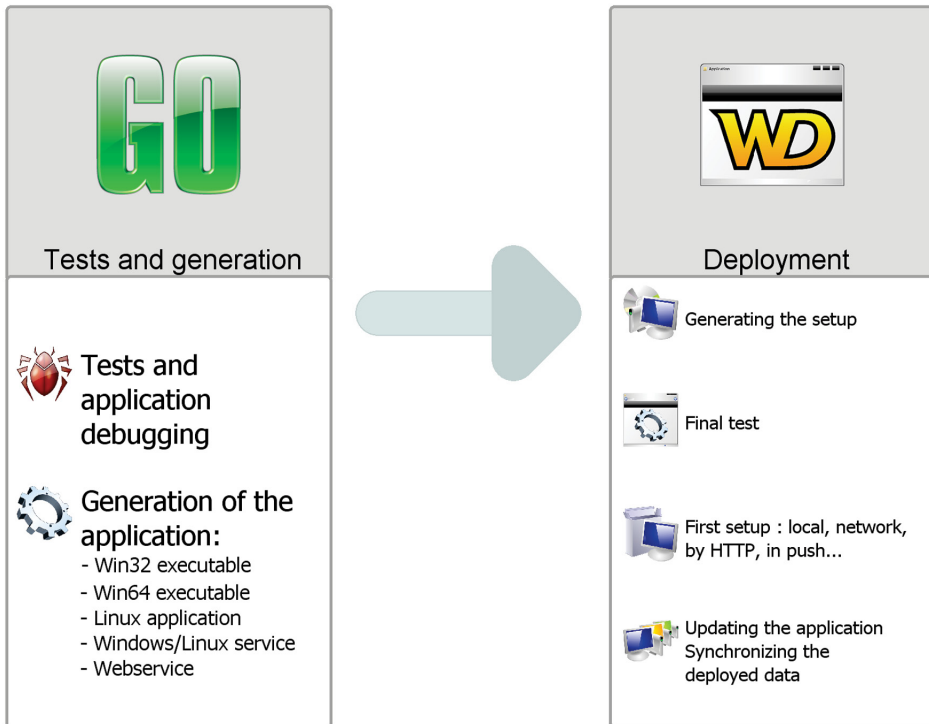
WinDev covers the entire development cycle of an application:



Details of the different steps:

Conception step: You have the ability to design an application from requirements, from a UML model of the processes or even from existing data files.

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



Test and generation step: WinDev offers several tools for automatic test to guarantee the reliability of the applications and to make sure that there is no regression during the development phase.

Deployment step: The deployment of a WinDev application can be done according to several methods: download, via a local area network (including Push mode) or via Internet. In any case, the HyperFileSQL data files will be automatically updated according to the evolutions made in the application.

GUI: The windows

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

WinDev includes an advanced window editor allowing you to easily and quickly build all possible types of GUI.

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

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



Modal, non-modal or MDI windows

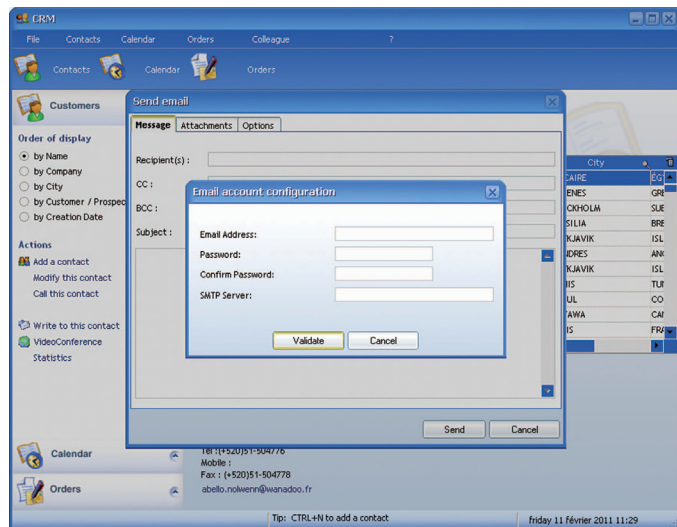
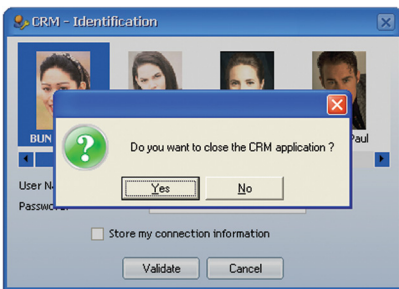
The applications created with WinDev can use several types of windows (and mix each one of these types).

Modal windows

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

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

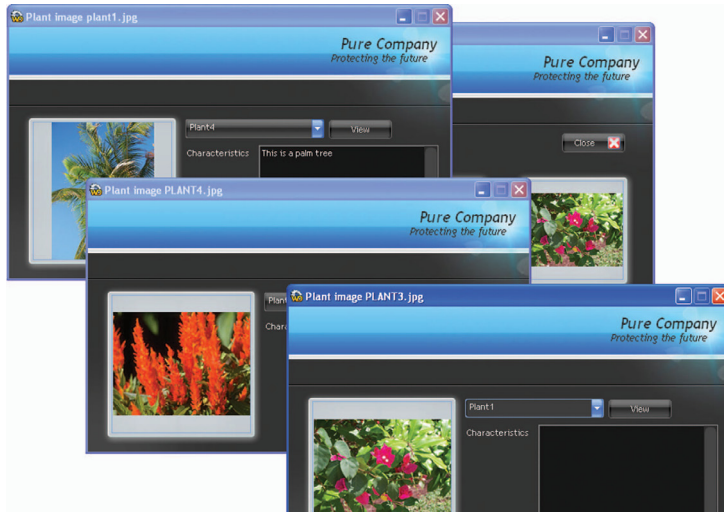
With WinDev, when a modal window is opened, a gray shadow appears on the windows located in the background: this is the DDW feature or "Dim Disabled Windows".



Non-modal windows

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

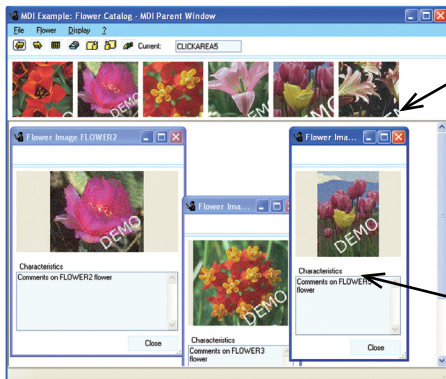
- as tool palettes,
- as dialog boxes whose modification directly affects the main window.



In this example, each description window of a flower is used to open the description window of another flower. All the 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. Two types of windows can be found in the MDI applications:



MDI parent window: All 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 enter information in these windows.

Internal window

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

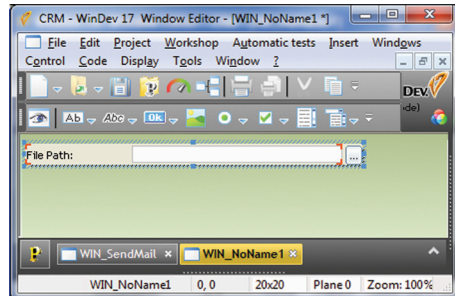
1. Creating an internal window.

To create an internal window, select "File .. New .. Window .. Internal window".

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

All types of controls can be used in this window.

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



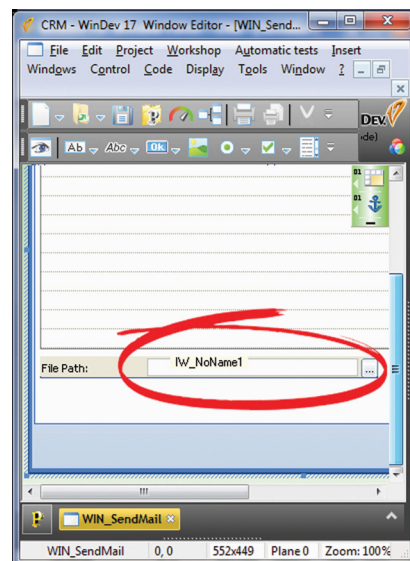
2. Using an internal window.

To use an internal window, you must:

- create an internal window control.
- in the control description, select the internal window to use and validate.

Notes:

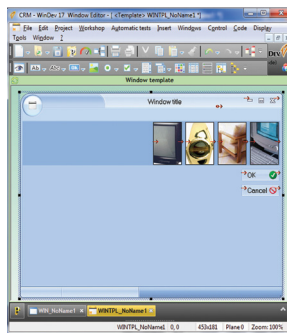
- You can also modify by programming the internal window used in the "Internal window" control.
- Limitations: The home area is a rectangular area and no overload is allowed. To perform overloads, we recommend that you use the control templates.



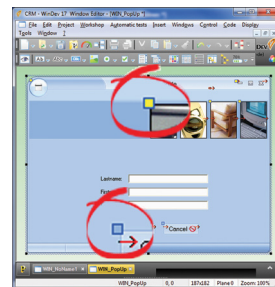
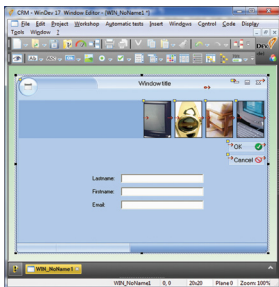
Window templates

WinDev enables you to create window templates. These templates contain all the graphic elements common to all the windows of your application. The modifications performed in a window template are automatically applied to all the windows that use this template.

A window template makes it easier to comply with the style book of an application.



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



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

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

The characteristics of the elements can be dissociated from the template. For example, the position of a control can be dissociated from the template so that the control can be located somewhere else while it still benefits from the other features (code, style, ...). We talk of **control inheritance**. In this case, the elements are identified by a blue square.

The windows in practice

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

WinDev proposes several solutions to create the windows of a project:

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

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

See "The controls in practice", page 39 for more details about handling the window controls.

This chapter presents the following topics:

- "Creating a window", page 21.
- "Simple operations on a window from the editor", page 22.
- "Main characteristics of a window", page 22.
- "Window and image", page 24.
- "Opening a "popup" window", page 26.
- "Dim disabled windows", page 27.
- "Internal window", page 27.
- "Window templates", page 27.
- "Processes associated with the windows", page 28.
- "The menus", page 29.
- "The status bar", page 32.

1 Creating a window

WinDev enables you to create several types of windows:

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

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

1.1 Creating blank windows

To create a blank window:

1. Select "File .. New .. Window".
2. Choose the "Blank" type.
3. Specify the skin template of the window.
4. Validate the creation of the window. A blank window is created.
5. Save the window ("File .. Save").

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

7. Create the controls in the window.

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

1.2 Internal window

An internal window is a window with no title bar and no status bar ... that will be directly included in the other windows of your application via the "Internal window" control.

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

To create an internal window:

1. Select "File .. New .. Window".
2. Select the "Internal window" tab.
3. Select the requested type of window and the associated skin template.
4. Validate.

1.3 Window based on a template

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

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

To create a window based on a template:

1. Select "File .. New .. Window".
2. Select the "Based on a template" tab.
3. Select the requested template and the associated skin template.
4. Validate.

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

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

2 Simple operations on a window from the editor

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

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

- Modifying the size of a window.
- Saving and copying a window.
- Importing and exporting a window.
- Modifying the tab order.

See the online help for more details.

3 Main characteristics of a window

3.1 Characteristics

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

- be **resizable**: the user can increase or decrease the window's size. Depending on the anchoring characteristics defined for each one of the controls, the position of the controls in the window can change during this resize operation (see "Anchoring the controls", page 44 for more details).
- have a **background image**: this image is displayed in the window and controls are displayed on top of this image. You can also define the characteristics for resizing the windows via the 9-image mode (see the next paragraph). See "Window and image", page 24 for more details.
- be **clipped**: the window will have a custom shape (window in the shape of a circle, computer, etc.). See "Clipped window", page 26 for more details.
- have a **drop-down menu**: this menu will allow users to quickly access the application's main features. To create a drop-down menu in a window, all you have to do is select "Windows .. Main menu .. Add the main menu". The first option is displayed in the window. The options can be handled via the popup menu of options. Then, the

menu options can be handled by programming. For more details, see "The menus", page 29.

- have a **toolbar**: this toolbar will display the help messages associated with the different controls of the window. This status bar is also used to display additional information. See "The status bar", page 32 for more details.
- be **multilingual**: all the languages defined for the window will be available for all the window controls. A window can be associated with more languages than the project (case of windows shared among several projects). See "Multilingual applications", page 139 for more details.
- **turn gray** automatically when it becomes unaccessible. This feature is called DDW, which stands for Dim Disabled Windows. See "Dim disabled windows", page 27 for more details.
- authorize **drag and drop** operations. See the online help for more details.

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

3.2 Description window

The description window is a window containing tabs, used to group all the configurable characteristics of the window.

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

3.3 Displaying the characteristics

To display the description window:

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

3.4 Characteristics by tab

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

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

General tab

The "General" tab is used to specify the name of the window and its title.

GUI tab

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

- visibility of the window,

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

Details tab

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

- animation,
- automatic closing.

Image tab

This tab is used to configure:

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

Languages tab

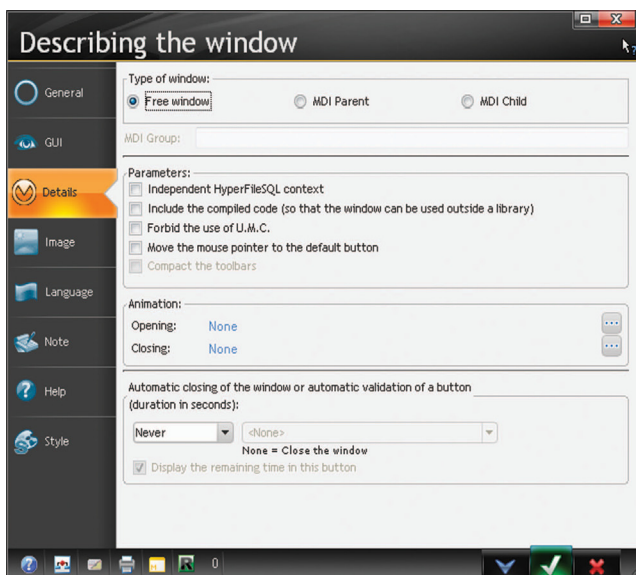
This tab is used to find out the languages supported by the window as well as the language displayed in the editor. You have the ability to add, modify or delete languages.

Notes tab

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

Help tab

The "Help" tab is used 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 is used to define the window's style:

- Presence of the title bar.

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

4 Window and image

4.1 Overview

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

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

Notes:

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

4.2 Configuring the background image

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

You can select the background image with the radio button or select an image from the catalog. The main image formats that can be displayed are: BMP, JPEG, TIFF or GIF.

You can also drag and drop the image file from the explorer toward the WinDev or WinDev Mobile window.

The options available for the image are:

- Management of multilingual images.
- Display Mode.
- Position.
- Transparency.
- Symmetry.
- 9-image mode.
- Window clipping according to the background image.

Management of multilingual images

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

To define a multilingual background image:

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

Display mode

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

- **100%**: the image is displayed at 100% in the top left corner of the image control.
- **Centered**: the image is displayed at 100% and centered in the image control.
- **Stretched**: the image is stretched to occupy the entire surface of the image control.
- **Tiled**: the image is displayed at 100% and tiled to occupy the entire surface of the image control.
- **Homothetic**: the image is proportionally enlarged to be entirely displayed in the image control.
- **Homothetic extended**: the image is proportionally enlarged in order for the smallest side of the image to be entirely displayed in the image control.
- **Homothetic centered**: the image is homothetic and centered in the image control.
- **Homothetic extended centered**: the image is extended and centered in the image control.
- **Homothetic without enlargement**: If the image can be entirely displayed in the window, it is displayed 100%. If the image is larger than the window, it is automatically reduced proportionally in order to be entirely displayed.

- **Homothetic centered without enlargement:** The image is centered and homothetic without increasing size.

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

Position

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

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

Transparency

The transparency options for the background image are as follows:

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

Transparency: none



Transparent magenta



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

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

Symmetry

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

Initial image:



The options are as follows:

Vertical symmetry



Horizontal symmetry



Horizontal and vertical symmetry



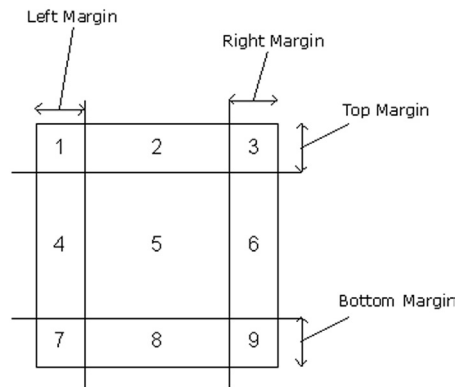
9-image mode

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

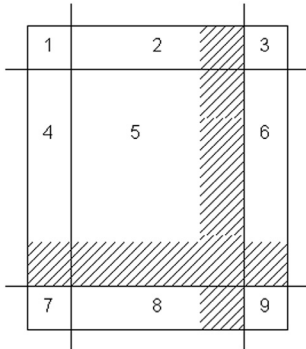
The 9-image mode consists in dividing the background image into 9 areas.

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

The margins allow you to define the size of each area.



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



The 9-image mode allows you to manage the operating mode of each area when the window is resized. For each area:

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

5 Opening a "popup" window

5.1 Overview

WinDev allows you to open "popup" windows. A "popup" window corresponds to a small-size window that opens on top of the current window. In most cases, this window allows the user to perform a selection. This window returns to the main window the value selected by the end user. This window is automatically closed as soon as a left click is performed (inside the popup window or not).

4.3 Clipped window

WinDev enables you to clip the windows. The windows will have custom shapes (window shaped like a circle, a computer, ...).

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

Caution: The borders of this image must be as clear as possible (no gradient, ...) in order to prevent bad display effects when previewing or opening the clipped window.

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

For example:



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

- opened by default below the control that opens the window (or above the control or on the left of the control if there is not enough space below the control). the opening location can be changed by programming.
- automatically closes as soon as the window loses focus or the **Close** function is called.
- only the closing icon appears in the window's title bar (if the title bar was defined in the style).

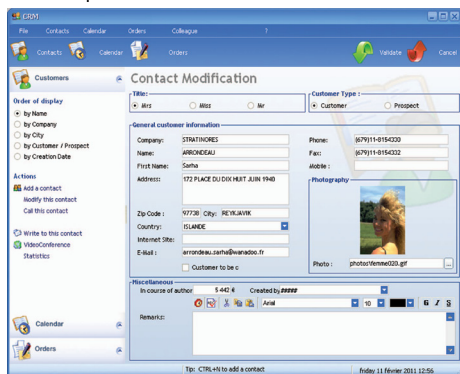
6 Dim disabled windows

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

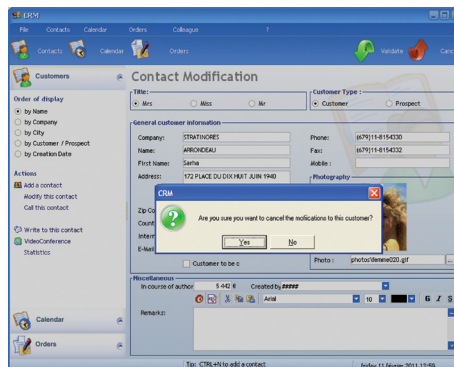
The inactive windows can be automatically dimmed. This feature enables you to always know which window to use.

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

For example:



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



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

The DDW features are available at the project level or they can be customized for each window.

You can also act on the DDW feature by programming.

See the online help for more details.

7 Internal window

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

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

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

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

See the online help for more details.

8 Window templates

WinDev enables you to create window templates. These templates contain the graphic elements and the code common to all the windows of your application.

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

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

Using window templates in your applications enables you to:

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

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

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

by overloading the template elements.
See the online help for more details.

9 Processes associated with the windows

9.1 Processes managed by default

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

- **Global declarations:**
Declaration of the variables global to the window
 - **Initialization:**
Run when the window is opened, before the code for taking focus. The initialization process of the window controls is run after this code.
 - **Closing:**
Run when the window is closed. Note: resuming the input in one of the window controls in this code prevents the window from closing.
 - **Focus Gain:**
Run when the window takes focus (the title bar is active) and another application window loses focus:
 - the user has clicked another window of the application,
 - **ReturnToCapture** was run from another window of the application.
- This process will not be run:
- when opening the first window of the project,
 - when running **OpenChild** or **OpenSister** from the current window,
 - if **Close** is called in the initialization code of the window,
 - if the user clicks a window from another program then re-clicks the initial window.
- **Focus Loss:**
Run when the window loses focus when going to another window. This process is always called before the 'gain of focus' process of another window in the following cases:
 - closing the window (before the closing process),
 - the user has clicked another window of the application,
 - **ReturnToCapture** was run from another window

of the application.

This process will not be run:

- if **ReturnToCapture** for another window has been called in the window's initialization code,
- if **Close** is called in the initialization code of the window,
- if the user clicks a window from another program then re-clicks the initial window.
- **Resizing:**
Run in the following cases:
 - minimizing the window,
 - enlarging or resizing the window,
 - restoring the window once it was minimized.
- **Whenever modified:**
Run when a window control is modified (data input for instance)
- **Assign the ..Value property (internal window only)**
Run when using the ..Value property in assignment on the Internal Window control.
- **Get the ..Value property (internal window only)**
Run when using the ..Value property in read mode on the Internal Window control.

9.2 Optional processes

Several optional processes can be managed.

To manage an optional process, all you have to do is select it in the icon bar of the code window.



You have the ability to manage:

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

10 Popup menus

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

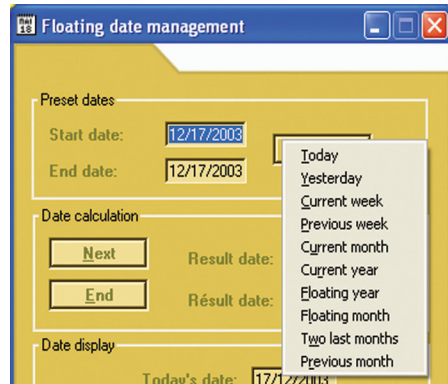
WinDev enables you to manage two types of popup menus:

- **the system popup menus** supported by Windows. These popup menus are displayed when a right click is performed on some types of controls (edit control, table, ...). See "Default system popup menu", page 30 for more details.
- **the system popup menus** supported by WinDev. These menus are created in the window editor and they can be handled by programming.

WinDev enables you to create custom popup menus and to associate these popup menus:

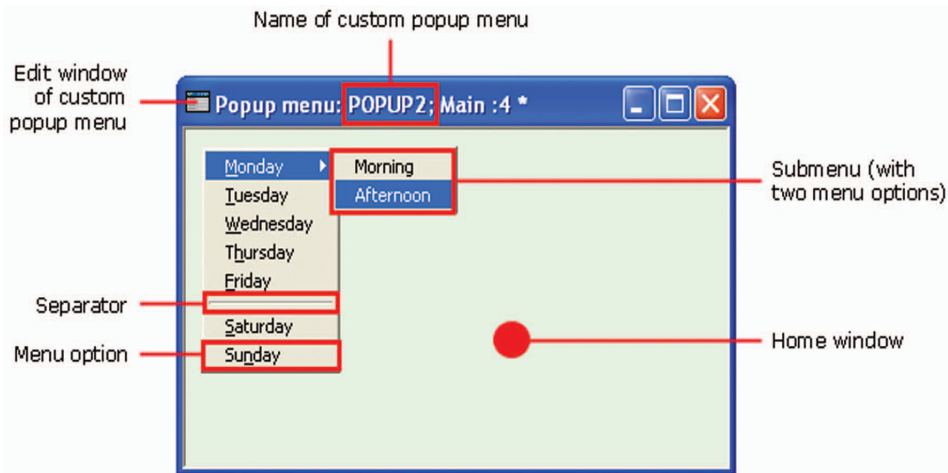
- with one or more controls found in a window.
- with a window.

For example:



10.1 Window for editing the popup menus

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



10.2 Available features

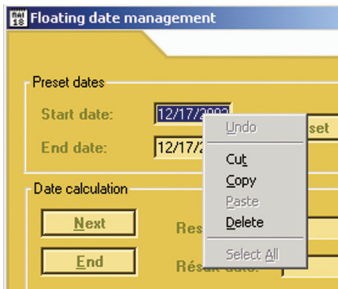
WinDev enables you to:

- Create a custom popup menu.
 - Associate a custom popup menu with one or more elements (window or controls).
 - Edit a custom popup menu.
 - Display the description window of a custom popup menu.
 - Delete a custom popup menu.
- See the online help for more details.

10.3 Default system popup menu

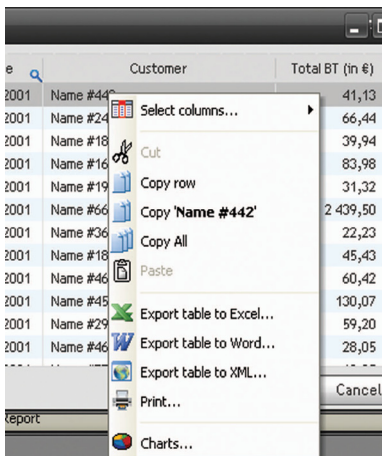
The following types of controls have a default system popup menu (displayed if no custom popup menu is associated with the control):

- an edit control:



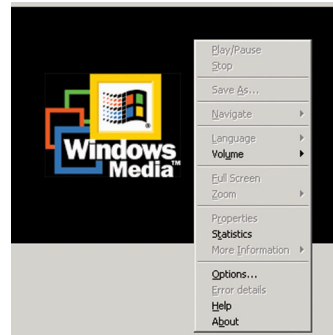
See the online help for more details.

- a table:



See the online help for more details.

- an ActiveX control:



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

- an HTML control:



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

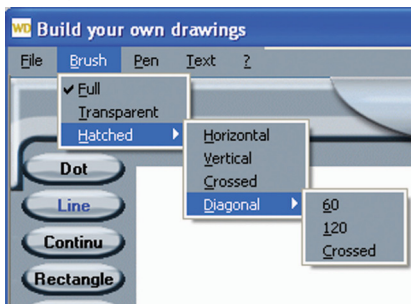
This default popup menu can be disabled from the editor or by programming.

11 The main menus

11.1 Overview

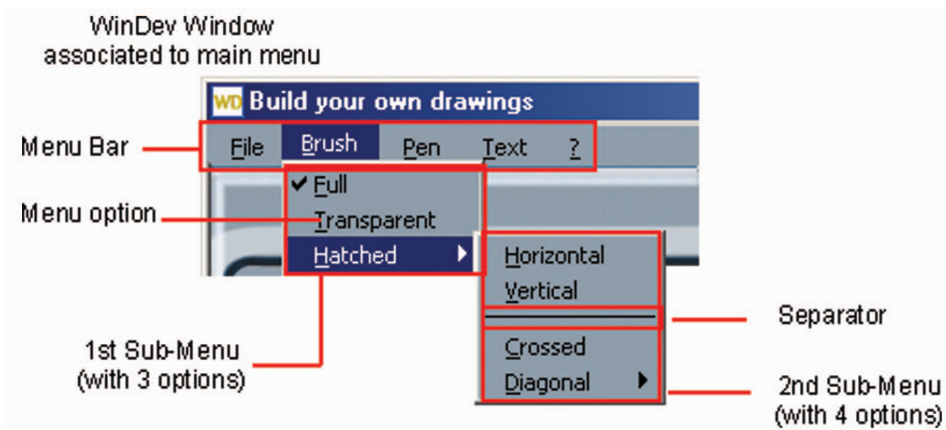
A main menu is necessarily associated with a window. This menu is displayed below the title bar of the window in WinDev. In most cases, the applications contain a menu window. This window only contains a main menu and it is used to access the other windows of the application.

For example:



11.2 Vocabulary associated with main menus

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



11.3 Available features

WinDev enables you to:

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

See the online help for more details.

Special case: Tool menu

WinDev allows you to automatically insert a "tool" menu ("?") into your application ("Windows .. Main menu .. Add the ? menu").

12 Menu options

12.1 Overview

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

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

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

To edit (or display in the editor):

- the main menu of a window:
 - open the relevant window in WinDev.
 - select one of the menu options. The options and the sub-options are automatically displayed.
- a custom popup menu, select "Windows ..Popup menu .. <PopupMenuName>".

12.2 Handling the menu options in the editor

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

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

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

The description window of the menu options allows you to:

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

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

See the online help for more details.

12.3 Handle the menu options by programming

The method for handling a menu option is identical whether the menu is a popup menu or a main menu.

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

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

13 The status bar

In the status bar of a window, you can display:

- a **information message linked to the current control**. This message is entered in the "Help" tab of the description window of the control. The message will be displayed in the status bar when the control is selected by the user.

- a **specific information message**. This message is displayed by programming by the **Message** function.
- a **progress bar**. This progress bar is displayed by programming with **Gauge**

To display various information at the same time, you have the ability to divide the status bar into

several boxes ("Insert a cell" from the popup menu of the status bar).

13.1 Description window of a status bar

The description window of a status bar is used to:

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

See the online help for more details.

13.2 Type of cell in the status bar

You have the ability to display:

- The progress bar displayed by the **Gauge** function.
- The help message associated with the control that has focus (help message defined in the control "Help" tab or by programming with the **..Message** property).
- The date of the current computer.
- The time of the current computer.
- The position of the cursor in the current edit control (Column and Row of the Caret).
- The status of the [Caps Lock] key, indicating whether the Caps Lock key is enabled or not.
- The status of the [Insert] key, indicating whether the [Insert] key is enabled or not.

- The name of the current user in an application that uses the user groupware.

Two specific options are also available:

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

13.3 Characteristics of a cell in the status bar

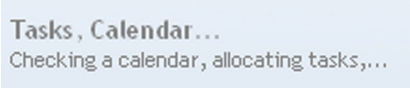





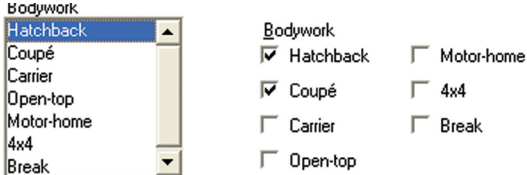


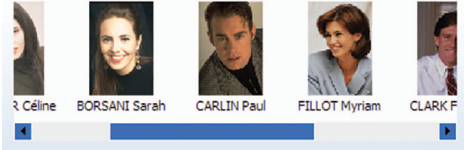






The description window of the status bar also enables you to define the display characteristics of the cells found in the status bar.





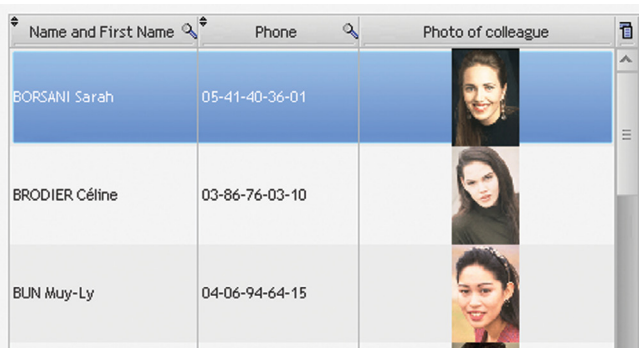


- The position of the cell in the status bar.
- The background image of the cell.
- The background color of the cell.
- The width of the cell.
- Whether the cell is editable.
- The cell anchor (how it behaves when the window is resized).








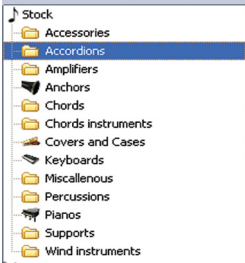


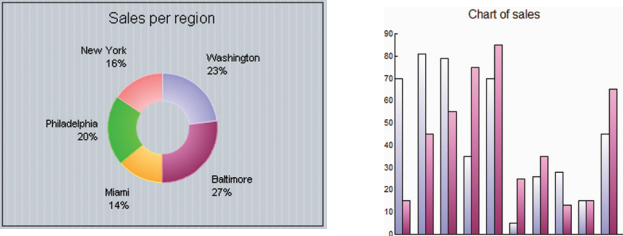

All these characteristics can be handled by programming via the properties available for the cells in the status bars.



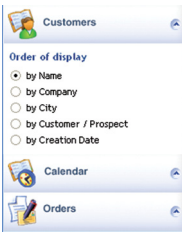







The different types of standard controls

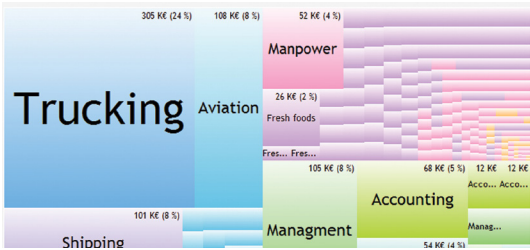

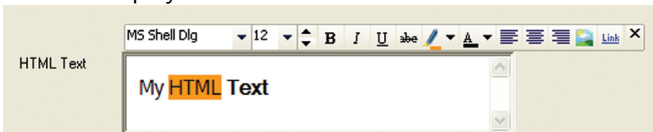




WinDev includes several controls that are supplied with the product. Here is how to use them based on the operation to perform:

You want to...	Use a control...
Display a text, a title, ... 	 Static
Select a value from a list (Country, City, ...) 	 Radio button  Combo box  List box
Select several values from a list (recipients of a message, ...) 	 Check box  List box
Select one or more values from a listview (picture directory, ...) 	 ListView
Select a value within a preset range 	 Slider  Round slider  Spin  Range Slider

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

<p>Display a video, an animation</p> 	 Web Camera  Multimedia  Conference
<p>Enter information</p> <p>Email <input type="text" value="freetechnicalsupport@windev.com"/></p>	 Edit control
<p>Program an action in a window (display another window, start a print, ...)</p> 	 Button
<p>Display information using a hierarchical system (directory)</p> 	 TreeView  TreeViewTable
<p>Display a bar chart, a line chart, a pie chart, ...</p> 	 Chart

<p>Display a progress</p> 	 <p>70% Progress bar</p>
<p>Group the controls by theme and display the themes one by one</p> 	 <p>Tab</p>  <p>Sidebar</p>  <p>HideShow</p>
<p>Display a bar code</p> 	 <p>Bar code</p>
<p>Display events over a period of time</p> 	 <p>Organizer</p>

<p>Represent weighted data</p> 	 <p>TreeMap</p>
<p>Enter or display HTML data</p> 	 <p>HTML edit control</p>  <p>HTML control</p>
<p>Allow the selection of interval or zoom</p>	 <p>Range slider</p>
<p>Display an ActiveX control</p>	 <p>ActiveX</p>

The controls in practice

WinDev proposes several types of controls that can easily be included in your windows. These controls can be created via the "Control" toolbar or via the "Insert" menu of the window editor.

All the controls can be handled by programming.

1 Creating a control

1.1 The different types of window controls

The following controls are available in the WinDev window editor:

- Edit control,
- Static control,
- Button,
- Image,
- Radio button,
- Check box,
- List box,
- Combo box,
- ListView,
- Sidebar,
- HideShow,
- Web camera,
- Conference control,
- OLE control,
- ActiveX,
- Shape,
- Bar code,
- HTML control,
- Supercontrol,
- Organizer,
- Scheduler,
- Rating,
- Table,
- Treeview table,
- TreeView,
- Organization Chart,
- Progress Bar
- Looper,
- Scrollbar,
- Slider,
- Spin,
- Tab,
- Ribbon,
- Internal window control,
- Toolbar,
- Splitter,
- Chart,
- Carousel,
- Cube,
- Xaml,
- Calendar,
- Internal window,
- Control template.

All the controls can be handled by programming. See the online help for more details.

1.2 Creating a new control

To create a control:

1. Select the type of control to create:
 - via "Insert .. Control/Shape/Special".
 - via the corresponding icon in the "Controls" toolbar.
2. The shape of the new control appears below the mouse cursor.
3. Click the position where the control will be created in the window. The control is automatically created.

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

1.3 Creating a control associated with an item

All the controls in a window can be associated with a data file item, EXCEPT:

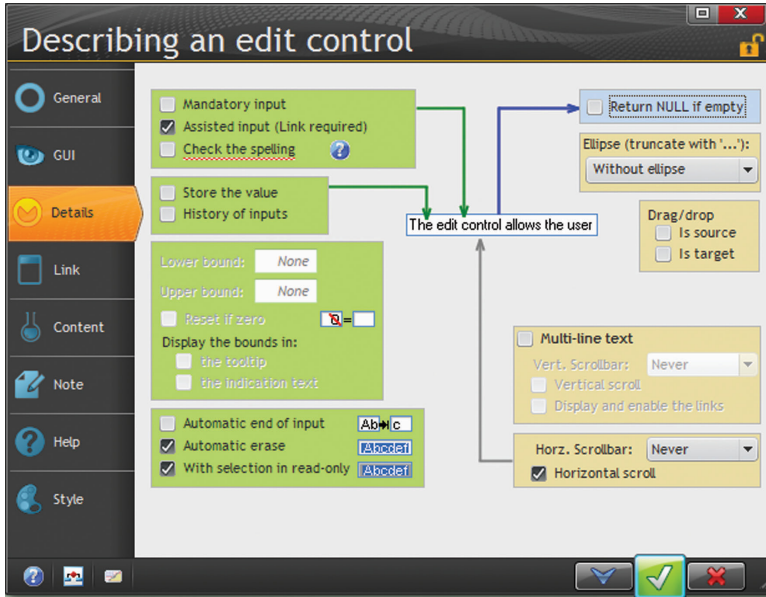
- button
- shape
- tab
- toolbar
- Xaml
- Carousel
- Cube
- HTML control

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

- Drag and Drop from the "Analysis" pane.
- Using the "File" tab of the control.

2 Characteristics of a control

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



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

2.1 Displaying the characteristics

To display the description window of a control:

- double-click the control.
- select "Description" from the popup menu of the control (right mouse click).
- select the control and select "Display .. Description of the selection".
- select the control and press [Alt]+[Enter].
- select "Display .. Options .. Modify the options" ([F12]), select the control and click the "Edit" button.

Notes:

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

2.2 Characteristics by tab

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

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

General tab

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

GUI tab

The "GUI" tab is used to define the settings for the control's interface:

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

Details tab

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

- input parameters
- Drag and Drop, ...

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

Link tab

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

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

Content tab

This tab is available for the window controls only.

The "Content" tab is used to define:

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

Notes tab

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

Help tab

The "Help" tab is used to enter and configure all the types of help systems associated with the control. A control can have:

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

A control can also be associated with a specific help page, found in a help page in CHM format. 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. The function named *WHelp* will be used to display the help page.

Style tab

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

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

3 Handling the controls in the editor

3.1 The handles of controls

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

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

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

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

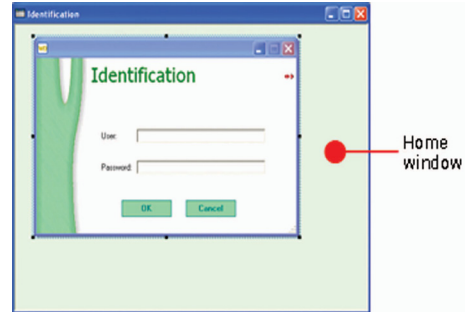
- The **gray handles** show the first control selected during a multiple control selection. This control will be the reference control.

3.2 Available operations

The window editor is used to:

- Select one or more controls (via the lasso for example)
- Group the selected controls
- Move a control
- Modify the caption of a control
- Display an advanced tooltip when a control is hovered by the mouse cursor, containing: the name of the control, its position, its size, its anchoring mode, its initial status.

- Use an off-window control. This control will be visible in edit and will not be visible at run time.



Note: An "off-window" control allows you to, for instance:

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

4 Alignment of controls

Control alignment is used to create "professional" and outstanding interfaces. To help you build standard interfaces, WinDev offers the following tools:

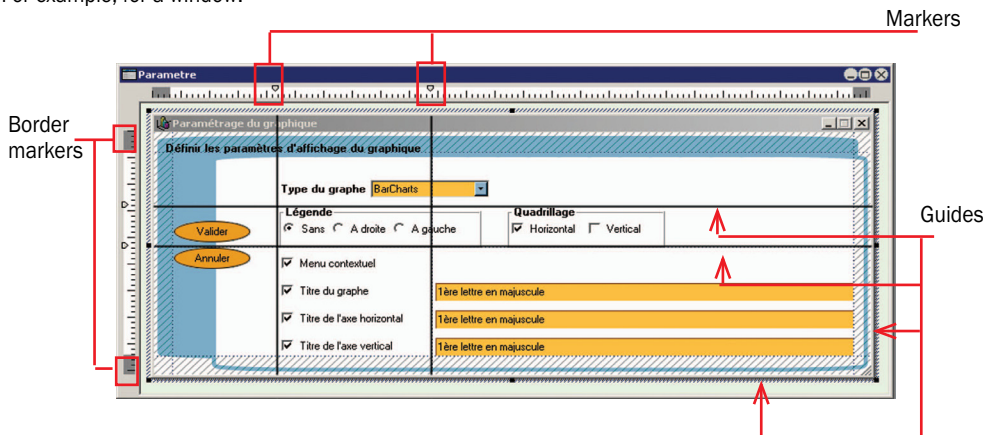
- the rulers.
- the grid.
- the real-time interface checker (automatically proposed when positioning a control).
- the advanced interface checker.
- the alignment options.
- the configurable zoom.

4.1 The rulers

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

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

For example, for a window:



Displaying the rulers

To display the rulers, select "Display .. Rulers". Two types of guides can be used:

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

Note: To turn the guides on or off for a window, select "Guide visible" from the popup menu of the rulers. Whether they are visible or not, the guides are always enabled: any control moved toward a marker is automatically snapped by it.

Handling the markers

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

4.2 Snap-on grid

The snap-on grid lets you place vertical and horizontal markers in a 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 inside a window, guides are displayed automatically 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 that enables you to harmonize the positioning of the controls in the various application windows.

The advanced interface checker offers for the current window or only for the selected controls to apply interface presentation rules from the Windows standard: control alignments, standardization of the button size, ...

4.5 The alignment options

To align several controls, WinDev proposes:

- a preset alignment (available via "Control .. Alignment.." or in the toolbar for control alignment).
- a custom alignment ("Control .. Alignment .. Custom alignment"). The custom alignment allows you to use specific alignment properties.

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

- **the horizontal or vertical alignment:** the controls can be aligned on the left according to the beginning of the control or according to the beginning of the input area. They can also be aligned to the right according to the end of the control or according to the end of the input area.
- **the regular spacing:** the spacing between the controls is adapted in order to be identical between each control (horizontally and vertically).
- **the size:** the size of the selected controls is automatically adapted in order for the controls to have the same height and/or the same width.

- **the reference control:** for the alignment and size of the controls, you can choose the reference control to be:
- **the first selected control** (the one with the white handles)
- **the last selected control**
- **the largest selected control**
- **the control located in the top left position of the selected controls**

5 Edit options of the controls in a window

The edit options ("Display .. Options .. Modify the options") are used to configure:

- the display options.
- the magnetism of controls.
- the click and double-click operations on the controls.
- the options for control selection.
- the position of the window when it is opened.

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

The information that must be displayed for each window in the window editor can also be selected via "Display .. Options..".

5.1 Making a control visible/invisible

The "Control .. Edit options .. Visible" option is used to make a control visible/invisible in the window editor.

5.2 Locking/unlocking a control

The "Control .. Edit options .. Locked" option is used to lock/unlock a control.

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

All other operations are possible on a locked control.

5.3 Framing (or not) a control

The "Control .. Edit options .. Framed" option is used to frame (or not) a control.

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

6 Anchoring the controls

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

The management of control anchoring includes two parameters:

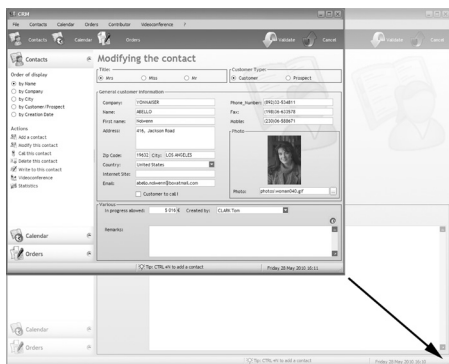
- **the control anchoring:** this parameter is used to define the modification that must be applied to the control according to the change of window size. The control can be moved to the left or to the bottom, its size can be increased in width

and/or in height, ...

- **the anchor rate:** parameter used to specify the percentage by which a control is moved or stretched. This parameter is used to:
 - keep a control centered in width in the window regardless of its size.
 - proportionally increase the size of some window controls.

These parameters can be configured:

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



The size of anchored controls is automatically modified.

6.1 Anchoring a control in the editor

To anchor a control:

1. Select one or more controls to anchor.
2. Define the anchoring of the control(s):
 - in the "GUI" tab of the description window of the control ("Display .. Description of the selection").
 - via "Anchor" from the popup menu (right mouse click) of the control.
3. In this window, choose the type of anchoring you want to use. The most common options are represented by icons:



No anchor: The control is not modified when the size of the window is modified. No anchoring option is selected.



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



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



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



Height: The control stretches toward the bottom when the height of the window increases. To change the speed at which a control stretches compared to the window, use the anchor rate in height.



Width and Height: The control stretches toward the right and toward the bottom when the window gets bigger. To change the speed at which a control stretches compared to the window, use the anchor rate in height and the anchor rate in width.



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



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



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



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





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





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

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

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

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

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

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

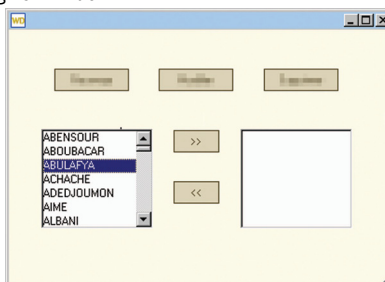
4. Define (if necessary) the different anchor rates.
5. Validate. If "Display .. Options .. Show the anchors" is enabled, the anchoring signs are automatically displayed in the control (red arrows).

Tip: If WinDev detects an anchoring problem in a window, a tip is automatically displayed in the development environment. Accepting this tip allows you to automatically optimize the interface of your resizable windows.

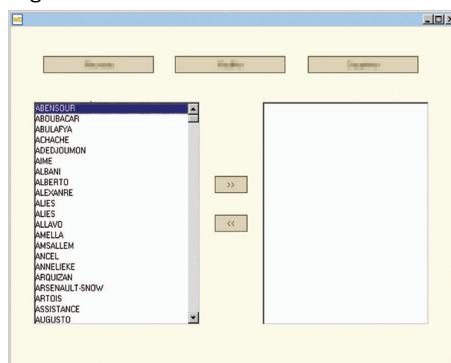
6.2 Example for configuring the control anchors

Let's see a detailed example, illustrating the anchoring possibilities for the window controls.



Original window:



Larger window:



To obtain this result:

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

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

7 Automatic execution of a button

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

7.1 Implementation

To automatically run a window button:

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

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

7.2 Programming

You can also define a button for automatic validation with *DelayBeforeClosing*

7.3 AAF: Automatic Application Feature

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

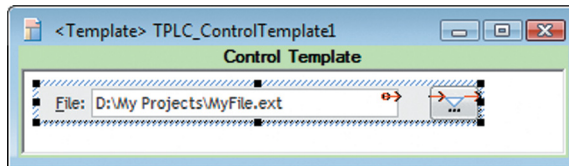
Control templates

WinDev enables you to create control templates. A control template is a set of controls that can be re-used in several windows.

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

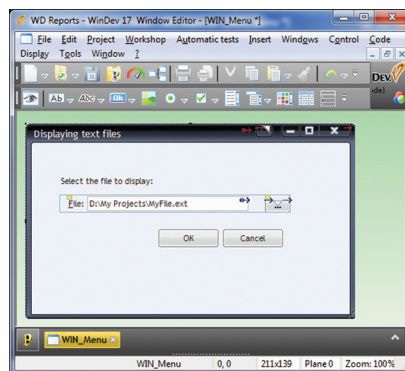
A control template is used to:

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



Defining a control template.

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



Using the template in a window.

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

To create a window based on a control template, create a "Control Template" control.

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

The characteristics of the elements can be dissociated from the template. For example, the position of a control can be dissociated from the template so that the control can be located somewhere else while it still benefits from the other features (code, style, ...). We talk of **control inheritance**. In this case, the elements are identified by a blue square.

Edit: The reports

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

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

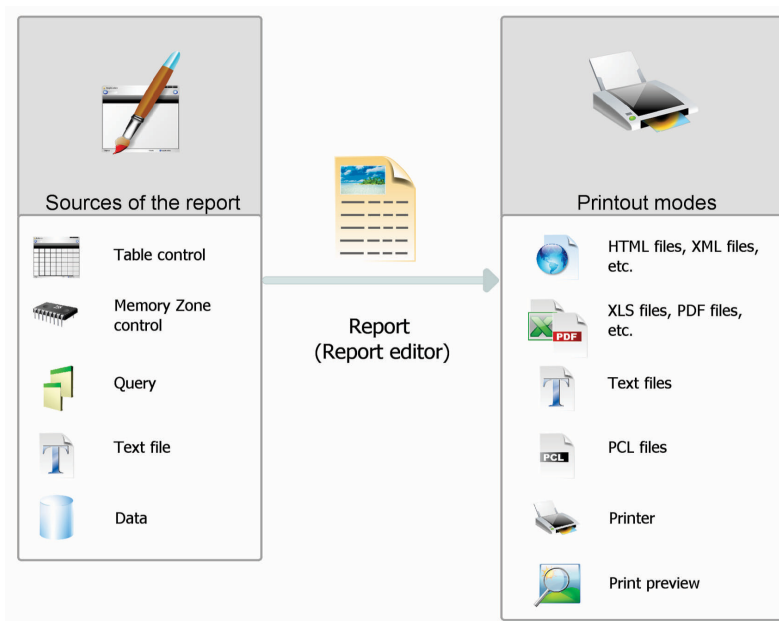
A report can be used to summarize and synthesize the data.

You can:

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

The diagram below presents a simplified definition of a report:

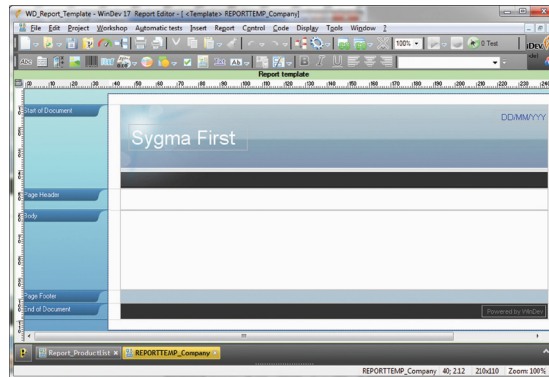
- the data to print comes from a data source (file described in an analysis, HyperFileSQL view, query, memory zone or text file).
- the report groups, sorts and formats the data.
- the report can be directed to the screen, an HTML file, a RTF file, ... or to a printer.



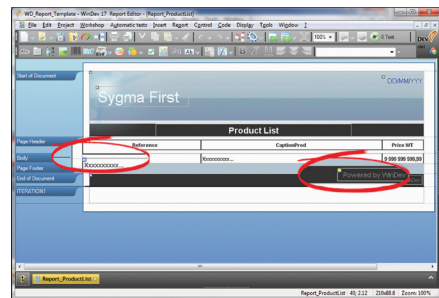
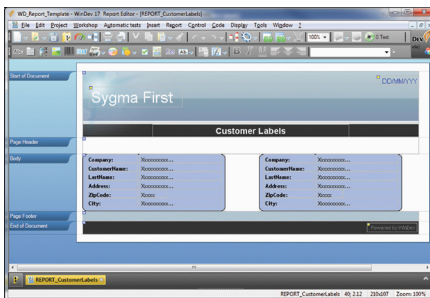
Report templates

In a company, most of the time, printouts follow a standard layout and style: date in the top right corner in a specific format, footer including time printed and file name, logo on the top left, ...

Report templates are used to easily standardize the layout of your reports.



Defining a report template in the report editor.



Using the template in different reports.

The elements belonging to the template are identified by a yellow square.

The overloaded template elements are identified by a blue square.

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

The characteristics of the elements can be dissociated from the template. For example, the position of a control can be dissociated from the template so that the control can be located somewhere else while it still benefits from the other features (code, style, ...). We talk of **inheritance**. In this case, the elements are identified by a blue square.



PAR 2

Development environment



Express



DEVELOP 10 TIMES FASTER

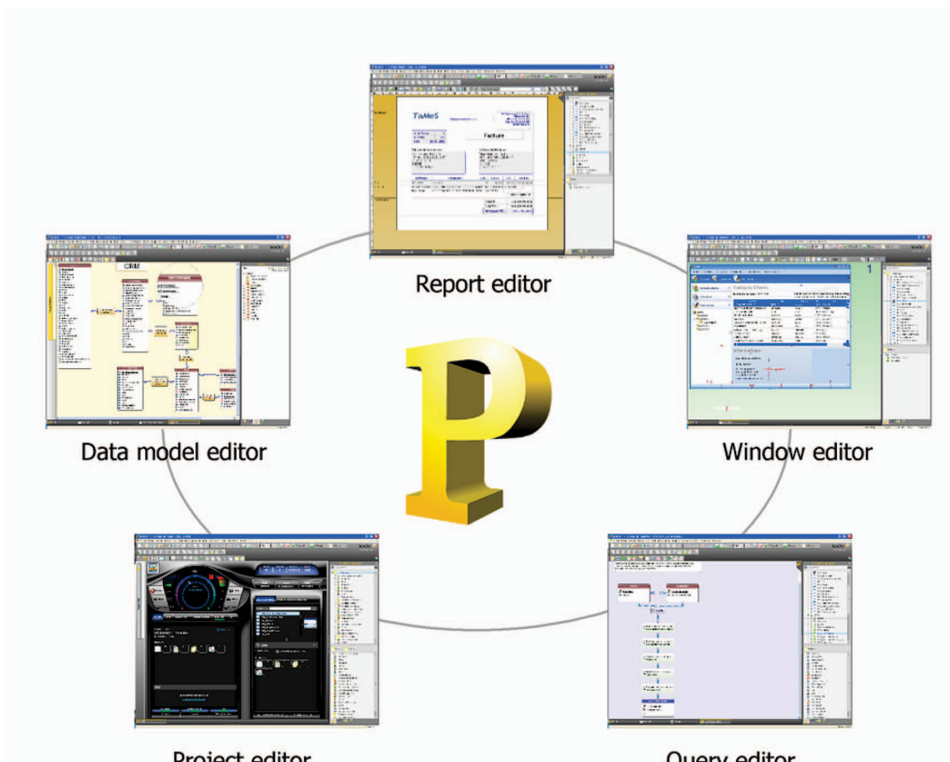
PCSOFT

The WinDev editors

To handle a project, WinDev proposes several editors:

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

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



User-friendly editors in practice

1 Introduction

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

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

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

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

2 The different editors

2.1 Project editor

The project editor includes two main features:

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

Let's see some characteristics of the project editor:

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

2.2 Data model editor

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

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

The data model editor allows two types of database descriptions:

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

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

- WYSIWYG editor ("What You See Is What You Get"). Ability to directly view all the files and links of your project.
- Creation and description of the files and 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, ...).
- Description independent of the data files.
- Encryption of the data files.
- Automatic modification of the data files when the structure of the files is modified.
- Management of the referential integrity.
- Automatic generation of the analysis.
- Zoom in the data model editor.
- Insertion of comments into the analysis graph.
- Interaction between the different panes and the analysis elements.

2.3 UML editor

The UML language is a graphical language allowing you to:

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

The UML editor is mainly used to:

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

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

Let's see some characteristics of the UML editor:

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

2.4 Query editor

The query editor is used to automatically create queries based on the data files. No programming hassle: windows, tables, combo boxes, reports, ... can be based on queries.

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

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

Let's see some characteristics of the query editor:

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

2.5 Window editor

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

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

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

Let's see some characteristics of the window editor:

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

2.6 Report editor

The report editor is used to easily create printed reports.

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

Let's see some characteristics of the report editor:

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

2.7 Code editor

The code editor is used to enter all the processes in WLanguage (the programming language included in WinDev). It is used to enter the source code:

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

Let's see some characteristics of the code editor:

- Automatic formatting of the information entered.
- Automatic completion.
- Glossary of functions.
- Immediate detection of typos and help for correction.
- Incremental compilation.
- Preview of the different processes of a window, control or report found in the project.
- Insertion of processes specific to the use of the mouse or keyboard.
- Management of the history of code modifications with ability to rollback.
- Ability to open several code windows (to perform code comparisons for example).

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

We recommend you read the "WLanguage" guide for more details.

2.8 Help editor

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

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

Let's see some characteristics of the help editor:

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

Project dashboard

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

The dashboard proposes two modes for viewing the project:

- The Developer mode.
- The Project Manager mode.

In Developer mode, the dashboard includes:

- buttons that light up progressively: if the buttons do not light up, then everything is OK. The red buttons signal a potential problem.
- lists of elements, giving quick access to the main options of the project.
- counters, used to manage the new features, the requests, ...



In Project Manager mode, the dashboard is used to get graphic information about the progress status of the project and about the quality of the project. This mode is recommended when using the Project Monitoring with a management of requirements.

WinDev/WebDev/WinDev Mobile: 100% compatible

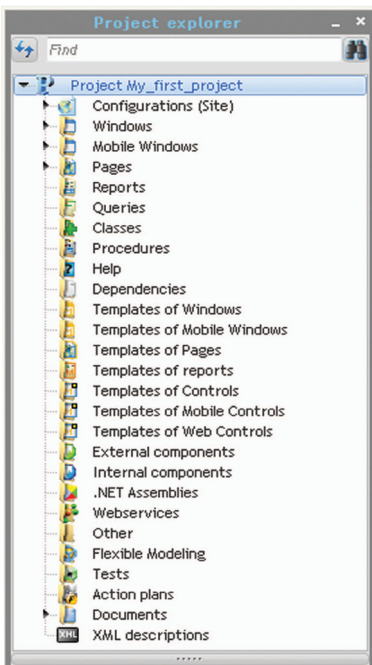
The created projects are often multi-target projects.

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

All the elements, except for the UI (pages and windows), are 100% compatible and sharable among the WinDev, WebDev and WinDev Mobile projects.

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

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



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

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

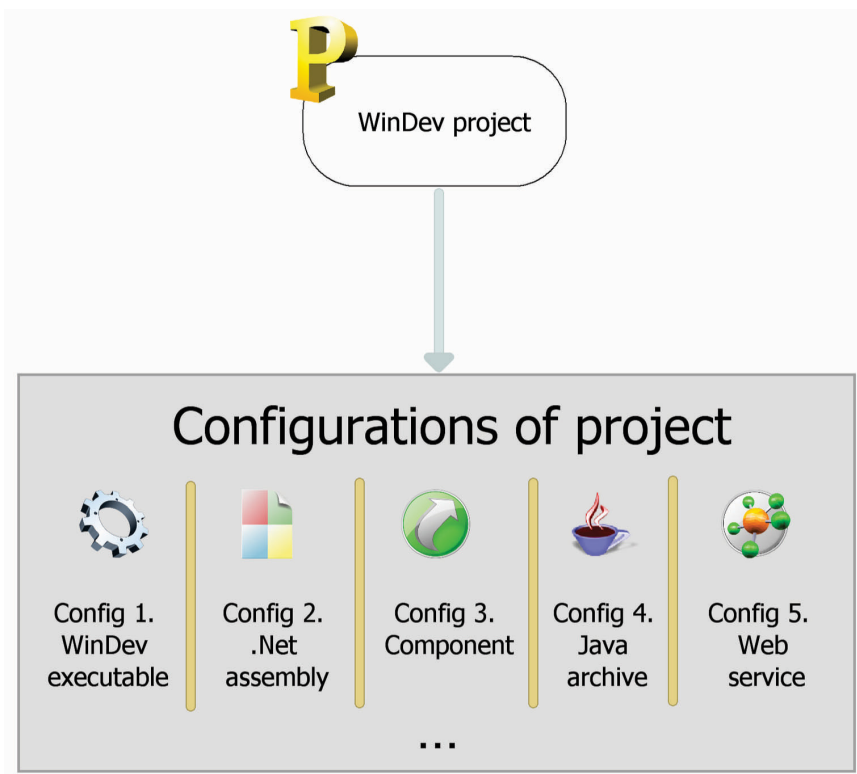
You now have the ability to view the elements of each target from each environment. A project in WinDev displays the thumbnails of the WebDev pages and the WinDev Mobile windows for example. Clicking a WebDev page from the project editor of WinDev opens the WebDev page (WebDev must be installed on the computer).

Project configuration

The project configurations are used to create several "targets" from the same project. From the same project, you have the ability to create:

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

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



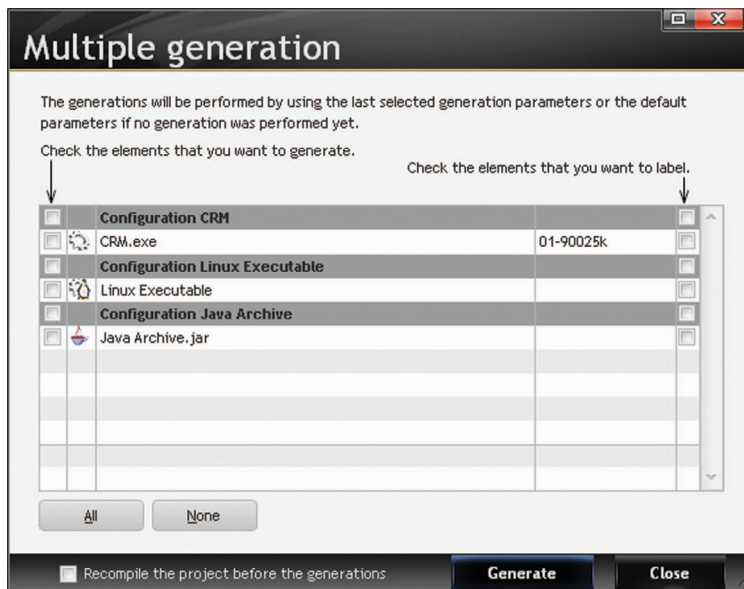
The multiple generation allows you to generate all the project configurations (or some of them) in a single operation.

Multiple generation

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

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

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



To start a multiple generation, select "Workshop .. Multiple generation".

Custom-folders: Organize your project

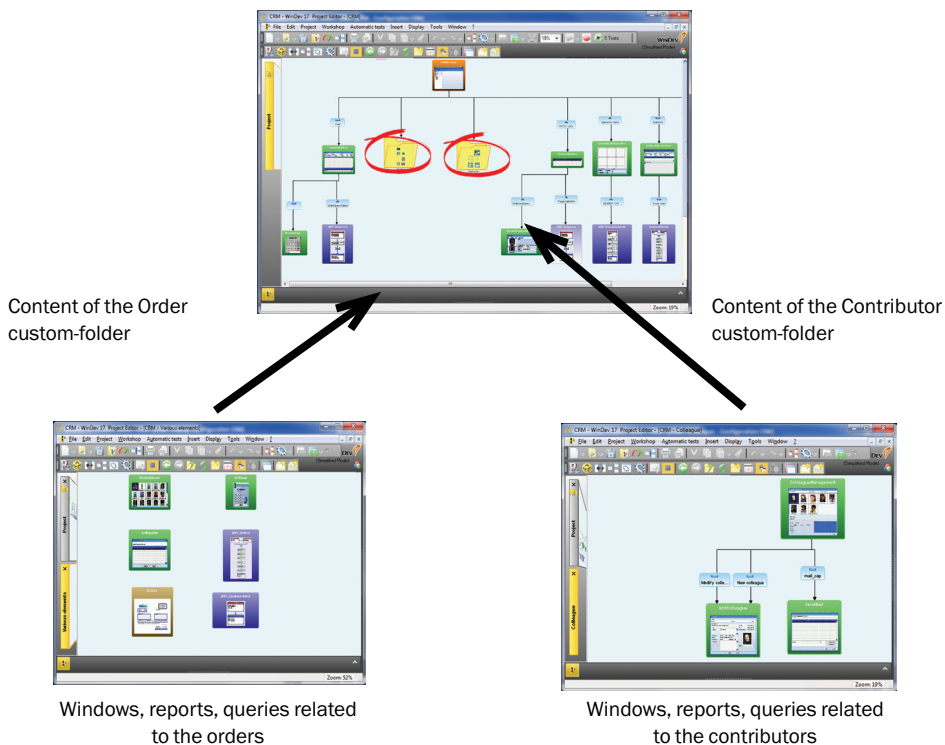
All the elements included in a project are listed in the Project Explorer. By default, the Explorer organizes the elements according to their type: windows, reports, classes...

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

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

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

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



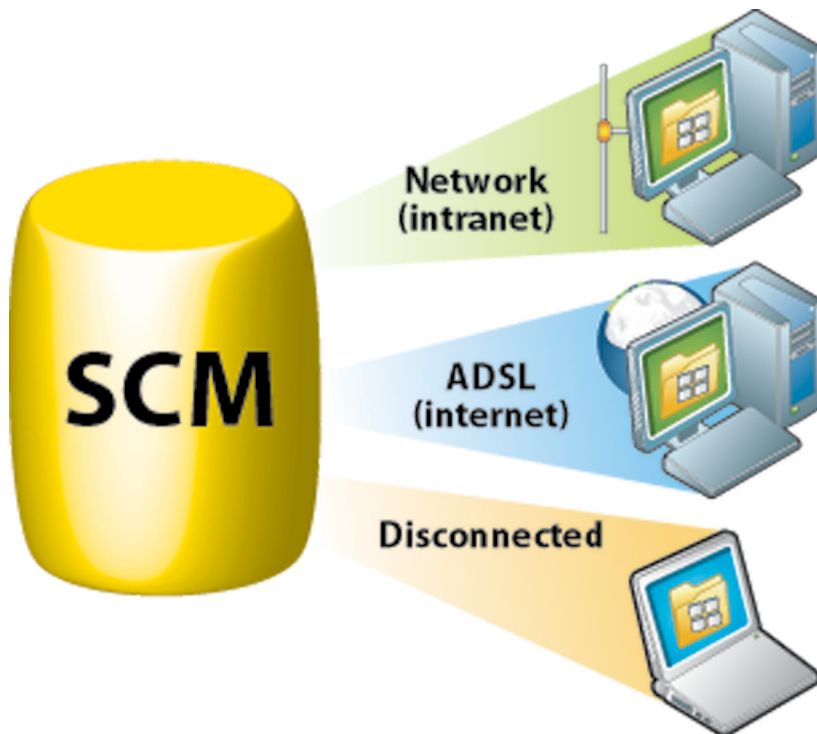
Notes:

- To organize your projects and to share a set of elements among different projects, WinDev also proposes the internal components.
- The "custom-folders" are displayed in the project graph and in the "Project explorer" pane.

Source Code Manager (SCM)

Overview

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

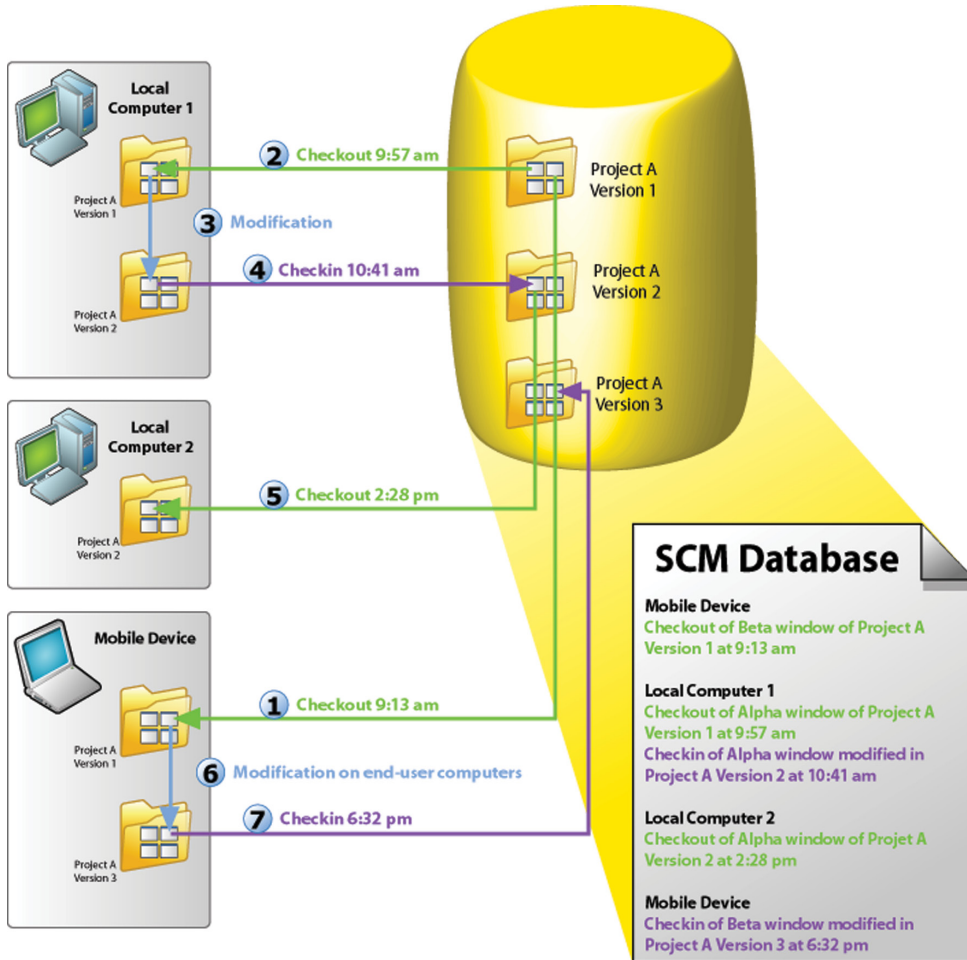


The elements found in the SCM can be shared:

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

Operating mode of the Source Code Manager

The following example presents the Source Code Manager:



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

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

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

Sharing a project in practice

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

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

- the source code manager.

- the developer groupware. It is kept for backward compatibility with the earlier versions. It will not be presented in details in this chapter. See the online help (keyword: "Developer groupware") for more details.

1 The Source Code Manager

1.1 Overview

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

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

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

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

The **SCM uses a source database**: procedures, classes, windows, reports, components, analyses, ...

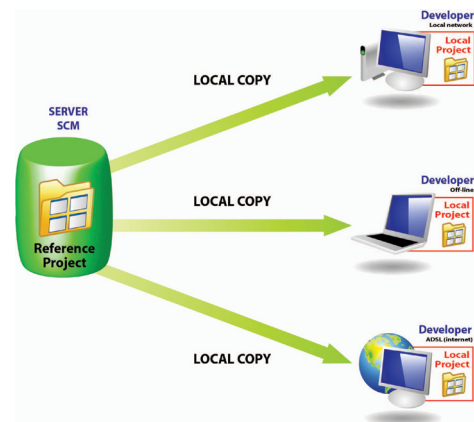
This database can be installed on a server (in HyperFileSQL Classic or HyperFileSQL Client/Server mode) or on a network computer in a shared directory.

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

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

1.2 Principle

Setup



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

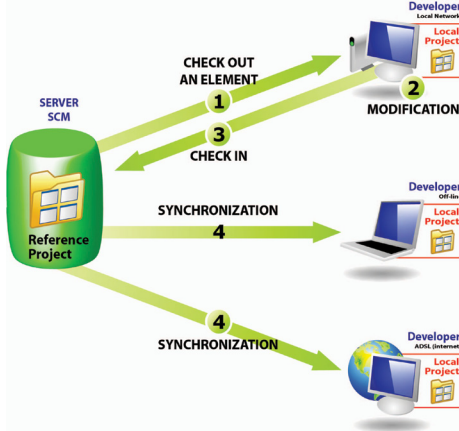
Each developer who uses the Source Code Manager gets a local copy of the project.

Use

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

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

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



Tips

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

- Use a dedicated server with a comfortable size disk (200 GB at least).
- Use the Source Code Manager (SCM) in Client/Server mode.
- The hard disks may encounter physical problems: if possible, use a RAID 1 system on your server (several disks storing the same information).
- Use a UPS to protect the power supply of your server.
- Perform regular backups of the SCM database (at least once a week).
- Place the server in a "secured" area and use a firewall.

2 Creating a database for SCM

2.1 Overview

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

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

The SCM database can be created:

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

2.2 When should a SCM database be created?

The SCM database must be created once only.

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

- when installing WinDev.
- when creating a project that uses SCM.
- when importing a project into the source code manager.
- in the SCM administrator directly.

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

2.3 Backups

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

3 Configuring the project in order to work with SCM

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

3.1 Adding a project into SCM

To add a local project into the SCM:

1. Open your project in WinDev.
2. In the project editor, select "SCM .. Add the entire project".
3. In the wizard, select the SCM database to use.

Note: If no SCM database was created yet, click the "Create a database" button. Specify whether you want to use:

- a source database found on a network share (SCM database in HyperFileSQL Classic format). Specify the directory of the SCM database (network computer or shared directory).
- a SCM database found on a HyperFileSQL Client/Server server. The SCM database will be in Hy-

perFileSQL Client/Server format. In this case, you must specify:

- the server
 - the database
 - the port used
 - The name and password of an administrator of the HyperFileSQL database.
4. Then, click the "Create" button.

3.2 Sharing resources

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

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

4 Working with SCM

4.1 Project options affecting SCM

Several options are used to configure a project handled by the Source Code Manager. These options are grouped in the "SCM" tab of the project description ("Project .. Project 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 found in the SCM database, this option proposes to retrieve the latest version of the project elements. By default, the latest version of the elements is automatically retrieved.

- **Propose to check in the elements when the project is closed:**

When the project is closed, this option is used to display the list of all the elements currently checked out so that they can be checked back in.

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

- **Propose to check in and to get the latest version of the elements when generating the executables, libraries, components, ...**

This option is used to display, when creating an executable, a library ... the list of checked-out elements in order to check them back in and to get the latest version of these elements. Therefore the executable, the component or the library can contain the most up-to-date elements.

By default, the generation of the executable, library, ... is performed with the project elements currently found on the local computer.

- **Check out/Check in the project automatically:**

This option allows for an automatic management of the "project file". If this option is checked, the project file is checked out only if the action performed requires it. Once the action has been performed on the project, the project file is automatically checked back in.

This option allows you to not have the "Master/Guest" management on the project. This option can also be enabled via "SCM .. Project management .. Manage the project checkout automatically".

4.2 Checking out an element

The different check-out modes

SCM proposes two modes for checking out the project elements:

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

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

To change the check-out mode, select "Tools .. Options .. General options of WinDev", then select the "General" tab. The "Checking out elements during the first modification" option is used to switch all the projects opened from the current product to the automatic mode.

Opening a project element to modify its characteristics

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

1. Check out the element from the Source Code Manager.
2. Select the check-out mode of the element. The check-out mode can be:
 - **exclusive:** nobody else will be able to check out this element until it is checked back in. The element can be checked out for test only.
 - **for test:** the element can be modified but the modifications will not be checked back in.
 - **multiple:** the element can also be checked out by other users. In this case, the differences between the different versions of the element can be displayed 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 in an element

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

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

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

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

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

4.4 Management modes of the project

Two management modes of the project are available with SCM:

- Management in Master Guest mode.
- Management in automatic mode (default).

Master and guest

The Source Code Manager distinguishes between 2 types of users:

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

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

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

To switch from master mode to guest mode, select "SCM .. Project management .. Become guest on the project (and check all in)".

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

To switch from guest mode to master mode, select "SCM .. Project manager .. Become master on the project".

Caution: Modification of the project options:

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

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

Automatic mode

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

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

5 Working in offline mode with the SCM

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

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

The principle is straightforward:

- before you disconnect, select "SCM .. Remote work .. Disconnect for a mobile use". Before the

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

- when you reconnect, select "SCM .. Remote work .. Reconnect and synchronize". Then, all you have to do is check in the modified elements.

See the online help for more details.

6 SCM administrator

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

It allows you to:

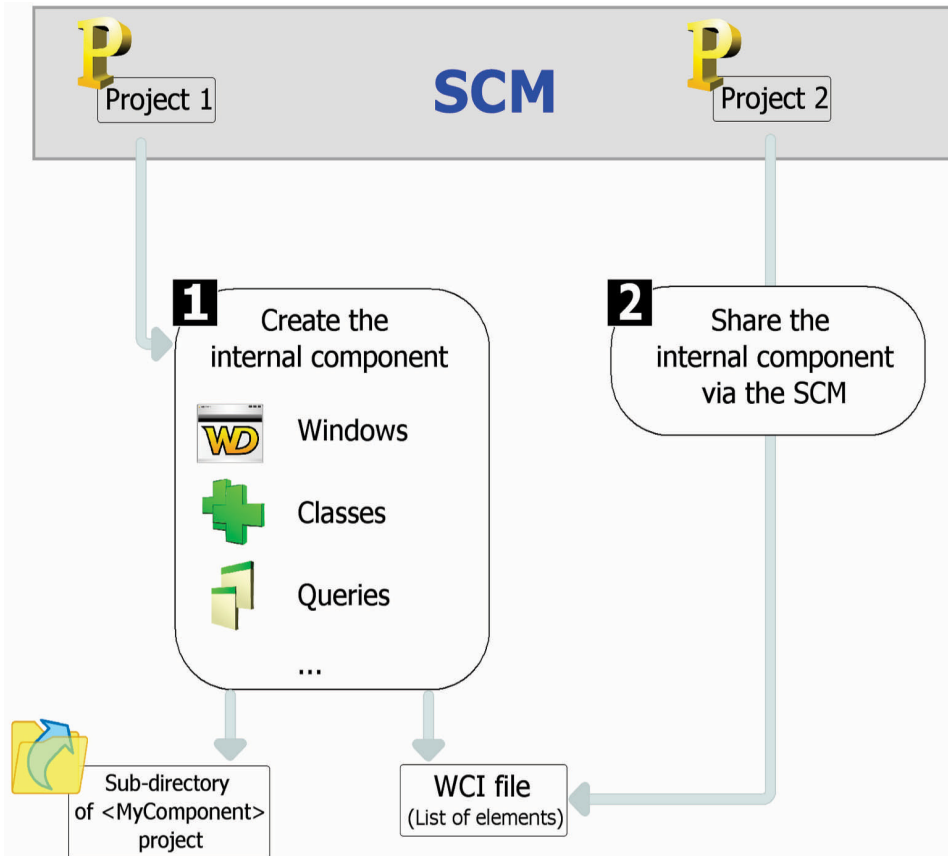
- manage the SCM databases (creation, connection to a scm database).
- manage the branches.
- manage the files and directories found in a project of the source database (Add, delete, rename, ... files and directories).

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

Internal component

An internal component is used to group several project elements. This grouping allows you to:

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



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

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

Internal component in practice

1 Overview

An internal component is used to group several project elements. This grouping allows you to:

- **Organize a project:** you have the ability to create internal components to group the project elements (by feature for example).
- **Share elements among several projects:** The elements found in an internal component can be shared among several projects via SCM. See "Sharing the internal components (via SCM)" (page 71) for more details.

One of the benefits of the internal component compared to an external component is that the internal

component can be debugged directly from the project that uses it.

Differences with an external component: An internal component allows you to include all the elements of the component in the interface of the project containing the internal component. 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, the corresponding project must be opened.

2 Creating an internal component

2.1 The different steps

To create an internal component:

1. Select "Workshop .. Internal component .. Create an 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 corresponding to the description of the internal component. This name will also be used to create a sub-directory in your project containing all the elements of the internal component.
- its caption.
- its description.

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

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

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

component. The inaccessible elements (or "private" elements) can only be used by another element of the internal component (the completion will propose these "private" elements only from the elements of an internal component).

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

- **Full share:** The internal component accesses the data files of the project. The internal component and the project use the same runtime contexts. This mode corresponds to the default mode if the component uses no analysis.
- **Data share only (advanced mode):** The internal component accesses the data files of the project. The internal component and the project use different runtime contexts. This mode is reserved to specific cases.
- **Standalone:** The internal component accesses its own data files. The internal component and the project use different runtime contexts. This mode corresponds to the default mode if the component uses an analysis.

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

6. Validate the creation of the internal component. You will have the ability to:

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

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

2.2 Internal component and analysis: case of full autonomy

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

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

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

3 Sharing the internal components (via the SCM)

The internal components can be shared among the projects via SCM.

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

- create the internal component from a project

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

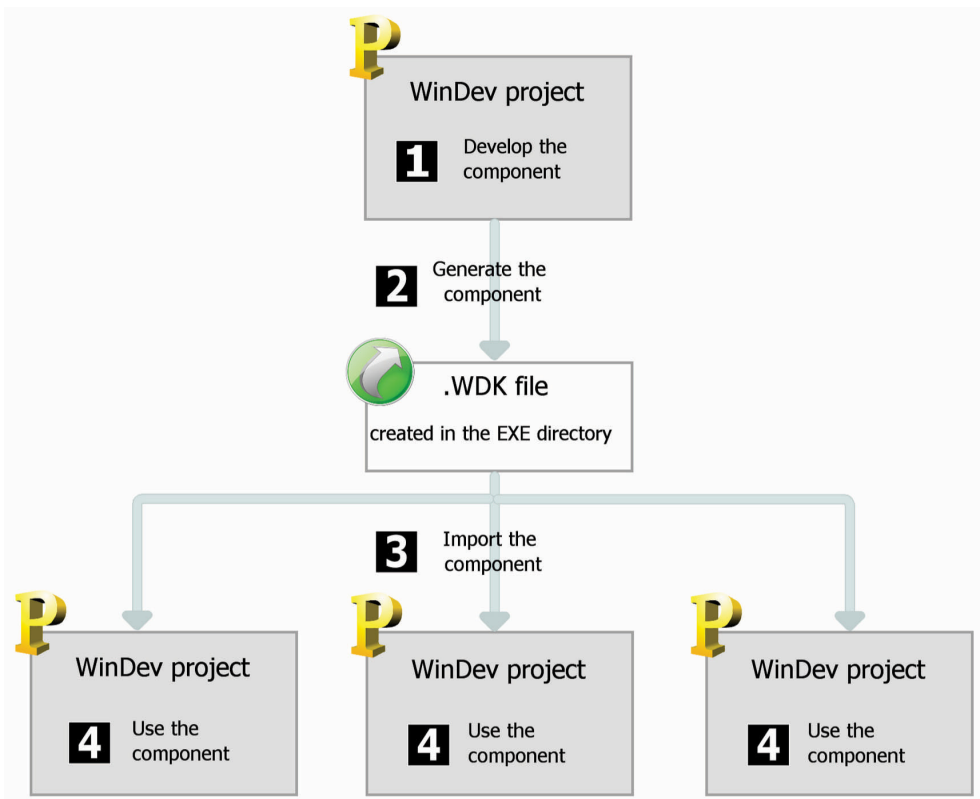
- import a project containing one or more internal components into SCM. The internal components will be automatically included in SCM.

External component

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

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

An external component can be distributed to other WinDev developers (for free or at a cost). These developers will be able to easily include the feature proposed by the external component in their application without having to access to the corresponding source code (if it has not been distributed as well). Then, the external component will be included in the application and distributed with it.



External component in practice

1 Overview

1.1 Definition

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

- Sending faxes
- Sending emails
- ...

Notes:

- To optimize the organization of your projects, you have the ability to use internal components. See "Internal component in practice", page 70 for more details.
- In the rest of this chapter, "component" means "external component".

A **component can be distributed to other WinDev developers** (for free or not). These developers will be able to easily include the feature proposed by the component in their application. The component will be included in the application and distributed along with the application.

When creating the component, the author specifies:

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

Note: Several component examples are provided with WinDev. These components are accessible from the "Wizards, Examples and Components" pane ("Components" option).

1.2 What is a component made of?

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

All the component elements are independent from the project where the component is included. The component can have its own analysis, windows, reports, data files, ... When creating the component generation, specify whether these elements will be made available (or not) to the WinDev project that includes this component.

The component is made of three files:

<Component-Name>.WDK	Component file. Contains all the component elements. This file is required to include the component in a project. This file must also be supplied in the setup program of the application that uses the component.
<Component-Name>.WDI	Description of the component (for developing applications only). This file is required to include the component in a project. This file must not be supplied in the setup procedure of the application that uses the component.
<Component Name>.WDO (optional file)	Optional file File in text format containing the list of the additional elements supplied with the component (data files, .JNI files, ...). See "The WDO file", page 78.

1.3 What is an external component made of?

A component contains:

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

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

Note: By default, if a component uses an analysis, data files, ... the WLanguage functions for Hyper-

FileSQL management will handle these elements in an independent context. These parameters can be modified in the advanced options of the component. See the online help for more details.

2 Creating and generating an external component

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

The creation of a component is performed in several steps:

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

The component can be distributed and re-used.

3 Distributing an external component

3.1 Overview

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

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

3.2 Direct use of the component

The component is created and used on the same computer.

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

Caution: if the component uses specific files (data

files, text files, ...), you need to create a <ComponentName>.WDO file after generating the component. This file contains the list of external files (data files, text files, ...) used by the component.

These files referenced in <ComponentName>.WDO will be automatically copied to the EXE directory of the project that uses the WinDev component.

3.3 Simple distribution of a component

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

When importing the component into a WinDev project, select the WDI file corresponding to the component in the distribution directory.

To distribute a component, you must supply:

- The files automatically generated by WinDev (<ComponentName>.WDK and <ComponentName>.WDI files).
- If necessary, the specific files handled by the component (data files, initialization files, ...) as well as <ComponentName>.WDO.
The <ComponentName>.WDO file contains the list of files that must be supplied with the component. See "The WDO file", page 78 for more details.

3.4 Distribution via a setup procedure

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

This setup mode is used to automatically manage:

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

To propose a setup procedure for a component:

1. Create the setup procedure of the component:

- directly via the "Workshop .. External Component .. Create the setup procedure of a component" option.

- from the list of components found in the project ("Workshop .. External Component .. List of components generated from this project", "Setup" button).

2. In the wizard planes, specify:

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

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

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

4 Using an external component in an application

A component can be re-used at anytime in any WinDev application: all you have to do is identify the directory of the component files and import it into the project. The component can also be imported via the SCM.

When a new version of a component is available, all you have to do is install the new files of this update in the setup directory of the component (according to the setup mode used). See "Distributing an external component", page 74 for more details.

nal component", page 74 for more details.

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

The method for creating the executable that uses the component is the same as the method for creating the executable for a standard WinDev application. The DLLs required for the component to work are automatically detected.

5 Deploying an application that uses an external component

5.1 Overview

The method for deploying an application that uses one or more components is the same as the method for deploying a standard application ("Workshop .. Create the setup procedure").

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

- <ComponentName>.WDK,
- <ApplicationName>.EXE,
- the files needed for the application's component to work.

5.2 Updating the components and the deployed executables

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

- the executable of the application, the component and the files required for them to work.
- the executable of the application and the files required for the application to work.
- only the component and possibly the files needed for it to work.

See the online help for more details.

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

1. The recompilation of the host project

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

earlier versions of 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 the problems of an earlier version.
- New features have been added into the component but they are not required to run the application.

6 Modifying an external component

6.1 Overview

A component that was created and generated can be modified at any time. The following modifications can correspond to:

- the addition of elements into the component.
- the deletion of elements from the component.
- the modification of rights on the component elements.
- the modification of one of the component elements.

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

6.2 The different types of compatibility

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

may encounter runtime problems if they are not in phase with this component.

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

- **backward compatibility:** the version of the component (.WDK file) used to compile the applications that use this component must always be greater than or equal to the version currently deployed

This compatibility mode is automatically managed. If the version number of the component installed with the application is lower than the version number used by the application, the application will not work.

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

7 Advanced characteristics of a component

7.1 Automatic documentation

A component must be supplied with a documentation in order to be reusable.

WinDev simplifies the creation of this documentation by proposing:

- **a general overview of the component.** This general overview is entered when the component is generated. When the component is used in a

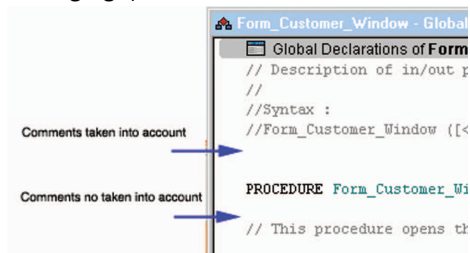
WinDev application, this overview will be visible when the component found in the "Wizards, Examples and Components" pane is hovered by the mouse cursor. See the online help for more details.

- **an automatic generation of the technical documentation** from the comments inserted into the source code of the component elements.

Which code comments are taken into account?

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

- The comments found at the beginning of the WLanguage procedures



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

When is the documentation generated?

The documentation of the component is generated during the first generation of the component ("Workshop .. External Component .. Generate a component").

When generating this documentation:

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

Note: Whenever the component is re-generated, you have the ability to re-generate the documentation associated with the component ("Regenerate" button in the "Component history" window).

Caution: If the "Regenerate" button is used, the modifications performed in the generation wizard will be deleted.

How do I access the documentation of a component?

The general presentation of the component is automatically displayed when the component found in the "Wizards, Examples and Components" pane is hovered by the mouse cursor.

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

- when a component is included in a project ("Workshop .. External Component .. Import a component into the project").
- when double-clicking the icon of the component found in the "Wizards, Examples and Components" pane or via "Description" from the popup menu of the component.
- in the description of the components included in the project ("Workshop .. External Component .. List of components imported into the project"). The "Description" button is used to get information about the component.

The **documentation specific to an element** of the component can be displayed:

- by double-clicking this element ("Project explorer" pane or project graph) or by pressing [F2] from the code of this element.
- in the list of components included in the project ("Workshop .. External Component .. List of components imported into the project").

7.2 Visibility of a component element

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

- If the element is accessible, the user of the component will see this element in the list of project elements. The user of the component will be able to handle these elements by programming (as if it were an element created in the project).

Note: However, the code of this element is not visible.
- If the element is not accessible, the user will not even know that this element exists.

Caution: Depending on the declaration mode of the project elements (class, set of procedures, ...), the accessible elements may vary. The table found on the next page summarizes the different methods for accessing an element according to its declaration mode.

7.3 The WDO file

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

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

These two files must be distributed along with the component.

If the component uses additional elements (data files, ...), you must add into the EXE directory of the project:

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

What is this WDO file?

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

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

This ".WDO" file can contain:

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

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

When including the component, the tree structure specified in the WDO file will be stored and reproduced in the EXE directory of the project. See "Using an external component in an application", page 75 for more details.

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

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

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

Distributing a component with WDO

To distribute a component that uses a WDO file:

- If no setup procedure is used for the component, you must supply:
 - the WDK file
 - the WDI file
 - the WDO file
 - all the necessary files referenced in the WDO file.
- If you are using a setup procedure for the component, the WDO file will be automatically created when creating the setup procedure of the component. In the wizard you must:
 1. Request the modification of the files to install ("Modify the list of files to install" in the "Additional modules").
 2. Select the additional files to install. The WDO file will be automatically created and installed with the component.

Generation modes

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



Applications

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



Services

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



Libraries and patches

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



External components

External components are application bricks allowing you to share one or more specific features among several applications. An external component generated with WinDev can also be used in a WebDev or WinDev Mobile project.



Webservices

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

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



Applications and .Net assembly DLLs

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



Java applications and applets

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

A Java Applet is an application meant to work inside a Web browser.

Creating the executable in practice


Once your application is entirely described (windows, reports, ...), you must create the corresponding executable program.

The executable program (".EXE" file) can be run directly: there is no need to start the application from WinDev.

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

1 Creating the executable program of a WinDev application

To create an executable program:

1. Select "Workshop .. Executable .. Generate the Windows executable" or click . The wizard for executable creation starts.

2. Specify:

- the **general options** of the executable (executable name, executable program icon, splash screen, ...).
 - the **operating mode** of the executable:
 - **Multi-instance**: This program can be started several times from the same computer; several instances will be open at the same time.
 - **Single-instance**: This program can be started only once from the same computer; a single instance will be open at a time. If the developer tries to start another instance of the program, the instance already running will be enabled.
 - the **operating mode of the library**. The library is optional. It can be created in a separate file (".WDL" extension) or included in the executable. Including the library in the executable program presents the benefit of having a single file to install or update, but it increases the size of the created ".EXE" file.
 - the **error message** that will be displayed if a programming error occurs.
 - whether the input of User Macro-Code (UMC) is taken into account or not. If the UMC is enabled, the end users will be able to create their own macros to modify the application.
 - the use of **patches for updates**. In this case, all you have to do is copy the file corresponding to the patch into the directory of the executable.
 - the languages supported by the executable.
 - the **files found 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's data files** (only if your application manages the user groupware).
 - **the location of the data files** that must be used by the executable.
 - the **version information**. This information will be displayed in the "Version" tab of the window for executable properties (accessible from the Windows explorer). The **version number** can be automatically **incremented** whenever the executable is created.
 - the mode for using the WinDev framework (using a framework specific to the application or a framework common to all the WinDev applications installed on the computer).
 - the manifest to use (for an execution on Windows Vista and above only).
 - whether the executable must be saved in the reusability center.
 - whether a **project backup must be performed**
3. Validate the executable description. The executable is automatically generated according to this information. The executable can be directly run after its creation.

Java application in practice

1 Overview

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

1.1 What is a Java application?

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

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

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

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

2 Features for a Java project

2.1 Creating a Java project

A Java project is created like any other WinDev project. All you have to do is select "File .. New .. Project". The wizard for project creation starts. In one of the first screens, the wizard allows you to choose the generation mode of your project. Select "Java" and follow the wizard.

By default, a WinDev project for Java corresponds to a specific configuration of the project. The type of project generation can be modified at any time. To do so, create a new project configuration.

1.2 Why develop a Java application with WinDev?

You own a WinDev application that works in Windows and you want to make it accessible regardless of the operating system? Generate a Java application to allow the users who are working on a MAC OS X system for example to use your application.

This enables you to share your network applications.

1.3 Principle for generating a Java application with WinDev

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

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

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

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

Note: we recommend that you create a new WinDev project to develop a Java application rather than transforming a WinDev application into Java.

2.2 JAVA RAD


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

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

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

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

2.3 Test of a Java project in the editor

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

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

2.4 Compiling a Java project in WinDev

The compilation of a Java project is used to:

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

See the online help for more details.

Note: as indicated in the section "What is a Java application?", page 83, the Sun J2SDK must be installed on the development computer to compile your project in Java.

2.5 Creating a Java archive

WinDev enables you to:

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

To create the Java archive, all you have to do is start the executable creation ("Workshop ... Java .. Generate Java archive"). A wizard helps you create the Java archive.

2.6 Deploying a Java application

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

To start the wizard for setup creation, select "Workshop .. Create the setup procedure".

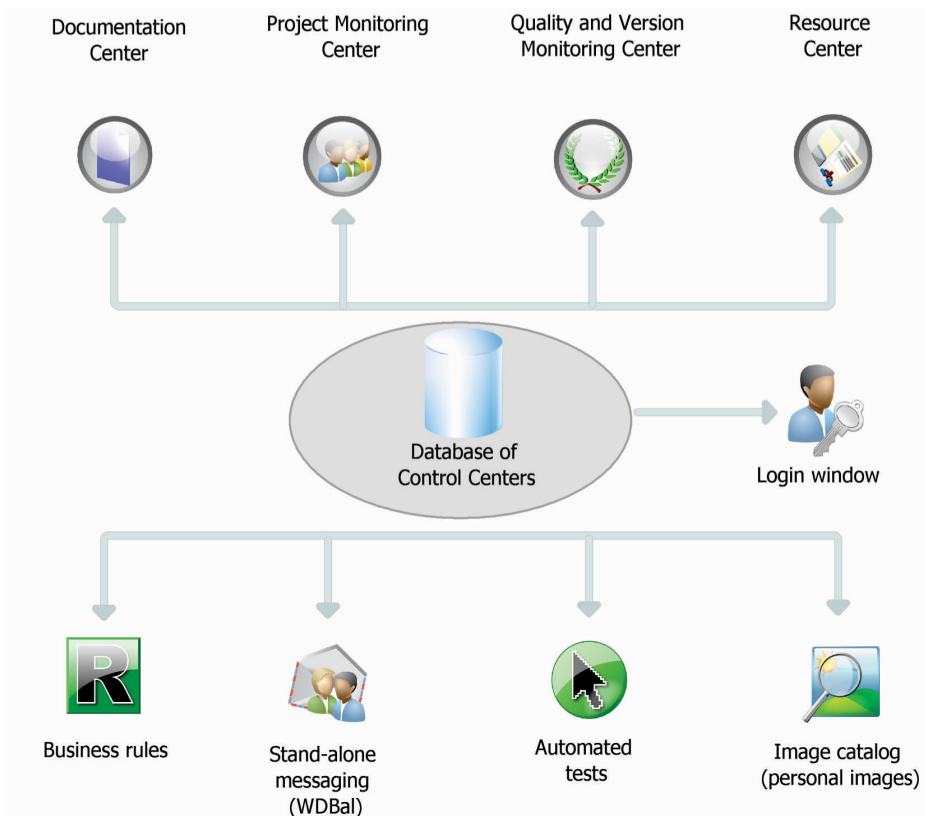
See the online help (keyword: "Java") for more details.

The Control Centers

To manage the life cycle of your development projects, WinDev proposes several Control Centers. The Control Centers are a set of tools intended to handle:

- the requirements of a project (assessments)
- the development, test and documentation tasks
- bug tracking and change requests sent back by users
- the business rules used by the company in its developments

Control Centers work with a database (HyperFileSQL Classic or Client/Server) shared among a large number of tools available in WinDev but also with WebDev and WinDev Mobile if you use them as well:



As soon as WinDev is installed, it will be proposed:

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

Managing the requirements

A requirement is a need that a product (or a service) should fulfill.

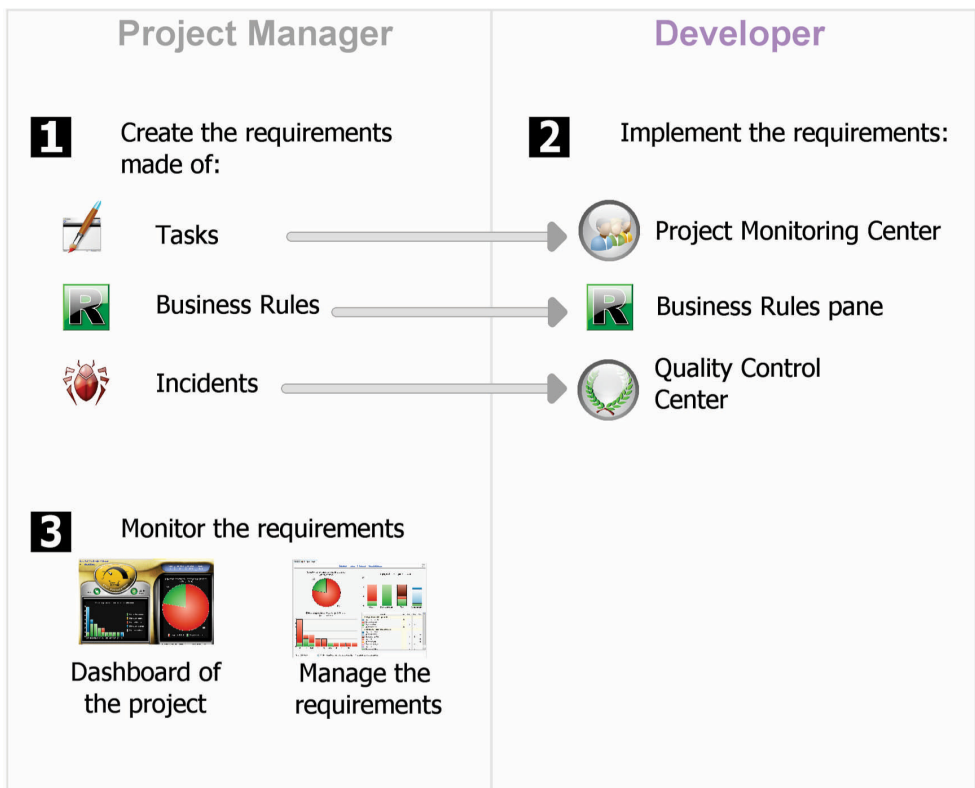
The Control Centers allow a project manager to manage the requirements of a development project.

To do so you must:

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

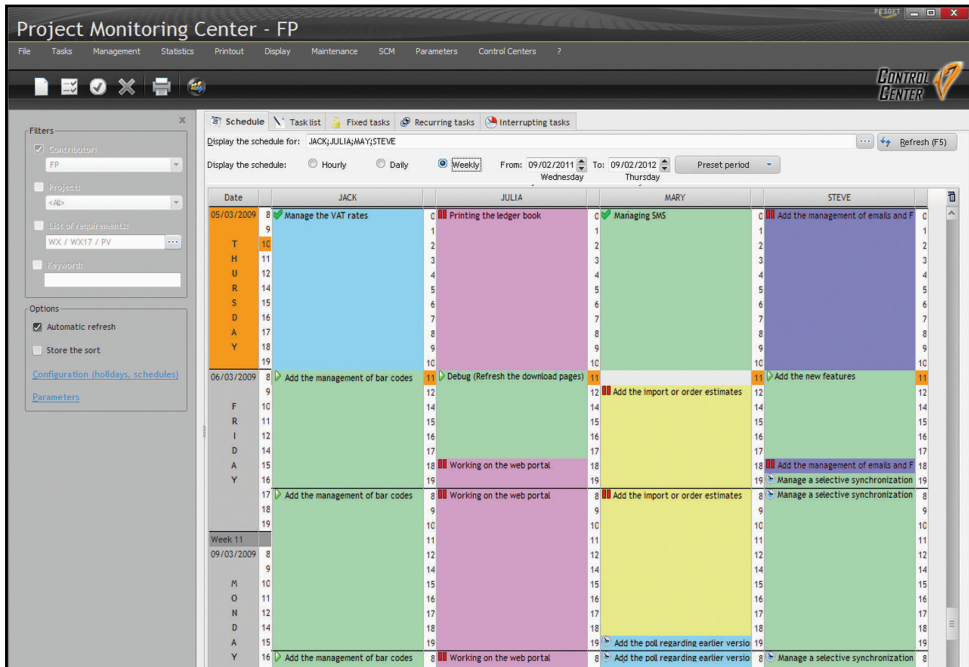
Each developer performs the different tasks assigned to him.

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



Project Monitoring Center

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



Operating mode of the Project Monitoring Center

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

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

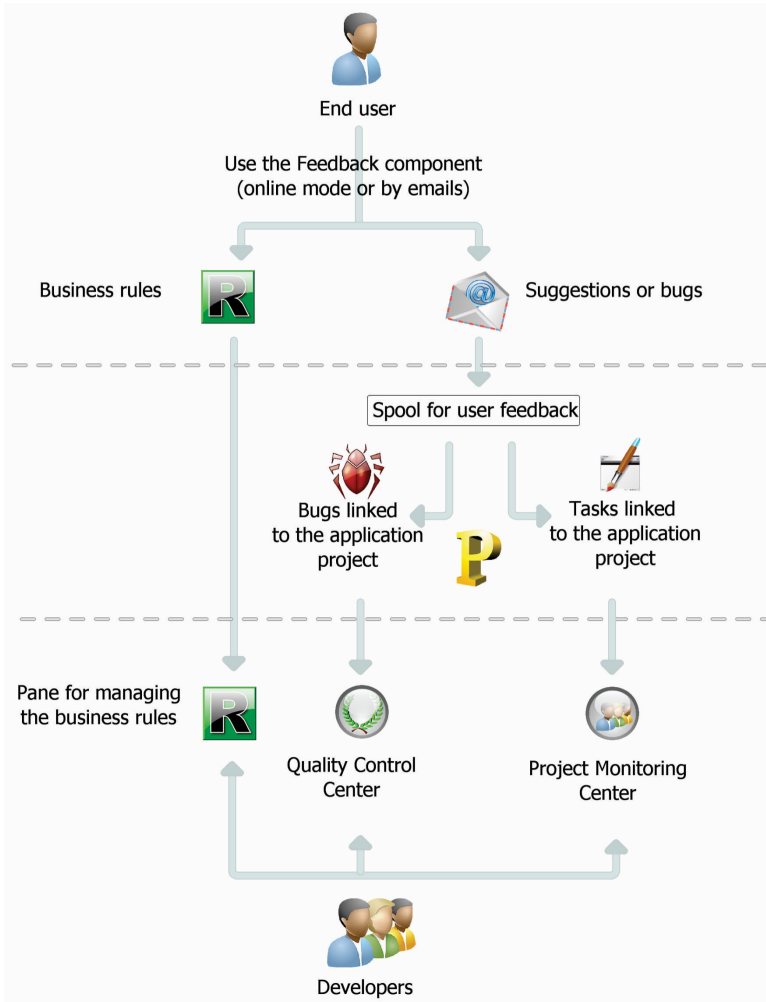
A task can be linked to a project element (window, report, ...). Whenever the relevant element is opened, the time spent on this element is counted and stored in the monitoring center. Conversely, the element corresponding to the task that you want to perform can be automatically opened from the task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

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

Managing the suggestions and incidents

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

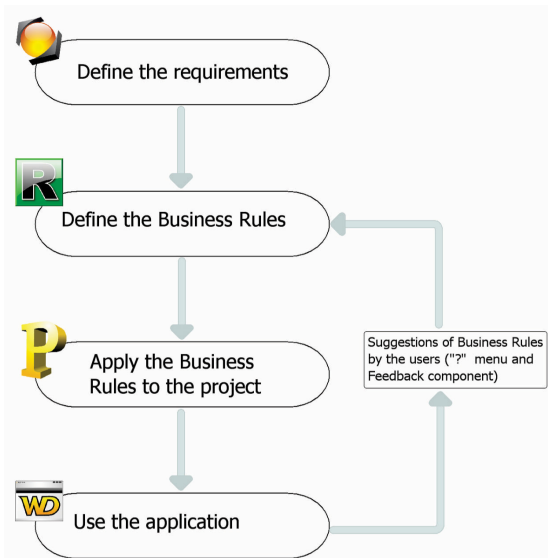
If your application contains an automatic help menu ('?' with the "Send suggestions...", option), the users of your application will be able to send suggestions, bugs or business rules.



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

Managing the business rules

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



A business rule can be simple or complex.

The business rules can come from:

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

During 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 rule currently implemented.



PART 3

Databases



Express



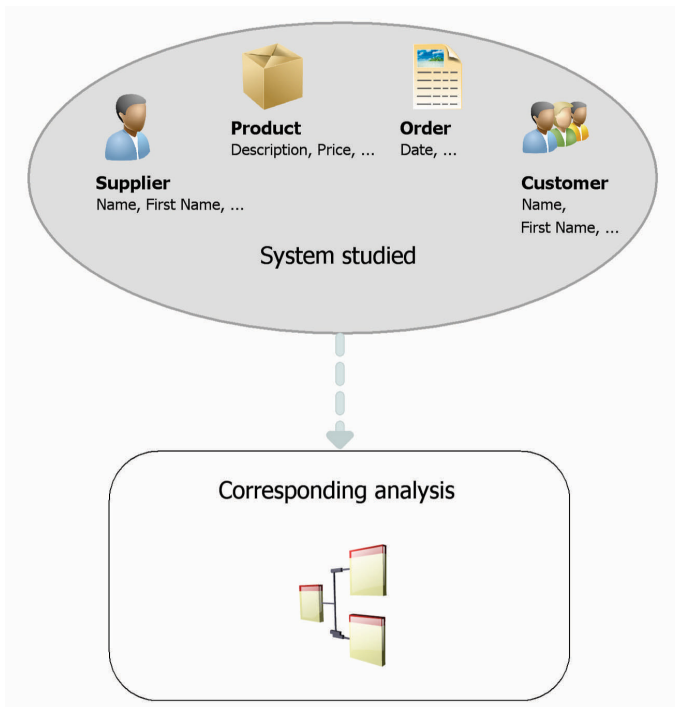
DEVELOP 10 TIMES FASTER

PCSOFT

Analysis: Structure of the database

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

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



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

In the analysis, the description of a data file can be linked to a type of data file (HyperFileSQL, Oracle, ...).

The analysis in practice

1 Overview

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

Two methods are available to describe an analysis:

- **1st method:** Creating the analysis directly (which means a Logical Data Model - LDM). This operation is presented in details in "Creating a LDM

(Logical Data Model)", page 94.

- **2nd method:** Creating the Conceptual Data Model (CDM) and creating the analysis from the CDM. These operations are presented in details in "Creating a CDM (Conceptual Data Model)", page 99 and "Generating the LDM from the CDM", page 101.

This chapter presents both methods.

2 Creating a LDM (Logical Data Model)

In a WinDev application, 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. Select "File .. New .. Analysis". The wizard for analysis creation starts.

2. Specify:

- **the name and directory of the analysis.** The analysis corresponds to a ".WDA" file. By default, this file will be created in the directory of the project analysis (<Project Name>.ANA directory). This directory must be accessible in read/write.
- **the caption of the analysis** that briefly describes the purpose of the analysis.
- **whether the analysis must be associated with the current project.**
- whether **the analysis must be password protected.** This password will be requested when opening the analysis (via the WinDev tools or by programming).
- the types of databases handled by the project.

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

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

2.2 Adding a data file

The data files in analysis are used to describe the structure of the data files used by the project. A data file found in the analysis can be:

- A new data file.
- A preset data file, supplied with WinDev.
- A data file imported from an existing database (using a specific format for example).

The following paragraphs explain how to create a data file.

To create a new data file:

1. Select "Insert .. Data file". The data file creation wizard starts.

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

3. Specify:

- **the logical name of the data file.** This name will be used to handle the data file.
- **a short description of the data file** that summarizes the subject of the data file.
- **the representation of a record** in the data file. This representation improves the meaning of the questions asked when describing the links. This option must be preceded by an indefinite article (A or AN).
- whether the data file includes an **"Automatic identifier" item.** The value of this item is unique for each record and it is automatically calculated by WinDev.

- **the type of the database for which the file will be created.** Depending on the selected type, this data file will be handled by the HyperFileSQL engine, by an OLE DB driver or by one of the native accesses of WinDev (SQL Server, Oracle, ...).
- whether **the size of the data file can exceed 2 GB.** This option can only be used if both the file system of the current computer and the file system of the computers where the application will be deployed support the NTFS format (Windows 2000, NT or XP).
Caution: If this option is checked, this file cannot be opened in Windows 95, 98 or Me.
- whether the **data file must support the replication.** This option is used to manage the automatic updates of identical and remote databases. See the online help for more details.

4. The created data file becomes the current file. The description window of the file items is automatically opened. It enables you to describe the items of the data file.

To import a preset data file description:

1. Select "Insert .. Data file".
2. Check "Select a description among the preset data files".
3. Choose the preset data file. This data file will be imported into the current analysis. This data file can be modified later.
4. Select the items of the data file to keep. These items can be modified later.
5. Specify whether the links must be automatically sought. If this option is checked, the items with the same name will be linked.
6. The imported data file is automatically inserted into the current analysis.

Note: To modify a data file or the items of a data file, select the data file and enable "Structure of files .. Description of data files" or "Structure of files .. Items".

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

Solution 1: from the data model editor

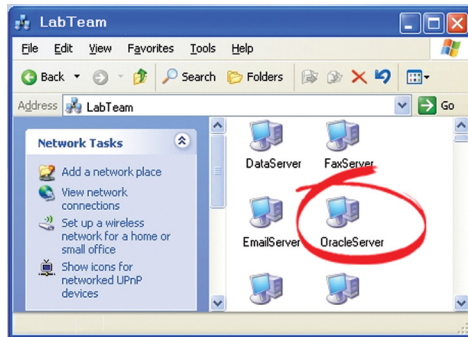
1. Select "Insert .. Data file".
2. Check "Use the data files of an existing database" then select the type of the database.
3. Specify the data format used by your project. You can keep the current format or migrate the files to HyperFileSQL format.
4. Select if necessary) the WinDev analysis into which the description must be imported (existing analysis or new analysis).

5. Specify the source database containing the descriptions to import and the type of this source database. Depending on the selected type, specify the requested information.
6. Select the tables or files whose description must be imported and validate.
7. The imported data file is automatically inserted into the current analysis.

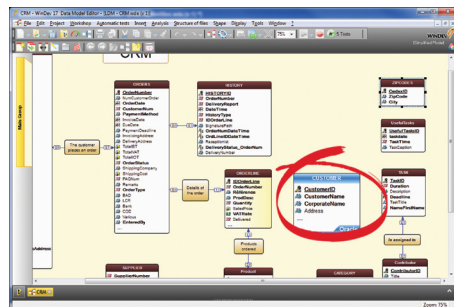
Solution 2: From Windows explorer

Directly drop the data file description from Windows explorer into the data model editor.

For example:



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



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

2.3 Creating an item

To create an item:

1. Double click the data file in which the item must be created. The description window for the items of the data file is opened.
2. Click the first empty row in the table of items.
3. Specify the name, caption and type of the item in the table.
4. In the right section of the screen, specify the details about the new item (type, size, default value, sort direction ...).
5. In the bottom section of the screen, specify the details about the shared information.
6. Validate the description of the item.

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

2.4 Creating a link

Different types of links can be created between the files. See "Characteristics of the links defined in an

analysis", page 96 for more details.

To create a link:

1. Select "Insert .. Link". The mouse cursor turns into a pen.
2. Select the two data files to link. The description window of the link is automatically opened.
3. To define the cardinality between the two data files:
 - select the cardinalities among the proposed ones (0,1; 1,1; 0,N; 1,N).
 - answer the questions asked. The cardinalities will be automatically updated.
4. To describe the advanced cardinalities, check "Display the advanced cardinalities" and answer the questions asked.
5. Enter the caption of the link by briefly describing the purpose of the link.
6. Specify the keys to link.
7. Specify the integrity rules. These rules are used to ensure the data integrity when one of the relation keys is modified or deleted.
8. Validate. The link is automatically created.

3 Characteristics of the links defined in an analysis

3.1 Owner file and member file

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

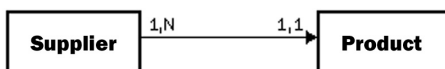
- the owner file is the owner of the key.
- the member file is a member of the analysis files containing a copy of the key.

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

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

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

This type of link is represented as follows:



3.2 The cardinalities

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

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

1. For each record of the data file, to **at least** how many records in the other data file this record is linked?

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 in the data file, what is the **maximum** number of records in the other data file it is linked to?

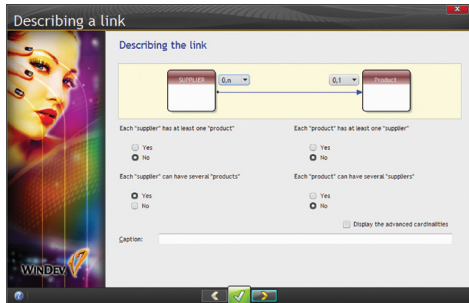
This answer provides the second part of the cardinality (**maximum cardinality**):

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

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

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

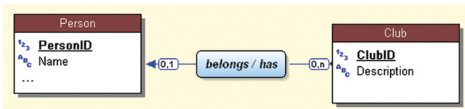
To be clearer, these two questions can be broken down into four questions. For example, to describe a link between the Supplier file and the Product file:



- each "supplier" has at least one "product": Yes/No?
- each "supplier" can have several "products": Yes/No?
- each "product" belongs to at least one "supplier": Yes/No?
- each "product" can belong to several "suppliers": Yes/No?

Example of cardinalities:

This example presents two different cardinalities:



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

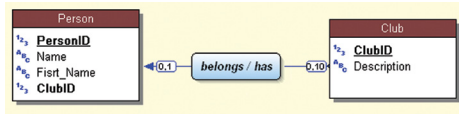
3.3 The advanced cardinalities

The cardinalities can be defined more precisely. We talk about **advanced cardinalities** then. These cardinalities are used to exactly specify the minimum cardinality and the maximum cardinality.

To define the advanced cardinalities, check "Display the advanced cardinalities" in the description of the link ("Insert .. Link").

Example of advanced cardinalities:

This example presents two different cardinalities:



- **Cardinality 0,1:** A person can be a member of only one sport club. This person is not required to belong to a sport club.
- **Cardinality 0,10:** A club can have up to 10 members.

3.4 The referential integrity

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

The referential integrity consists in checking that:

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

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

See the online help for more details.

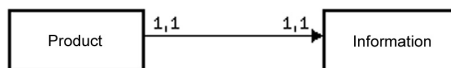
3.5 The different types of links

Several types of links can exist between the data files:

- parallel
- optional
- complement
- shared
- complex

Parallel link

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



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

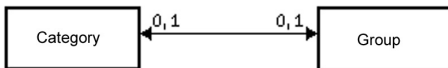
Note: This type of link is rare because the two files can be combined into a single file.

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

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

Optional link

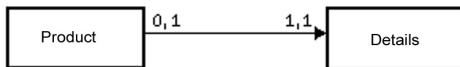
For an optional link, **each record** found in a data file (Category) is associated **with no record or with a single record** in another data file (Group). **Each record** of the other data file (Group) is associated **with no record or with a single record** in the first data file (Category).



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

Complement link

For a complement link, **each record** of a file (Product) is linked **to no record or to a single record** in another file (Details). **Each record** of the other data file (Details) is **necessarily associated with a record** in the first data file (Product).



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

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

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

Shared link

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

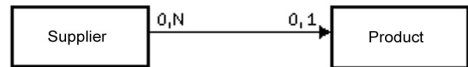
To manage a shared link, the key of the Supplier file is copied into the Product file. It becomes a multiple key to increase the speed of integrity check.

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

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

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

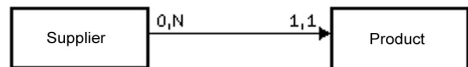
Shared link with a 0,N - 0,1 cardinality



With this type of link:

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

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



With this type of link:

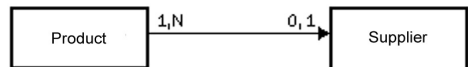
- an owner may have no member (a supplier may have no product).
- each member has a single owner (each product has a single supplier).

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

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

With this type of link:

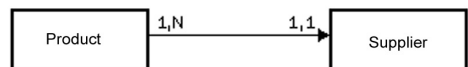
- each owner has at least one member (a supplier has at least one product).
- a member may have no owner (a product may have no supplier).



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

With this type of link:

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



Complex link

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

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

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

A complex link is made of two shared links.

Example of complex link

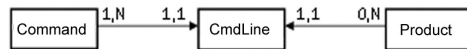
An order (Orders file) can contain one or more products. A product (Product file) can be used in several orders.

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

The OrderLine file contains:

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

The links between the data files are as follows:



4 Operations available on an analysis

WinDev enables you to easily perform the following operations on an analysis:

- Duplicating/Copying a LDM: Duplicating a LDM allows you to have two identical LDMs with different names.
- Deleting a LDM.
- Renaming a LDM.
- Associating a LDM with a project.
- Enlarging or reducing the display of a LDM in the editor.
- Moving the display of the LDM in the editor.

WinDev lets you perform the following operations on a data file found in the data model editor:

- Duplicating/coping a data file.
- Deleting a data file.
- Renaming a data file.

WinDev lets you perform the following operations on a data file found in the data model editor:

- Duplicating/copying an item.
- Deleting an item.
- Rename an item.

See the online help for more details.

5 Creating a CDM (Conceptual Data Model)

In the MERISE methodology, the conceptual data model defines the objects and the actions that must be taken into account in the database. This model only represents the structures of information and the exchanges of information for the system used, independently of the physical constraints linked to the nature of the database used.

5.1 Creating a CDM

To create a CDM:

1. Select "File .. New .. CDM". The wizard for creating a CDM starts
2. Specify:

- **the name and directory of the CDM.** The CDM corresponds to a ".MCD" file. By default, this file will be created in the directory of the project analysis (<Project Name>.ANA directory). This directory must be accessible in read/write.

- **the caption of the CDM**, which briefly describes the object of the CDM.
- **whether the CDM must be directly associated with the current analysis.** This analysis can be an existing analysis or a new analysis that will be created.

3. The created CDM becomes the current CDM. WinDev automatically proposes to create the first entity.

4. Create all the elements (entities, associations, properties and relationships) of you CDM.

5.2 Creating an entity

To create an entity:

1. Select "Entity .. New entity". The description of the entity is displayed.
2. Specify:
 - **the name of the entity.** This name will be the

logical name of the file created when the LDM is generated from the CDM.

- **the caption of the entity** that summarizes the subject of the entity.
 - whether an **"Automatic identifier" property** must be defined in the entity. The value of this property is unique for each record and it is automatically calculated by WinDev.
 - **notes** presenting the operating mode of the entity ("Various" tab).
3. The entity becomes the current entity. WinDev automatically proposes to describe the properties of this entity.

5.3 Creating an association

To create an association:

1. Select "Association .. New association". The description of the association is displayed.
 2. Specify:
 - **the name of the association.** When generating the LDM from the CDM, this name will be the logical name of the created data file if this association includes at least one property or if it belongs to a complex link.
 - **the caption of the association**, which briefly summarizes the subject of the association.
 - **the notes** presenting the operating mode of the association.
3. The association becomes the current association. WinDev automatically proposes to describe the properties of this association.

5.4 Creating a property

To describe a property:

1. Double-click the entity or the association in which the property must be created. The window for describing the properties is displayed.
2. Click the first empty row in the table of properties or click "+".
3. Specify the name, the caption, the type, the size and the notes of the property.
4. For an entity property, check "Identifier" if the property is the identifier of the entity.
5. Validate the description of the property.

5.5 Creating a relationship

To create a relationship:

1. Select "Entity .. Trace a relationship". The mouse cursor turns into a pen.
2. Select the two entities to link or the entity and the association to link.
3. Define the cardinalities of the relationship.

5.6 Defining the cardinalities

To define the cardinalities:

1. Select the relationship whose cardinalities must be defined.
2. Select "Entity .. Information about the relationship".
3. Choose the cardinality among the proposed ones.
4. Validate the description of cardinalities.

6 Main operations that can be performed on a CDM

WinDev enables you to easily perform the following operations on a CDM:

- Delete a CDM.
- Rename a CDM.
- Specify a zoom factor to reduce or enlarge the display of the current CDM.
- Moving the display of a CDM.

WinDev enables you to perform the following operations on a CDM entity:

- duplicate/copy an entity.
- delete an entity.
- Rename an entity.

WinDev enables you to perform the following operations on a CDM association:

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

WinDev enables you to perform the following operations on a CDM property:

- Delete a property.
- Rename a property.

See the online help for more details.

7 Generating the LDM from the CDM

To associate a CDM to a project, you must generate the associated LDM.

Reminder: It is the LDM that is associated with the project (not the CDM).

The generation of the CDM is used to automatically create the corresponding LDM once the validity of the CDM has been checked.

To generate the LDM from the CDM:

1. Select "Analysis .. Generate the logical model".
2. Define the parameters for generating the LDM from the CDM:

- **the name and path of the generated LDM.** The LDM 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.
- whether **the captions of the associations without property** (of the CDM) are stored to identify the links (of the LDM).
- **the type of access to the data file.** Depending on the selected type, the data files will be handled by the WinDev engine, by an OLE DB driver or by one of the native accesses of WinDev (SQL Server, Oracle, ...).
- **the type of the database** used.
- the format of the characters used in the analysis.

3. Validate the generation. The LDM corresponding to the CDM is automatically created.

Special case: Updating an existing LDM

If the name and path of the LDM specified when generating the CDM correspond to an existing LDM, the existing LDM will be updated with the new description of the CDM.

When updating an existing LDM, the generation checks:

1. The existence of the entity in the LMD:

- if the entity is new, the corresponding data file is created.
- if the entity already exists, its properties are updated.

2. The existence of the properties in the LDM:

- if the property is a new one, the corresponding item is created.
- if the property is already found, with the same description, no action is performed.
- if the property already exists, with a different description, a conflict occurs. The data model editor proposes:
 - to ignore the new description of the property and to perform no modification.
 - to update the existing item with the new description of the property.
 - to create a new description synonym with the new description of the property.

3. The existence of relationships and associations.

When the association requires the creation of one or more relationship files, the data files and the LDM items are updated in the following way. When a relationship has been modified, the link of the LDM is recreated or updated:

- if the identifying properties of one of the entities have been modified, the cardinality of the existing links are updated and new links are created.
- if no identifying properties of any entity has been modified, cardinalities are simply updated.

Notes:

- The deletion of entities, relationships or associations is not taken into account when a CDM is generated into a LDM.
- The deletion of the entity properties can be taken into account by unchecking "Keep the LDM modifications" in the options for generating the LDM from the CDM. In this case, the data files of the LDM only contain the items of the corresponding entity in the CDM after the conversion.

8 Generating the analysis (LDM)

The generation of the analysis is performed before the programming phase and after the description of the data files. This generation is used to:

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

As long as the analysis (the LDM) has not been generated, the description of the analysis (i.e. the data files) cannot be used in the project.

The generation ("Analysis .. Generation") includes three steps:

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

9 Managing the analysis versions

WinDev enables you to manage the different versions of your analysis ("Analysis .. Version management").

You have the ability to work with the current analysis or with a specific analysis.

Note: when an analysis is damaged, you can directly select the damaged analysis to restore one of the previous versions for instance.

All the versions of the analysis are viewed in a graph. The yellow square indicates the generation number of the analysis and the caption indicates the generation date.

The "small rectangles" displayed between two versions indicate:

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

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

The options available in the version manager are:

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

9.1 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: you will not be able to open data files corresponding to an analysis with a version number greater than the restored version. In this case, you must also restore the data files corresponding to the version of the restored analysis or delete the existing data files in order to re-create them.

9.2 Canceling the last generation

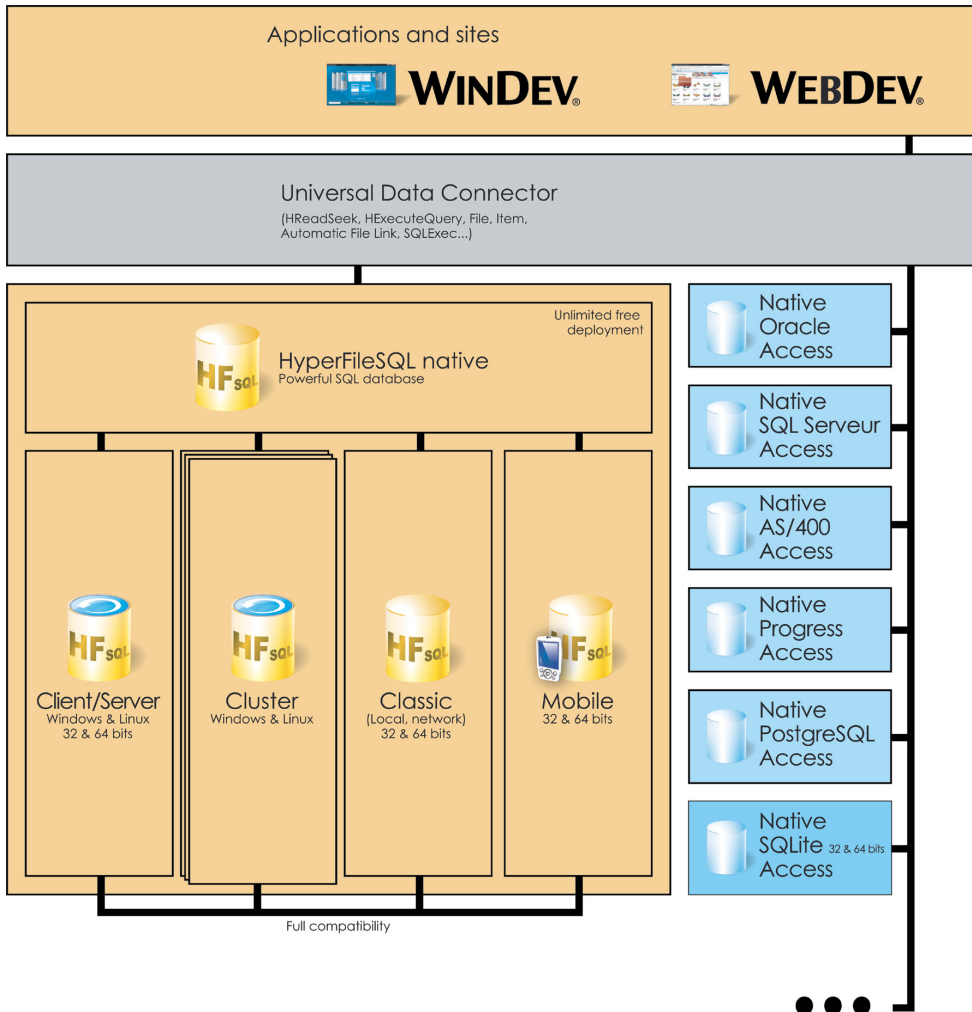
Canceling the last generation of the analysis enables you to restore the analysis in its state before the last generation. The modifications performed since are not applied.

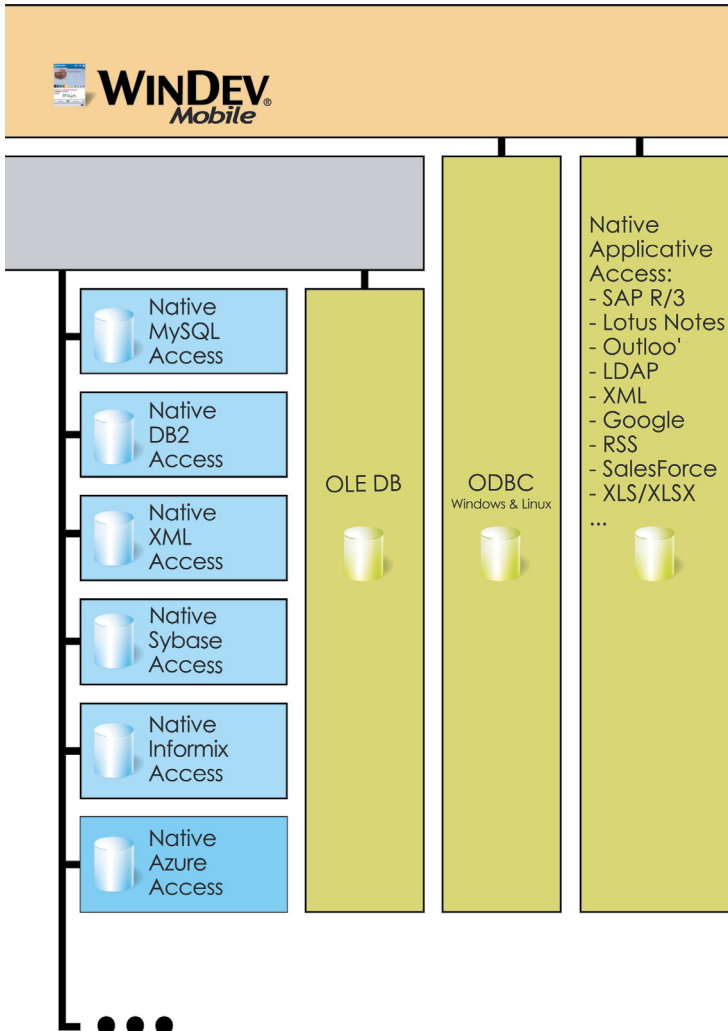
9.3 Resetting the version number to 1

The generation number of the analysis can be reset to "1" ("Analysis .. Reset the version number of the analysis to 1"). In this case, the version number of the analysis is reset to one. No specific action is performed on the data files.

The different types of accessible files

WinDev, WebDev and WinDev Mobile propose an easy access to most of the databases on the market.



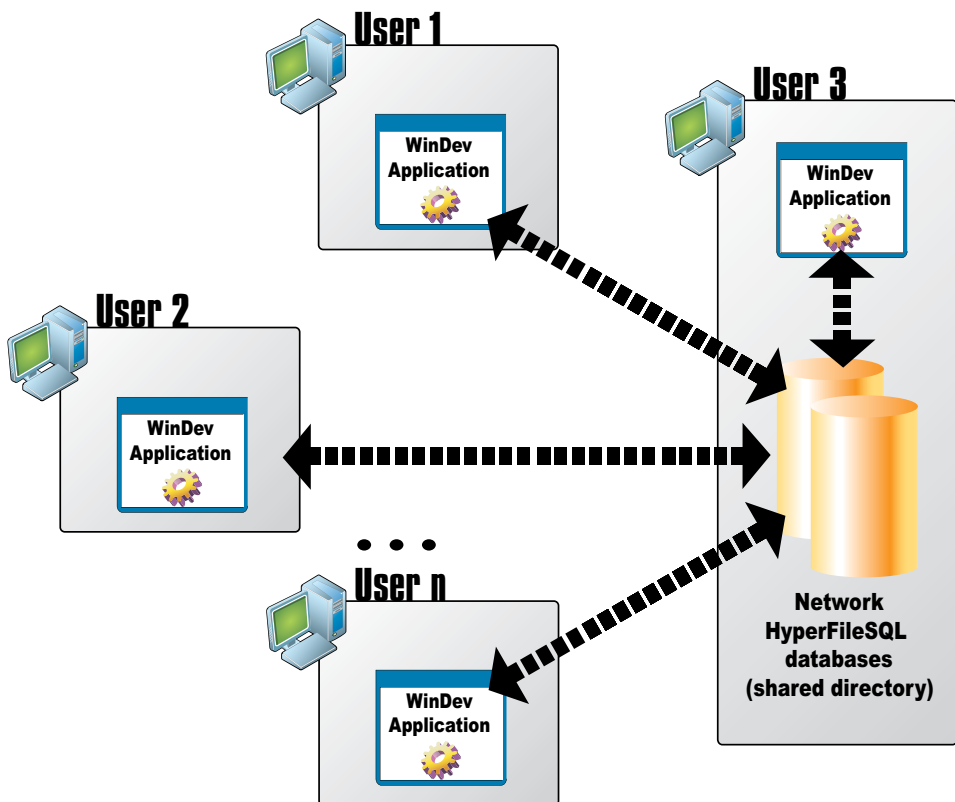


HyperFileSQL Classic

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

The characteristics of HyperFileSQL Classic are as follows:

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

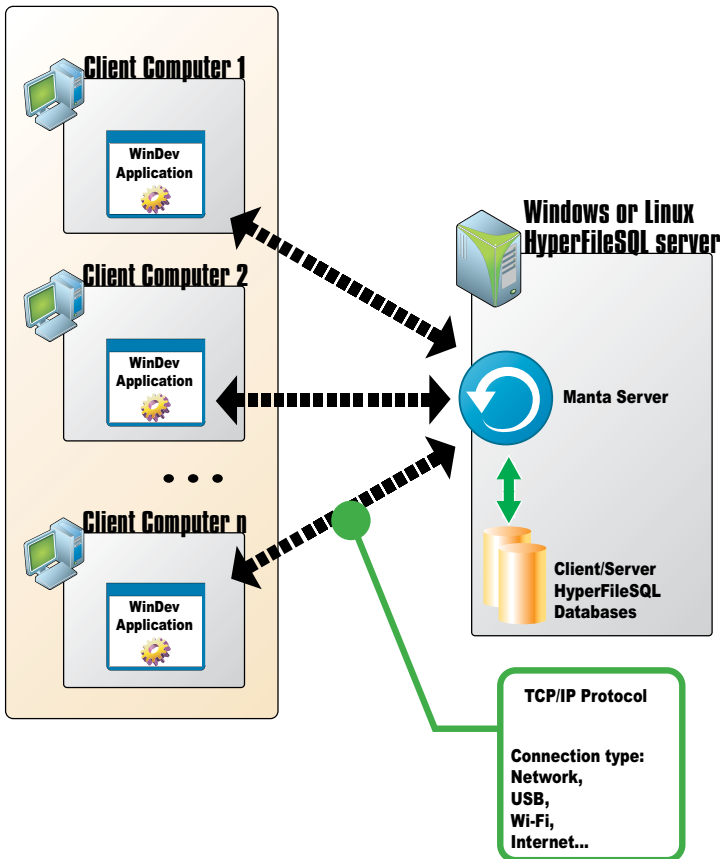


HyperFileSQL Client/Server

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

The characteristics of HyperFileSQL Client/Server are as follows:

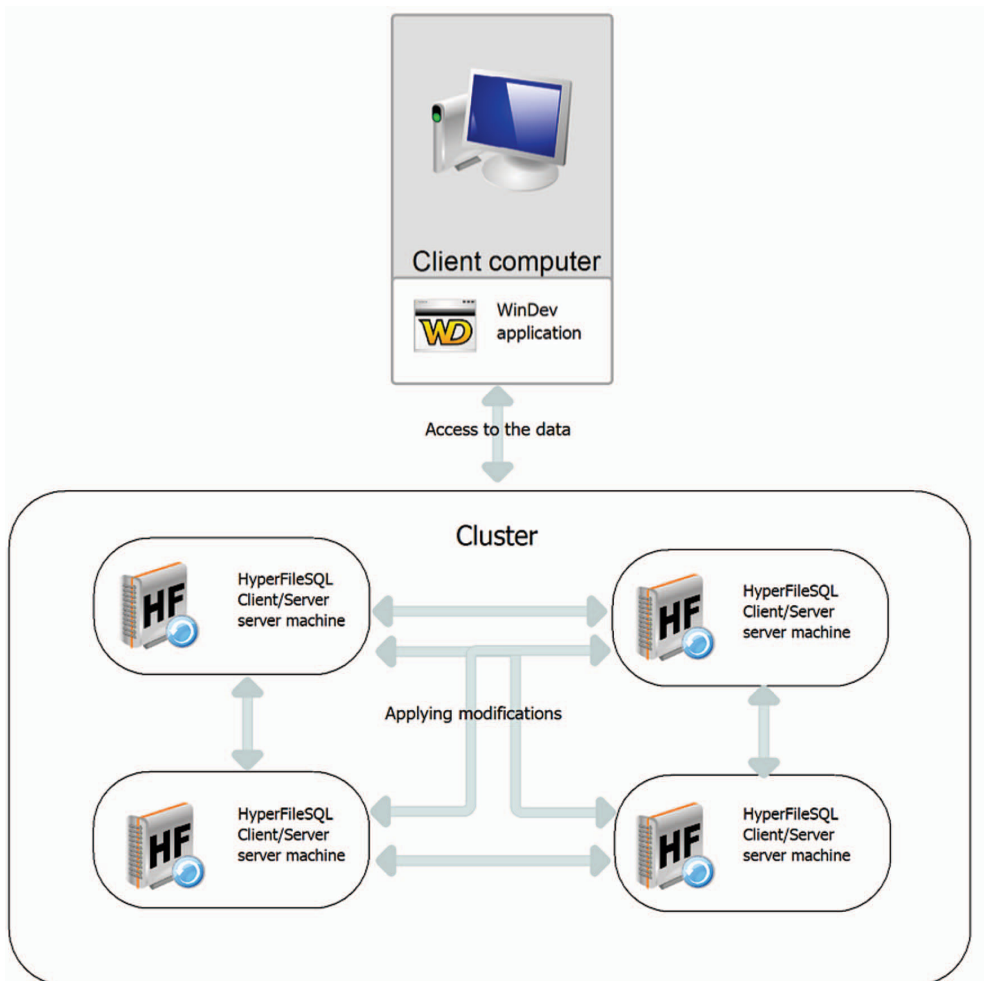
- A HyperFileSQL Client/Server application can be run on different user computers (called client computers) that communicate with the database server via the network.
- The data files are found on a server. Only the server physically accesses the data files.
- All the processes (query, read/write operation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



HyperFileSQL Cluster

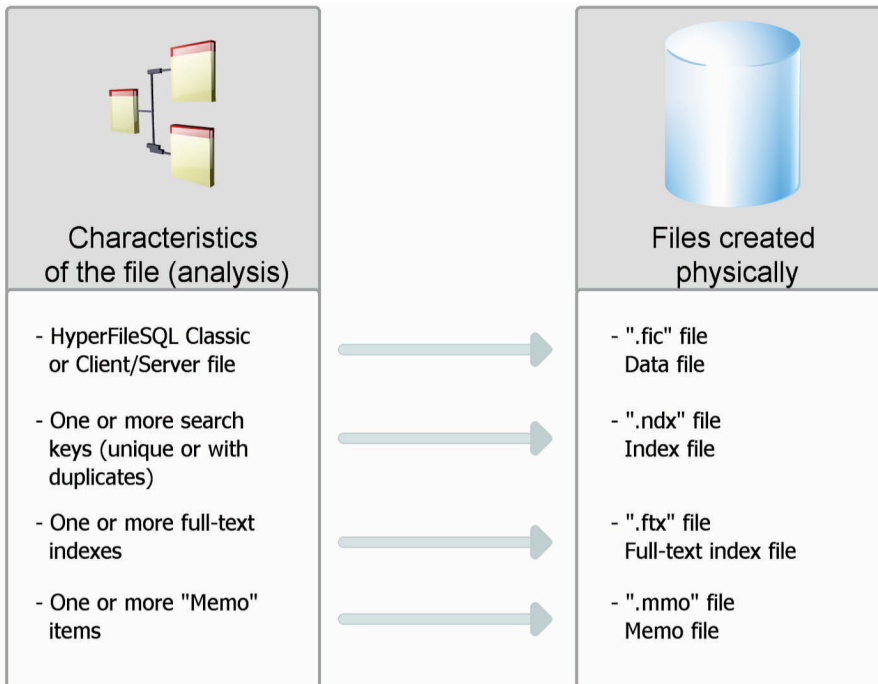
HyperFileSQL Cluster is an extension of the HyperFileSQL Client/Server database. In a database cluster, all the different HyperFileSQL servers contain a copy of the databases and they are synchronized in real time.

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



HyperFileSQL: the files physically created

The data model editor is used to describe the structure of the data files. Different files are physically created according to the information entered in the data model editor.



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

Associating controls and data

A window can display information coming from:

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

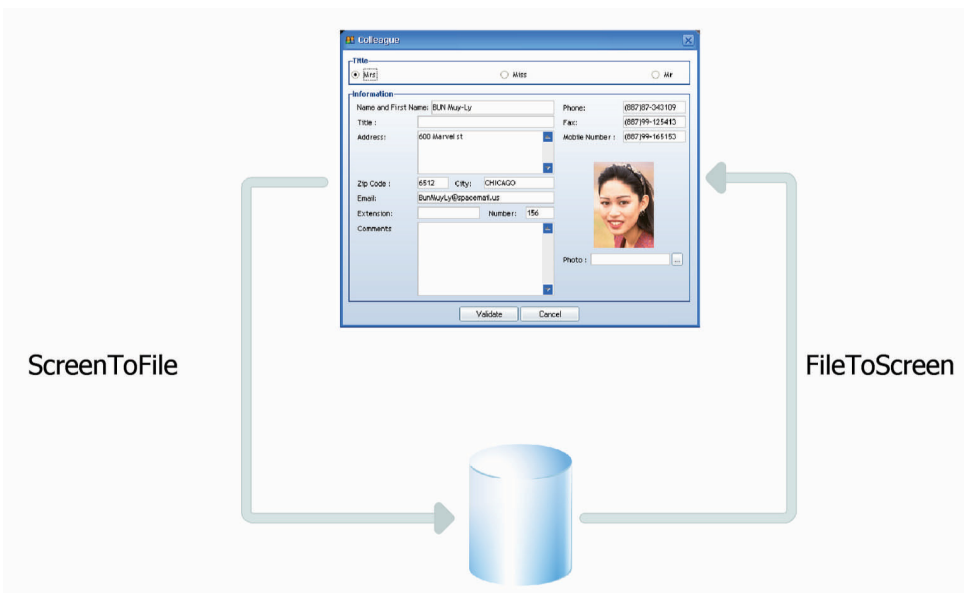
To display this information in a window, the controls of this window must be linked to:

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

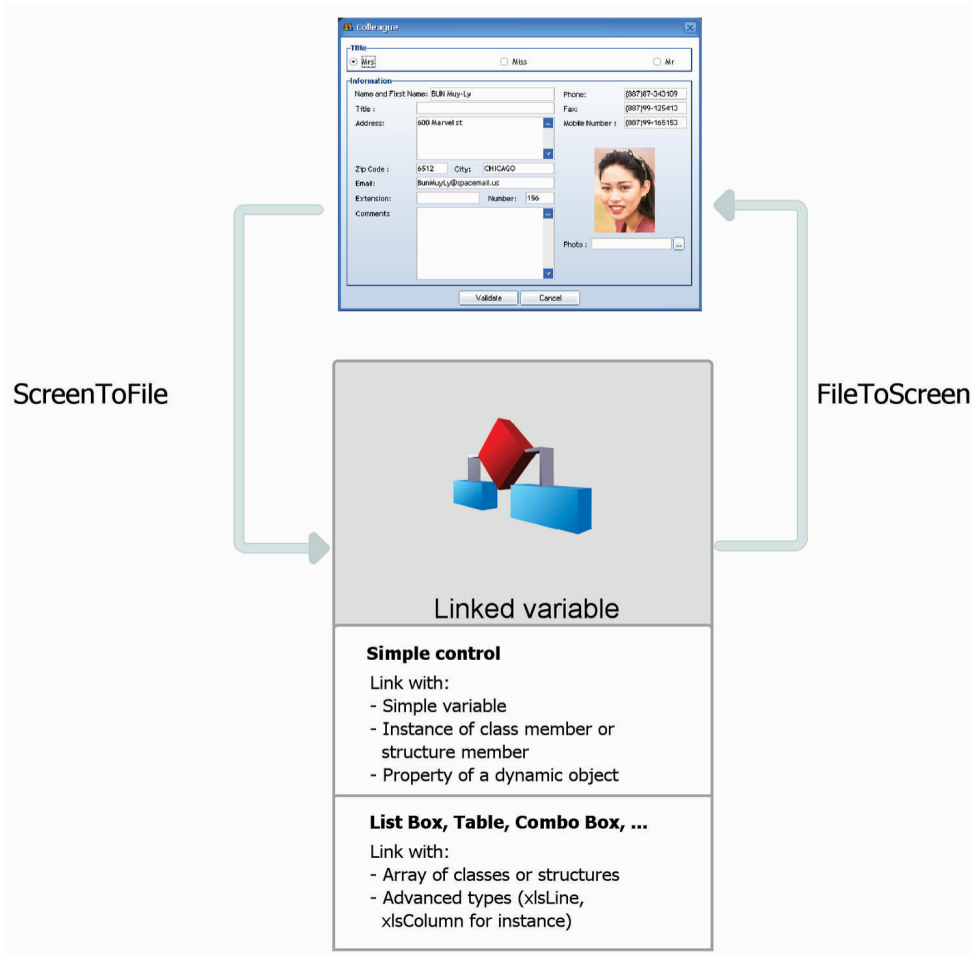
The method for displaying and retrieving the information is straightforward:

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

Link between control and item



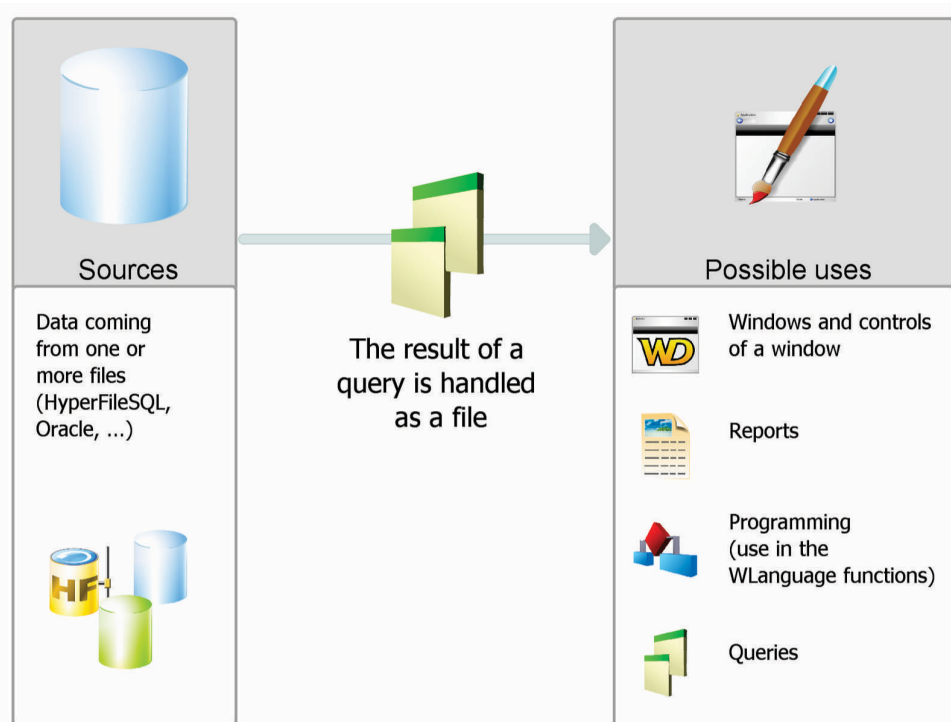
Link between control and variable



Queries

A query is used to interrogate a database in order to view, insert, modify or delete data. The structure of the query defines the data used. A query can interrogate one or more data files.

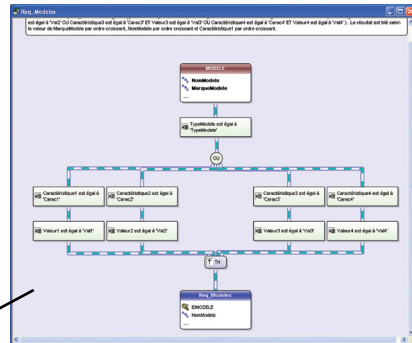
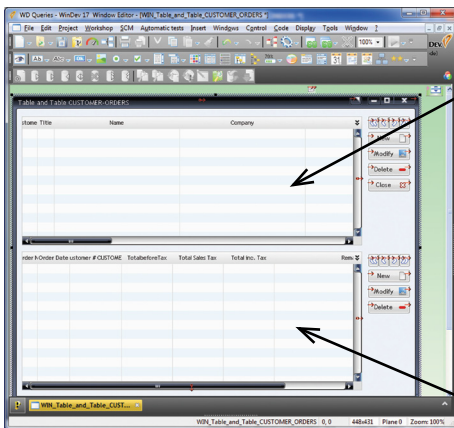
The query editor enables you to easily create queries without programming.



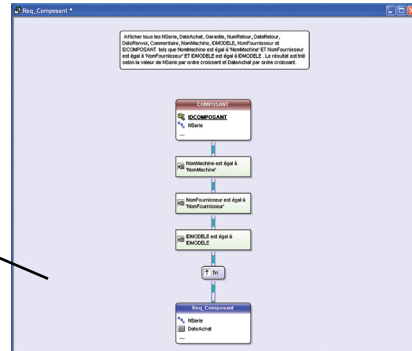
Note: In programming, a query can be handled like a data file. Especially, it can be associated with a display control (a table for example) that will present the data returned by the query.

Embedded queries

The controls found in a window can be linked to a data file or to an existing query. These controls can also be linked to a query created when the control was designed.



Embedded query:
MyWindow_1\$Query



Embedded query:
MyWindow_2\$Query

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

The Table/Looper controls

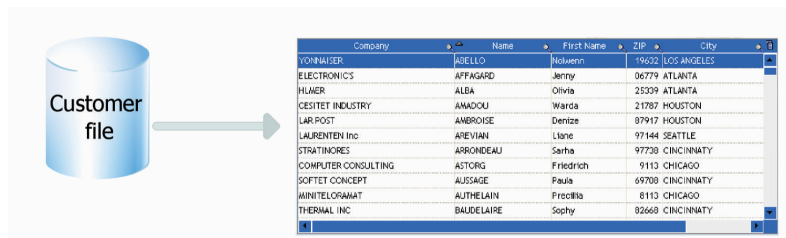
The Table/Looper controls can be used to display a set of information (the content of a data file for example). The content of these controls can come from three different sources:

- "Direct access file" table/looper
- "Memory" table/looper
- "File loaded in memory" table/looper

Note: These three fill modes will be presented in details in this page for the Table control. The same concepts apply to the Looper control.

"Direct access file" table

A browsing table with direct access is used to directly display the data coming from a data file, a query or an array variable. Reading the data file lets you display the data in the table. For each line displayed, the data file is read: the record read is displayed in a table row.

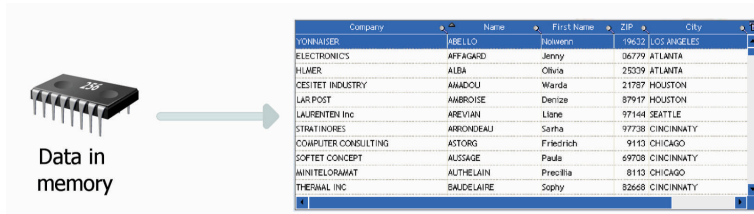


The data displayed that is not linked to the data file is not kept when refreshing the table display (when scrolling for example).

The WLanguage functions starting with "Table" are used to handle the browsing tables with direct access. Adding or deleting a row in or to the table triggers the addition or deletion of the record in the linked data file.

"Memory" table

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



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

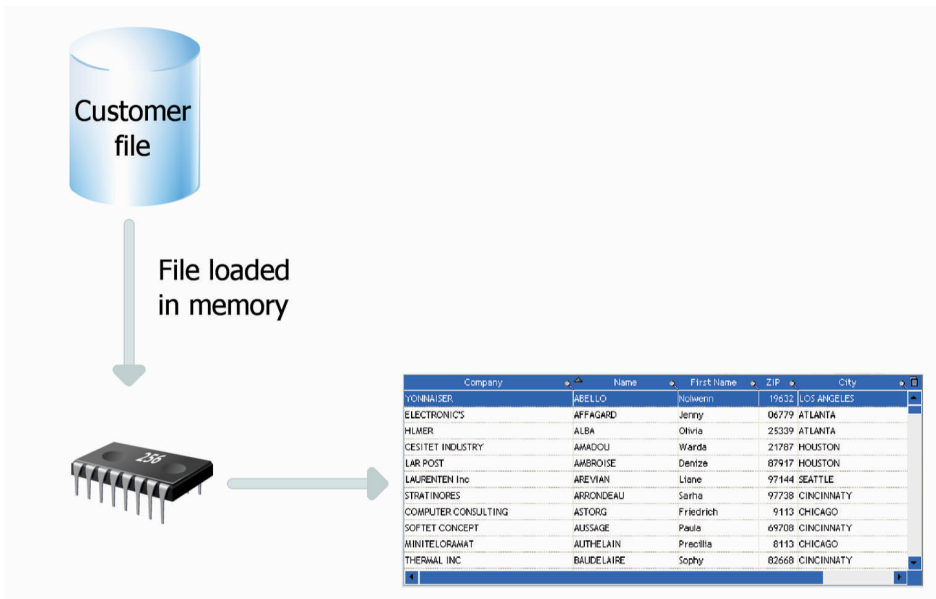
"File loaded in memory" table

The browsing tables loaded in memory combine the benefits of the browsing tables with direct access and the benefits of the memory tables.

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

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

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

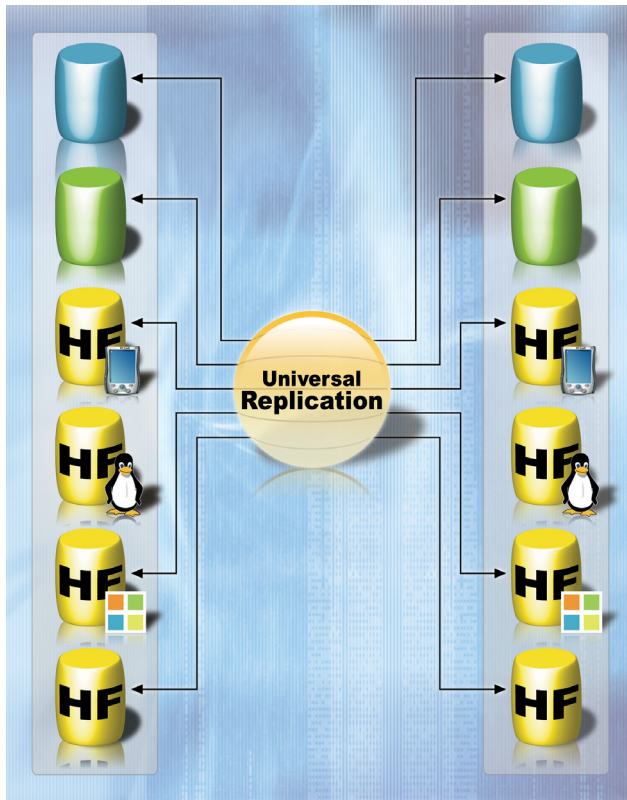


Note: the different fill modes are available for list boxes, combo boxes, tables, and loopers.

Universal replication

The universal replication is used to update databases of the same format or databases of different formats (HyperFileSQL, Oracle, SQL Server, ...). You can for example perform a synchronization between a HyperFileSQL database and an Oracle database.

The universal replication uses a centralized model: all the databases are synchronized with a master database. Then, the master database carries over the modifications to the other databases.



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

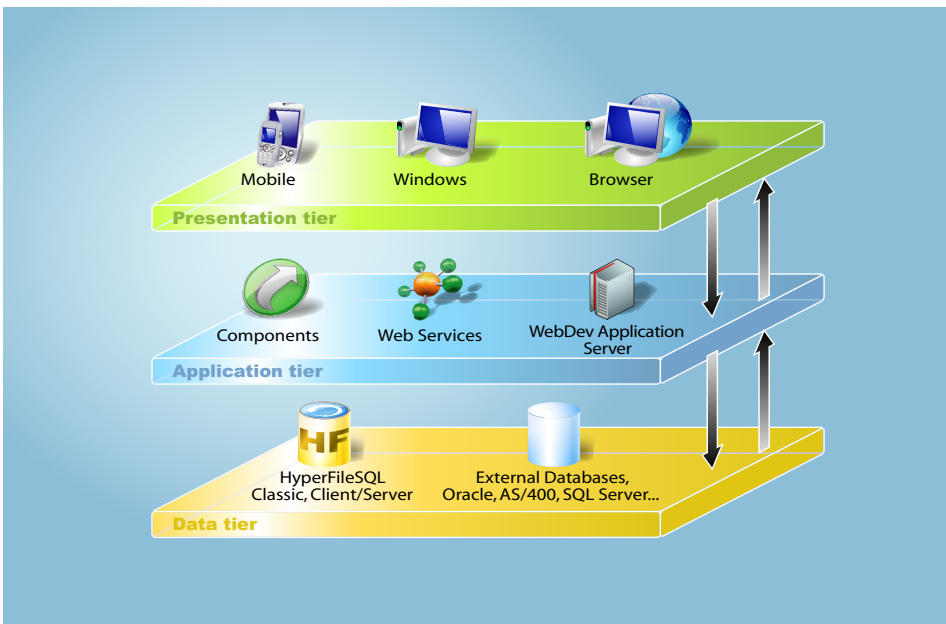
These adaptations must be done by programming by *HRplFilterProcedure*

3-tier architecture

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

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

The reason for separating the layers is to facilitate the maintenance and the future evolutions of the application. This provides better security because the access to the database is allowed via the process tier only. It also optimizes the teamwork and the multi-target development.



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



PART 4

Developing applications: advanced concepts



Express



DEVELOP 10 TIMES FASTER

PCSOFT

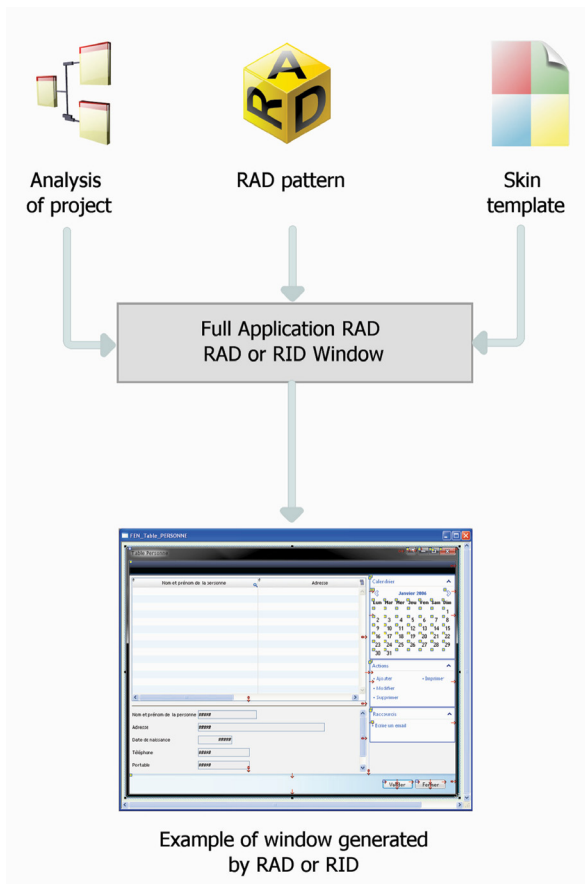
RAD RID

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

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

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

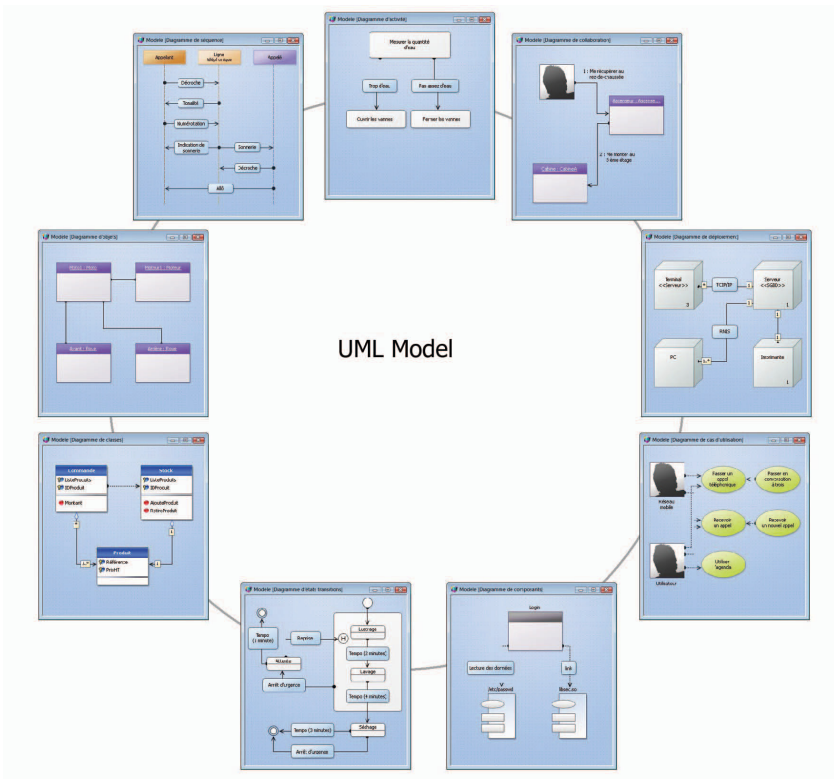
In RID generation, the generated windows only contain the controls linked to the analysis items. The code required for these windows to operate must be written by the developer. Only the code required for the additional pattern elements to operate is added. Your custom code can be entered directly.



The UML model

WinDev allows you to create nine types of UML models:

- **Class diagram:** describes the overall structure of a system.
- **Use case diagram:** represents the features of the system from the 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:** presents the chronological order of the messages sent and received by a set of objects.
- **Collaboration diagram:** represents the structural organization of the objects that send and receive messages.
- **State-transition diagram:** represents a sequence of states.
- **Deployment diagram:** describes the hardware (nodes) used in a system implementations and the execution environments deployed on the hardware.



The UML model in practice

1 Overview

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

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

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

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

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

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

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

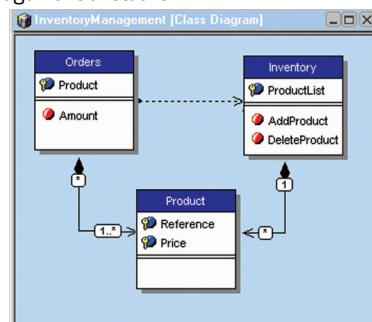
2 The different UML diagrams

2.1 Class diagram

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

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

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



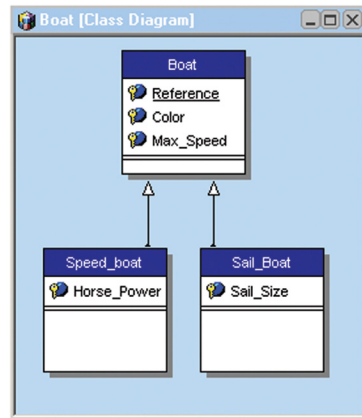
A class diagram contains the following elements:

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

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

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

- **Public:** the element is visible by all the other classes.
- **Protected:** the element is visible by the class itself and by the sub-classes.
- **Private:** the element is visible by the class only.
- **Relationship:** describes the behavior of the classes among themselves. Three types of relationships are available:
 - **Association:** Structural relationship between classes. For example, the *Orders* class is linked to the *Product* and *Customer* classes. A Customer can place several Orders. An order contains several products. An order must necessarily contain at least one product.
 - **Dependence:** Use relation establishing that a class's instances are linked to the instances of another element. For example, the *Orders* class uses the *Stock* class: before adding a product into an order, we must check whether this product is in stock.
 - **Generalization:** Relationship between a general class (parent) and a specific class (child) that derives from it. For example, the *Sail Boat* class and *Speed Boat* class are derived from the *Boat* class.



- **Package:** used to divide and organize the representation of the diagram (like a directory organizes files). Each package can contain classes and relationships.

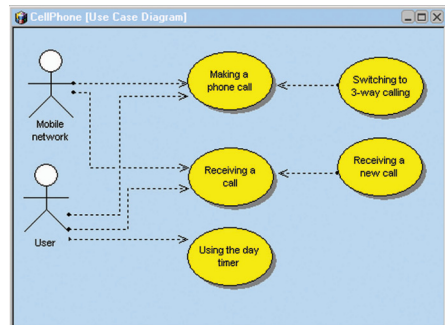
Via the generation of a class diagram, it is possible to create the structure of the WinDev classes used in your application.

2.2 Use case diagram

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

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

For example, you can describe the behavior of a cell phone with a use case diagram.



A use case diagram contains the following elements:

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

- **Use case:** describes a sequence of actions performed by the application. For example, *Placing an order, Entering an invoice, Creating a new customer form, ...*

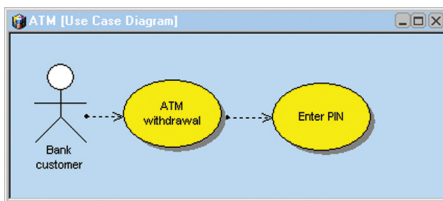
A use case describes what an application does but does not specify how the application does it.

- **Relationship:** describes the behavior of the actors in relation to the use cases. Three types of relationships are available:

- **Association:** Structural relationship between two linked elements.

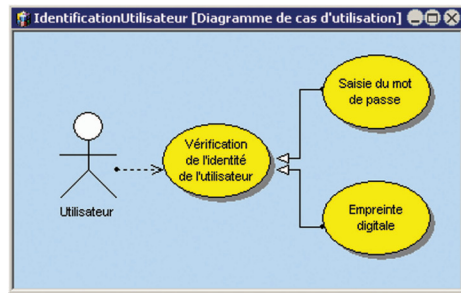
- **Dependency:** Relationship establishing that an element uses another one. For example, a bank *customer* may get *cash* from an ATM. In this case, the *Get Cash* action depends on the *Customer*.

In order to get cash, the Customer must enter his PIN number. In this case, the *Get Cash* action depends on the *Password Input*.



- **Generalization:** Relationship used to organize the elements according to a hierarchy. For example:

- two types of *Customer* actor are available: *Individual customer* or *Enterprise customer*
- the identity check can be performed according to two different methods: enter a password or check a fingerprint.



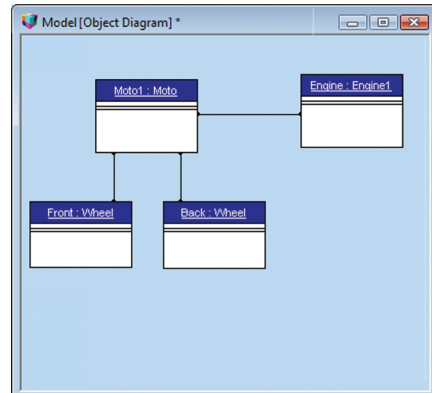
- **Package:** divides and organizes the representation of the diagram (like the directories organizes the files). Each package can contain actors and use cases.

2.3 Object diagram

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

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

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



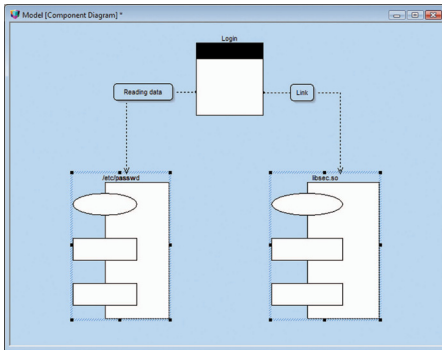
An object diagram contains the following elements:

- **object:** represents the instance of a class. If a class diagram is open, you have the ability to create an object from a class found in this diagram (drag and drop from the "UML analysis" pane).
- **composite object:** visually represents an object made of other objects. For example: a window containing scrollbars, buttons, ...
- **link:** represents the relationship between the various objects.

2.4 Component diagram

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

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



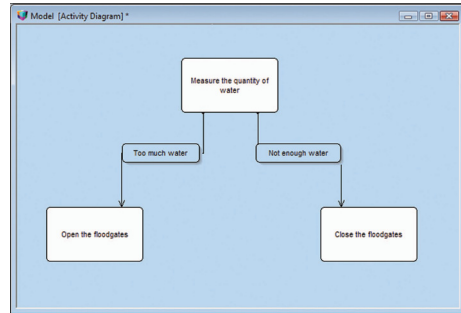
A component diagram contains the following elements:

- **module**: represents the different physical elements that constitute an application. For example: a file, a library, ... A module can be represented:
 - by a **specification** that shows the interface of the module. This specification can be **generic** for the customizable classes.
 - by its **body**, that presents the implementation of the module.
- **task**: represents a component having its own control thread.
- **main programs** of the application.
- **sub-programs**: groups the procedure and functions that do not belong to classes.

2.5 Activity diagram

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

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



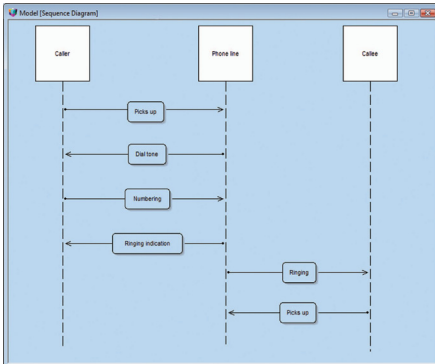
An activity diagram contains the following elements:

- **activity**: presents a specific step in the execution of a mechanism. For example: "Create an estimate", "Open a window", ...
- **synchronization bar**: used to synchronize the different activities:
 - by indicating the activities that must be performed before a given activity. For example: "Press clutch" and "Change gear" before "Release clutch".
 - by indicating the activities to perform in parallel.
- **object**: used to attach activities to the object that performs these activities. For example, the "Order" and "Pay" activities are attached to the "Customer" object; the "Teach" and "Check the knowledge" activities are attached to the "Teacher" object.
- **send signal**: represents the sending of a signal to an object.
- **wait for signal**: represents the wait for a signal coming from an object.
- **transition**: represents the passing from one finished activity to another. For example: "Too much water", "Not enough money", ...

2.6 Sequence diagram

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

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



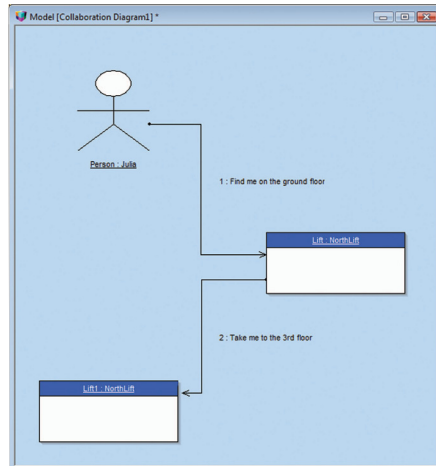
A sequence diagram contains the following elements:

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

2.7 Collaboration diagram

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

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



A collaboration diagram contains the following elements:

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

2.8 State-transition diagram

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

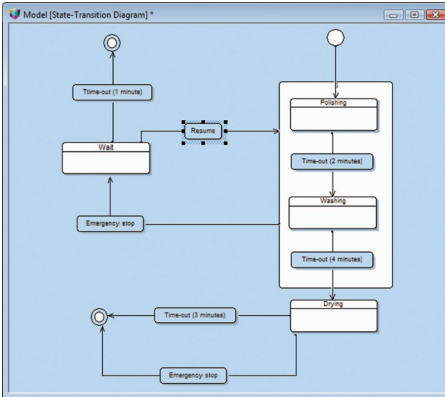
A state is defined by its duration and by its stability.

A transition represents the instantaneous change from one state to another one.

A transition is triggered:

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

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



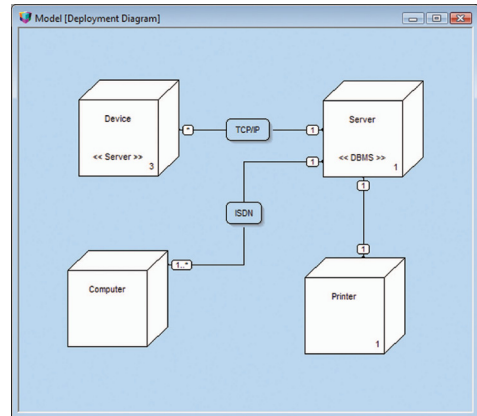
A state-transition diagram contains the following elements:

- **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.
- **super-state**: used to structure the diagram by indicating several distinct levels between states.
- **history**: represents the last active state of a super-state.
- **stub**: used to symbolize the states found in a super-state. This allows you to link these states to other states that do not belong to the super-state.
- **transition**: represents the passing from a state to another one.

2.9 Deployment diagram

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

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



A deployment diagram contains the following elements:

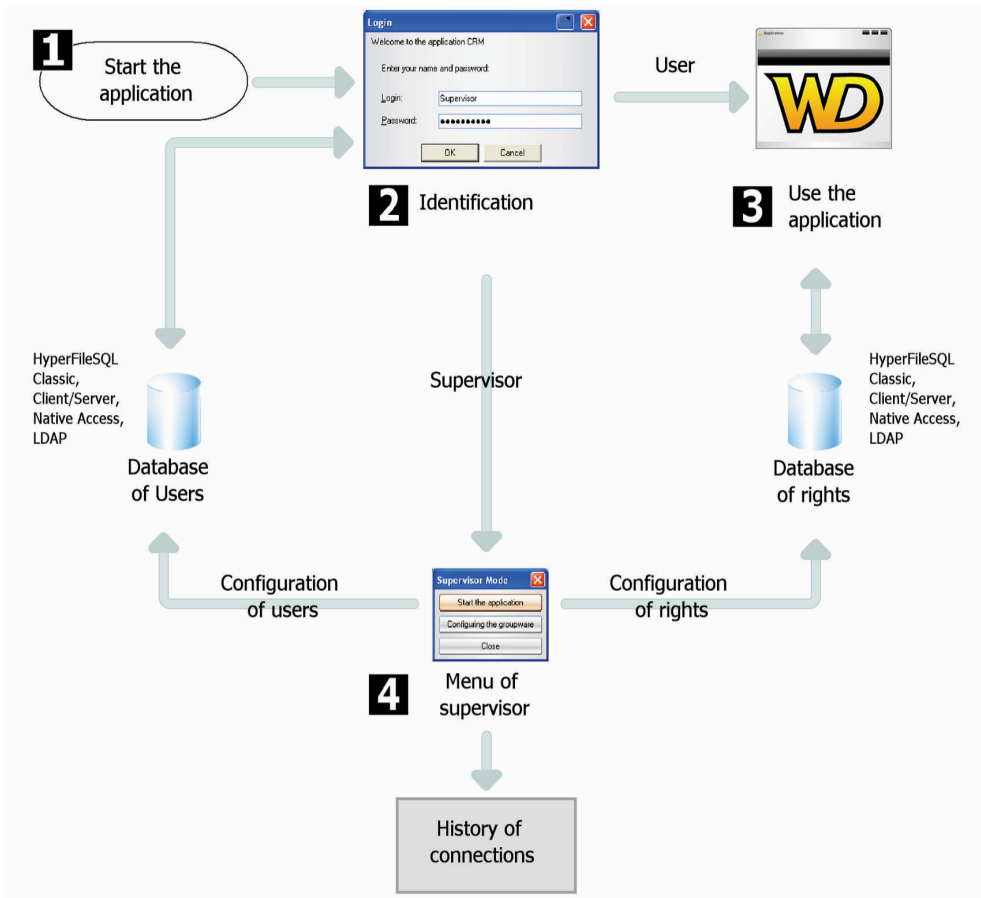
- **node class**: represents a class of hardware resources. For example: a server, a pc, a printer, ...
- **node instance**: represents a hardware resource. For example: server number 3, printer number 7, ...
- **connection**: describes the communication support between two nodes. For example: TCP/IP.

User groupware

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

Two types of users are taken into account:

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



The user groupware in practice

1 Overview

An application requires the definition of the role of the different contributors. It is often necessary to define several access levels according to the user.

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

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

- Seeing the price list
- Modifying the price list
- Entering orders
- Entering customers.

The accesses differ according to the user. Some examples:

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

To manage these access levels in your WinDev applications, all you have to do is include the user groupware in your application. In just a few mouse clicks, a standard application can be changed into an application that manages several access levels. This feature can be easily implemented.

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

2 How does the user groupware operate?

An application that uses the user groupware includes two levels:

- the user level
- the supervisor level

2.1 The user level

The user connects to the application via a login window and he can access the features that have been allowed for him.

Note: the user groupware enables you to connect by using an LDAP directory.

2.2 The supervisor level

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

The configuration window allows the supervisor to:

- start the application.
- configuring the groupware.

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

See "Configuring the application that uses the user groupware", page 132 for more details.

Note: The information about the users and their rights is stored in data files in HyperFileSQL format. The files regarding the users can be common to several applications. The data files containing the rights granted to the users on the different windows found in the application are specific to the application and they cannot be shared.

3 Implementing the user groupware

3.1 Adding the user groupware into an application

To implement the user groupware in a WinDev application, all you have to do is select: "Workshop .. Configure the user groupware". The window for configuring the user groupware is used to define:

- The management mode of the user groupware.
- The way to start the user groupware.
- The location of the data files of the user groupware.

Mode for including the User Groupware

Two methods can be used to include the user groupware:

- **Default integration:** the programming of the user groupware is automatically included in your application.

A new window is included in your project: the login window. It is displayed when the application is started and it allows the user to connect to the application. You can customize this window by applying the skin template of your application for example.

- **Custom integration:** the programming of the user groupware is automatically included in your application. **All the windows required to manage the user groupware (login and management of users) are automatically added to your project.** This enables you to customize all the windows used by the user groupware

See "The WinDev windows", page 134 for more details.

Starting the user groupware

Two modes are available for starting the user groupware:

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

Automatic login in test mode

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

LDAP directory

If the company of your customer uses an LDAP directory, you have the ability to point the user groupware to this directory. Therefore, the account and the password found in the LDAP directory will be automatically requested from the user.

Two modes are available:

- **Without rights management:** In this case, no groupware data file will be created. If the application is started in automatic mode, the login window will ask the user to identify himself. If the user is saved in the LDAP directory, the application will start; otherwise, it will be closed. A supervisor cannot configure the rights on the windows.
- **With rights management:** Only the users found in the LDAP directory will be able to connect. The rights can be configured on the windows.

See the online help for more details.

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

Data files of user groupware

Two categories of data files are managed by the user groupware:

- **the data files used to identify the users.** These data files can be common to several applications.
- **the data files used to manage the user rights** for the different windows of the application. These data files are specific to the application.

The password of the data files

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

To change this password, enter the new password. The characters typed are displayed as stars.

Note: This password is used for example when reindexing or when opening data files with the WDMAP tool.

See "The data files", page 134 for more details.

3.2 Running the test of the application

When running the test of an application that manages the user groupware, the first window displayed is the login window (regardless of the first window defined in your application).

By default, a single user exists: the supervisor. To login as the supervisor, enter the following information in the login screen:

- Name: SUPERVISOR
- Password: SUPERVISOR

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

Notes:

- If you do not want the first window of your application to be the login window, check "Manual start" in the options of the User Groupware. All you have to do is use the WLanguage function named *gpwOpen* to open the login window.
- The windows for managing the user groupware are run before the initialization code of the project.
- The window defined as the first window of your application in the project will be opened after the login window (if the user is not the supervisor).
- **If you do not want to start the user groupware while running the application tests**, uncheck "Workshop .. Configure the user groupware". If the user groupware is re-enabled later, the data files previously used by the user groupware will not be erased.

3.3 Creating the WinDev executable

When creating the executable that manages the user groupware ("Workshop .. Executable .. Generate the executable"), a screen specific to the user groupware is displayed.

This allows you to define the location of the common data files of the user groupware (files containing information about the users).

3.4 Installing an application that uses the user groupware

To install an application that uses the user groupware, select "Workshop .. Create the setup procedure".

A screen specific to the user groupware enables you to configure the location of the data files for the user groupware.

Notes:

- If the data files of the user groupware have been configured for your client, they must be selected when preparing the setup program. Therefore, you must customize the list of files to install.
- If the groupware data files have not been configured, only the Supervisor user will be able to login when the application is started for the first time.

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

4 Configuring the application that uses the user groupware

Once the application is deployed, the user groupware is configured by the application supervisor.

This setting consists in creating users and groups of users, and in granting them specific rights for each control found 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 user groupware, you must:

1. Start the application (executable or test from WinDev) and connect as supervisor:

- Name: SUPERVISOR
- Password: SUPERVISOR

2. Select "Configure the groupware".

4.1 Managing the users

To create a new user, you must specify:

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

The user can be defined as being a supervisor of the application.

A user can be modified or deleted. When deleting a user, you have the ability to delete the entire user or to delete his rights only.

The users can be organized in groups. When creating a group, you have the ability to copy the rights previously defined for another group or user.

A group can be created or deleted. When deleting a group, you have the ability to delete the group and its users or to delete the group only. In this case, the users are no longer associated with a group.

Notes:

- The supervisor password should be changed when it is first used.
- The "Default" group is the group proposed by default in the user groupware. By default, this group has all the rights on the application.
- The <None> group and the DEFAULT group cannot be deleted.
- If you are using an LDAP directory, you can import users found in the directory in order to manage the rights of these users.

4.2 Managing the rights

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

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

The elements managed in the windows, internal windows and window templates are as follows:

- the controls
- the groups of controls
- the menu options

The states available for each element are as follows:

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

5 Tips for developing an application that manages the user groupware

5.1 Using groups of controls

To simplify the configuration of the user groupware according to the users, the controls should be organized in groups of controls.

In your windows, you can create groups of controls corresponding to the controls that must be displayed for a type of user.

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

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

5.2 Visibility of controls

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

- when describing the element (7-tab window)
- by programming (**..State** or **..Visible** properties)

When configuring the user groupware, the supervisor can define different visibility characteristics. The characteristics defined by the supervisor have priority. For example, a button is used to make a control active. This control was grayed by the supervisor. Your code will be ignored and the control will not be active.

5.3 Defining the rights

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

- configure the rights for your application before you create the setup program. Then, all you have to do is add the groupware data files when creating the setup procedure of the application.

- supply a programming documentation that gives the names of the controls, groups of controls and options that must be configured according to the level of use defined for the application.

6 The elements of the user groupware

6.1 The WinDev windows

Depending on the integration mode of the user groupware in your application, one or more windows can be included in your project. These windows are as follows.

Note: the operating mode of these windows is presented in details in "Configuring the application that uses the user groupware", page 132.

Login window (GPWLOGIN.WDW)

This window allows all the application users to connect to the application. In this window, the user must enter the name and password (optional) that were previously defined by the application supervisor. During the startup, you have the ability to use the "SUPERVISOR" login and the "SUPERVISOR" password.

Window of supervisor menu (GPWSupervisorMenu.WDW)

This window is available in custom mode only.

This window allows the supervisor to either start the application, or to configure the rights.

User Form window(GPWUserForm.WDW)

This window is available in custom mode only.

This window allows the supervisor to enter a new user and his personal details.

Association of Users and Configurations window (GPWAssociationConfiguration.WDW)

This window is available in custom mode only.

This window allows the supervisor to manage the users, the groups and the rights.

Configuration Details window (GPWConfigurationDetails.WDW)

This window is available in custom mode only. This window allows the supervisor to configure for each window the rights granted to a user (or to a group) on all the controls, groups of controls and menu

options found in the window.

Configuration Choice window (GPWConfigurationChoice.WDW)

This window is available in custom mode only. This window allows the supervisor to copy a configuration that was previously defined (for a user or for a group).

Configuration Form window (GPWConfigurationForm.WDW)

This window is available in custom mode only.

This window allows the supervisor to create a user or a group.

Connection History window (GPWConnectionHistory.WDW)

This window is available in custom mode only.

This window allows the supervisor to view the connection history (and to purge this history if needed).

6.2 The data files

Two categories of data files are taken into account:

- The data files used to identify the users. These data files can be common to several applications.
- The data files used to manage the rights granted to the users on the different elements of the application. These data files are specific to the application and cannot be shared with another application.

By default, these data files are created in HyperFileSQL Classic format, but you can use HyperFileSQL Client/Server data files. See the online help for more details.

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

To change this password:

1. Display the window for configuring the user groupware ("Workshop .. Configure the user groupware").
2. In the "Files" tab, enter the new password. The characters typed are displayed as stars.

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

User data files

These data files are as follows:

GPWUser	File of users. Contains all the characteristics of the users (login, last name, first name, password, ...)
GPWUserConfiguration	File of groups of users
GPWConnection-Log	File containing the history of connections. This file is found only if the history of connections was requested.

The items of these data files are presented in details in the online help.

Where are these data files created?

- By default, these data files (HyperFileSQL Classic mode) are created in a subdirectory of your project (EXE\GPW_<Project Name>). When including the user groupware, a different directory can be specified to run the test of your application. When creating a WinDev executable, you can specify a directory for the common data files.
- When creating the setup procedure, you can allow the user to configure the location of these data files ("Additional modules" plane). Therefore, a common directory can be selected for several applications that manage the user groupware.
- If "Manual start" was selected, **gpwOpen** is used to start the login window of the user groupware and to customize the directory for the data files of the user groupware. Regardless of the directories specified when creating and installing the application, this directory will be taken into account if it is specified.

Note: if no specific configuration is performed, the user data files (HyperFileSQL Classic mode) will be created in a sub-directory of the setup directory of the executable named \GPW_<Project Name>.

Data files of rights

These data files are as follows:

GPWConfiguration	File containing the description of the available configurations. Each configuration is associated with a group of users.
GPWElementConfiguration	Status of each application element for each configuration. For example, allows you to store whether the "Statistics menu" must be grayed for the "Assistant" user.
GPWElement	File containing the different elements of the application to configure (window controls, menu options, ...).

The items of these data files are presented in details in the online help.

Where are these data files created?

- By default, these data files (HyperFileSQL Classic mode) are created in a subdirectory of your project (EXE\GPW_<Project Name>).
- When creating the setup procedure, you can allow the user to configure the location of these data files ("Additional modules" plane). This enables you to select a specific directory for the data files containing the application rights.
- If "Manual start" was selected, **gpwOpen** is used to start the login window of the user groupware and to customize the directory for the data files of the user groupware. Regardless of the directories specified when creating and installing the application, this directory will be taken into account if it is specified.

Note: if no specific configuration is performed, the data files for the rights will be created in a sub-directory of the setup directory of the executable named \GPW_<Project Name>.

7 Modifying the user groupware

7.1 Customizing the groupware windows

The login window

You may want to customize the login window that is displayed when your application is started. If this window is included in your project, the style or the skin templates used by your application can be applied to this window.

Caution: this window must not be renamed.

All the windows of the user groupware

If all the windows of the user groupware have been included in your project, they can all be customized. However, only the login window will be seen by all the users. The groupware configuration windows are only seen by the application's supervisor.

Caution: The GPWLogin and GPWSupervisorMenu windows must not be renamed.

7.2 User groupware and multilingual application

If your application is a multilingual application, this feature must also be managed in the windows of the user groupware. By default, the windows of the user groupware support English and French.

To use the user groupware in a multilingual application, you must:

1. Configuring the user groupware as follows:
 - all the windows of the user groupware must be included in the application.
 - the "Manual start" option must be selected.
2. Translate the different windows of the user groupware. By default, the windows of the user groupware are available in English and in French. To translate these windows into other languages, all you have to do is translate the different captions, ... (by using the WDMMSG tool if needed to extract the messages to translate).

Reminder: WDMMSG is an optional WINDEV tool used for extracting and re-integrating all the messages to translate.

3. Running the application in the requested language. To do so, you can:

- If necessary, ask the user for the language in which the application must be run (via a specific window for example).
- Configure the runtime language of the application with **Nation**.
- Open the first window for managing the user groupware with **gpwOpen**.

```
// Initialization code of the
// project
nLanguage is int =
    Open(Select_Language)
// Configure the language
Nation(nLanguage)
// Start the first window
// of the user groupware
gpwOpen()
```

See "Multilingual applications", page 139 for more details.

7.3 Modify the first window of the application according to the user

To change the first window of the application according to the user, use **gpwGetUserInfo** in the initialization code of your project. This option enables you to get information about the user such as his name or password for example.

Depending on his name, you have the ability to open a given window.

Example:

```
IF gpwGetUserInfo(...
    gpwLogin) = "FP" THEN
    Open(Win_Management)
ELSE
    Open(Menu)
END
```

7.4 Starting the login window from a window of the application

When the user groupware is included in an application, you can choose whether the user groupware will be automatically started or not. This option can be modified at any time.

By default (automatic start), the login window is the first window of the application. It is run with the initialization code of the project.

In some cases, you may want to open a configuration window before the login window. To do so you must:

1. Select "Manual start".
2. open the login window anywhere in your code with `gpwOpen`

7.5 Defining the rights on the menus

To define the rights on the popup menus of a WinDev application, all the windows of user groupware must be included. Then, you can modify the window for managing the rights (`GPWConfigurationDetails.WDW`).

To get the list of popup menus, use `EnumSubElement` (associated with the `enumFirst + enumContextualMenu` constants).

Special case: the menu "?" of a WinDev application:

Some options of the automatic menu must be used with great care. These options are as follows:

- Optimize the access speed to the database.
- Save the data.
- Save the application.
- Repair the database.

The operating mode of these options with the user groupware is as follows:

- By default, these options are grayed if the user is not the supervisor.
- To allow a non-supervisor user to access these options, modify the rights of the internal window named `IW_MenuHWinDevHelp` that contains the popup menu.

7.6 Configuring the data files of the user groupware

Two types of data files are used by the user groupware:

- data files of users.
- data files of rights.

The location of these data files can be configured:

- when including the user groupware for the tests.
- when creating the executable.
- when installing the application (by the person that performs the setup).
- by programming.

See "The data files", page 134 for more details.

7.7 Reinitializing the data files of the user groupware

To reinitialize the data files of the user groupware, you must delete the data files specific to the user groupware (file of users and file of rights). These data files are automatically re-created when the application is started.

Note:

- If only the user data files are deleted (`GPWUserConfiguration` and `GPWUser`), only the supervisor will be able to start the application.
- If only the data files of rights are deleted (`GPWElementConfiguration`, `GPWConfiguration` and `GPWElement`), all the users will have all the rights on the entire application.

See "The data files", page 134 for more details.

7.8 Modifying the data files used by the user groupware

The structure of the data files specific to the user groupware cannot be modified.

Note: These data files contain confidential information and therefore they are encrypted.

7.9 Managing the analyses

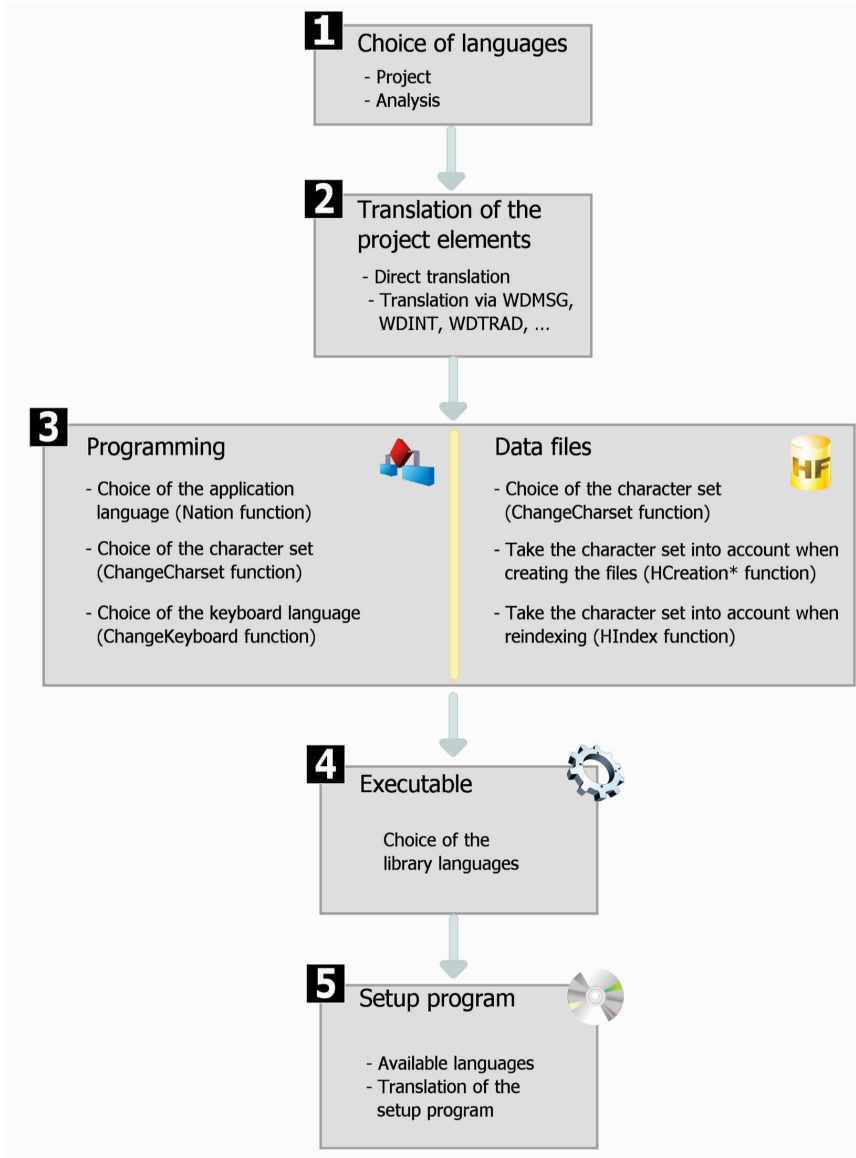
If a specific analysis is open while the user groupware is being configured, you need to re-run the `gpwlnitAnalysis` function (used to open the user groupware analysis).

Note: At run time, several analyses cannot be opened at the same time.

Multilingual applications

A multilingual application is an application that can be distributed in several languages. WinDev takes into account the different languages of the application at all steps of the application development.

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



Multilingual applications in practice

1 Overview

A multilingual application is an application that can be distributed in several languages. The different 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. Entering the different project elements (window, code, ...) in the different languages of the project.
3. Defining the project language by programming.
4. Managing specific character sets in the data files.
5. Creating the executable and the setup program.

Notes:

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

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 ("Project .. Project description .. Languages").

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

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

- for all the new elements or objects created in the WinDev editor,
- for any element or object opened in the WinDev editor.

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

Linguistic options

The display of the numeric data found in the project (number, currency, date, ...) can be configured in each language supported by the project.

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

This setting is performed in the "Languages" tab of the project description ("Project .. Project description").

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

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

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

Executable program

When creating the executable program, you have the ability to specify the languages that will be taken into account by the executable. You can for instance create a different executable program for each language supported by the project.

This feature is used to reduce the size of the executable.

2.2 Languages supported by the analysis

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

The configuration of the languages supported by the analysis is performed via "Analysis .. Analysis description .. International".

The different languages configured in the analysis will be proposed:

- when configuring the shared information of items. The description of the controls linked to the items (options, captions, ...) can be entered in the different languages supported by the analysis. When generating a "full application RAD" or a window RAD, this information is automatically taken into account by all the languages common to the analysis and the project.

- for the information printed in the program documentation (notes of the data file or item).
- for the information managed by "Reports and Queries". Indeed, if "Reports and Queries" is supplied with your WinDev application, you have the ability to translate the name of the data file and the name of each item. Therefore, the user will be able to use the names corresponding to his own language when creating his reports and queries. This information can be entered in the "Reports and Queries" tab of the description of the data file and items.

2.3 Languages supported by the different project elements

By default, the different project elements (windows, reports, code, classes, set of procedures, ...) support the same languages as the project in which they have been created.

An element can support more languages than the project (if the element is shared among several projects that support different languages for example).

3 Translating the interface into several languages

Once the different languages supported by the application have been selected, the information displayed by the application must be translated into these different languages.

To translate the interface into several languages, several elements must be taken into account:

- The selected translation mode.
- The supported languages (languages that use a special character set or not).
- The use of specific windows (user groupware, automatic management of HyperFileSQL errors, print preview).
- The messages displayed by programming.

3.1 Selected translation mode

WinDev proposes several translation modes:

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

Automatic translation of the interfaces

For each object, different multilingual areas are displayed in the description windows of the object. The multilingual areas enable you to enter the information in the different languages supported by the project.

To translate this information from WinDev:

1. Configure the translation options of the software used ("Tools .. Options .. General options of WinDev .. Translation"). These options enable you to define:

- the default translation tool (and the transmission mode for the text to translate and for the translated text),
- the source and target languages.

2. If a translation tool is specified, a "Translate" button will be displayed in each element containing information to translate. This button enables you to translate the selected text via the specified translation tool.

3. To perform the translation, all you have to do is select the text to translate and click the "Translate" button.

Checking out and checking in the information to translate

WDMMSG is used to extract all the project messages (captions of controls, ...) and check them back in 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 special character set

If your application supports languages that use specific character sets (Greek, Russian, ...), you must enter the translation of the different messages by using these character sets.

WinDev enables you to automatically manage the use of specific character sets in the editor.

Indeed, as soon as the edit cursor of the mouse is located in an input area of a language that uses a special character set, the corresponding input language (character set used by the keyboard) is automatically displayed.

Therefore, if a caption is entered in the Russian section of the caption description, the keyboard will automatically switch to the Russian character set.

Reminder: To use specific character sets, the files corresponding to the requested character sets must be installed in the regional options of Windows (control panel).

3.3 Translating specific windows

Managing the HyperFileSQL errors

By default, the windows for managing the HyperFileSQL errors are provided in English and in French. To translate them into another language, you must:

1. Include the default error windows in your project. The windows are provided (with their WLanguage code) as examples in the \Programs\Data\Preset Windows\HyperFileSQL - Automatic Help Windows.
2. Customize the management of errors to use the HyperFileSQL pages for error management (**HOnError**).
3. Translate the messages (see the previous paragraph).

Automatic management of errors

By default, the windows for automatic error management are supplied in English and in French. To translate them into another language, you must:

1. Include the default error windows in your application. These windows are supplied (as well as their WLanguage code) as examples in the following directory: \Programs\Data\Preset windows\Assertion - Error - Exception.
2. Translate the messages (see the previous paragraph).

User groupware

By default, the user groupware is supplied in English and in French.

To translate an application that uses user groupware, you must choose the "Custom" mode when the user groupware is included in your application ("Workshop .. Configure the user groupware").

The different windows of the user groupware will be included in your application.

Then, all you have to do is translate these windows.

Note: Uncheck "Auto run" if necessary: this enables you to display a window for language selection before the login window.

Print preview

By default, the preview window is available in English and in French.

To translate the preview window, the windows corresponding to the print preview must be included in your application. These windows are supplied (along with their WLanguage code) in \Programs\Data\Preset windows\Preview. See the online help for more details.

Then, all you have to do is translate these windows.

System menu

A default popup menu is available for the edit controls and for the Table controls. This menu allows the user to easily perform several operations.

To translate these menus, you must own WDINT.

3.4 Translating the messages found in the WLanguage code

In your code, several WLanguage functions allow you to communicate with the user via character strings. These messages must also be translated into the different languages of your application.

To translate a character string entered in the code editor:

1. Select the character string to translate.
2. Select "Translate" from the popup menu of the selection. The window for translating messages is displayed.
3. Enter the translation and validate. The code line

is displayed in the code editor:

Click on **Button1** *
 Info ("WinDev is fantastic"³)

The flag indicates that translations exist for this character string. Click this flag to display the translation window.

The number (2 in our example) indicates the number of translations entered for this character string.

Note: To transform all the messages found in your code into multilingual messages, select "Code .. Multilingual messages .. Convert simple strings into multilingual messages".

4 Choosing the language by programming

A multilingual application is an application that can be distributed in several languages. The user will be able to choose the runtime language of the application. You can for example:

- ask for the runtime language during the first startup of the application.

- implement an option (menu option or button for example) allowing the user to change the current language for the application.

5 Managing specific character sets in the HyperFileSQL data files

When an application is used to display or save data in HyperFileSQL data files, this data is entered in the current language.

No specific management is required if the application is used in a country that uses a Latin character set.

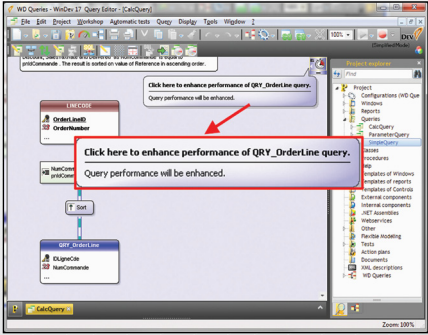
If the application is used in a country that uses a specific character set, this character set must also be supported by the data files. Therefore, the searches, the sorts, ... will be performed according to the selected character set.

Optimizing an application

WinDev proposes several methods for optimizing your applications:

- **The Aided Application Development (AAD)**, that suggests code or interface improvements when developing the application.

Developing the application

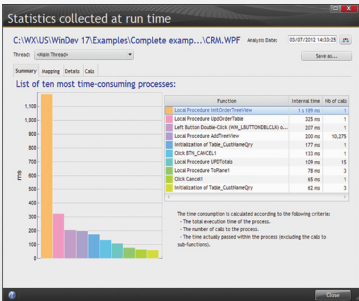


Optimizing your queries, improving the anchors in your windows, ...

These suggestions are made on a regular basis while you develop your applications.

- **The performance profiler**, which is used to quickly view time-hog processes. Then, all you have to do is optimize the requested process.

Test of the application



Analyzing the performance of the application while a test is running, viewing the longest processes to optimize the source code.

To start the performance profiler, select "Project .. Performance profiler .. Analyze the performance".

- **The static audit of the project**, which flags potential programming problems (empty groups, orphan elements, etc.) as well as large files that can take a very long time to load.

Static audit

The screenshot shows the 'Static audit' window with a 'Description' tab. It lists several issues under 'Potential dangers' and 'Quality'. The 'Potential dangers' section includes: 'The project is consequent and the sources are not managed by the SCM', '1 element not found on the disk at 1 component without source path', '6 errors, 15 warnings and 22 information messages', and 'The analyzer has detected that 7 queries can be optimized'. The 'Quality' section includes: '10 unused orphan elements have been detected in the project' and 'The analyzer has detected that 7 queries can be optimized'.

Audit report signaling the problems found, analysis of the library (to detect the large files) and analysis of the setup pack.

To start the static audit, select "Project .. Edition audit .. General report".

- **The dynamic audit of the project**, which will detect the loss of resource, the unprocessed errors and the slowness of the application in its production environment.

Dynamic audit

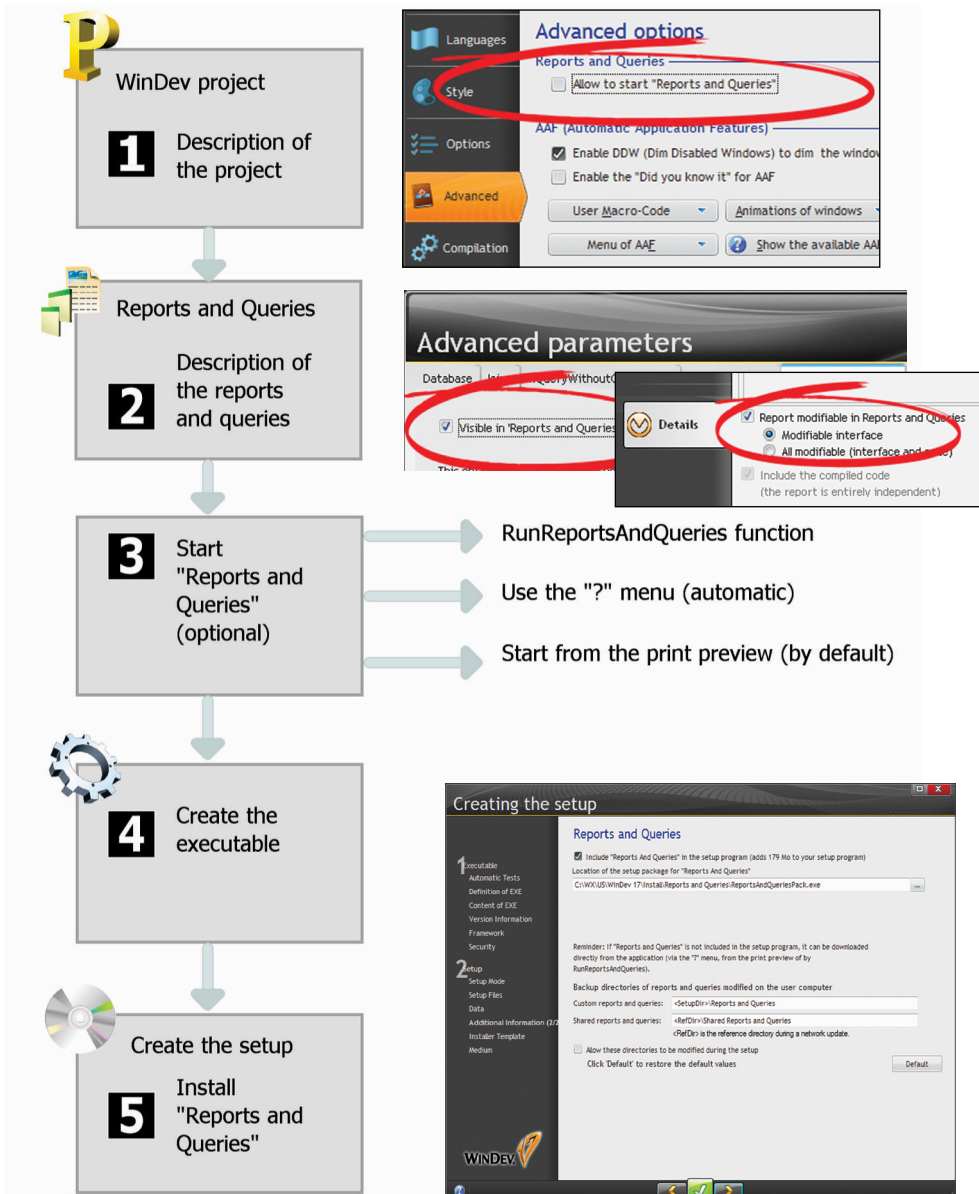
The screenshot shows the 'Dynamic audit' window with a list of detected errors and warnings. The list includes various messages such as 'The project is consequent and the sources are not managed by the SCM', '1 element not found on the disk at 1 component without source path', '6 errors, 15 warnings and 22 information messages', and 'The analyzer has detected that 7 queries can be optimized'.

Analyzing the performance of the application during a test, viewing the processes containing exceptions or errors not processed properly.

To start the dynamic audit, select "Project .. Test mode .. Debug the project while the audit is enabled".

Distributing "Reports and Queries"

With "Reports and Queries", which can be freely distributed with your applications, the users will be able to customize the reports according to their own requirements. Only the selected reports will be customizable.

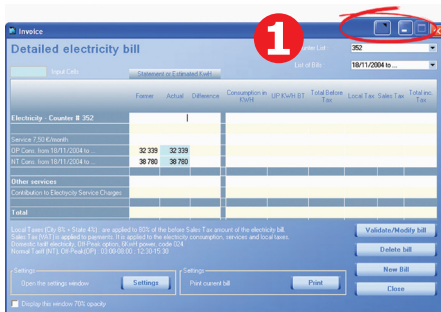


UMC: User Macro Code

The User Macro Code (UMC) allows end users to enter their own WLanguage code and to include it in an existing application.

Let's consider the following case:

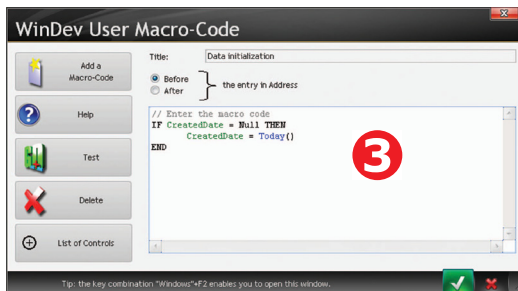
A window for entering a customer form requires the presence of an email address but it does not check its format. The user wants to check the presence of the "@" character in the address.



An icon found at the top of the window allows the user to customize the processes.



The 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 enters the WLanguage code. The F1 key is used to display the help for WLanguage.

To enable this feature (which is disabled by default), simply specify that the project supports UMC:

- in the project description,
- when creating the executable.

User Macro-Code: UMC in practice

1 What is the UMC?

UMC is an option allowing advanced users to create their own processes in the application. The end user can enter WLanguage code and add it to an existing application.

The added processes can be forwarded to the developer so that they can be added into the source code of the application.

Several WLanguage features are accessible to the user and restrictions can be defined by the developer. For example, forbid the use of the functions for file management. To insure the integrity of data, the commands for modifying or adding records are not available to the user.

2 How do I enable/disable UMC?

This feature can be disabled for a window or for the entire application.

To disable UMC for a window, check "Forbid the use of U.M.C." in the "Details" tab of the window description.

When creating the executable, you can allow the entry of User-Macro Code or forbid it for the entire application. If you authorize it, two options are available:



- Allow the use of HyperFileSQL commands (HReadFirst, HFilter, HAdd, ...).

- Accept the suggestions made by the users.

This second option enables the application to send an email when the user creates a macro code. The "*.UMC" file corresponding to the macro code is sent in attachment. This way you can easily get customer suggestions and automatically include them in your project ! Indeed, WinDev includes an option for importing the "*.UMC" files ("File .. Import .. User Macro-Codes ...").

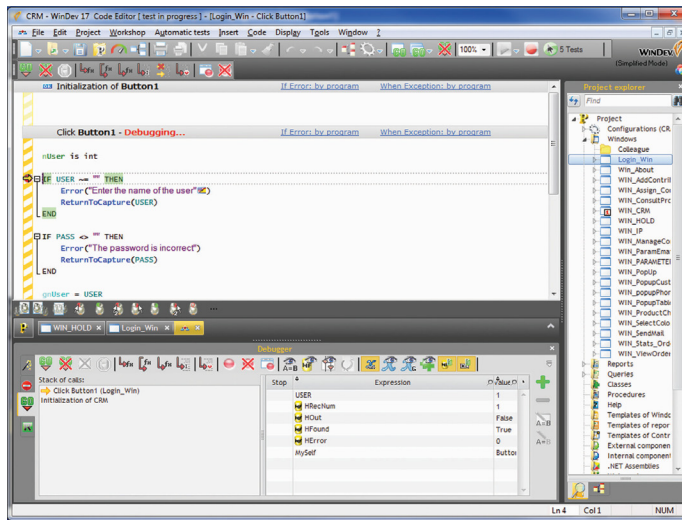
The different available tests

Several methods are available to run the test of your application. The table below presents these different modes.

Type of test	Description
Test of the entire project 	Simulates the start of the application by the executable. Used to run the test of the entire application, even if its development is not completed. As soon as a problem occurs in the execution of the project, the debugger can be started to identify and fix the problem.
Test of the current element 	Used to run the test of this element as soon as its development is completed. The debugger can be run as soon as a problem occurs in the reports or windows.
Test of a project step by step (via the debugger)	Used to run the debugger when the application is started. This solution enables you to monitor how the application runs via the debugger. You also have the ability to debug a specific executable, an already running executable or an executable running on another computer.
Unit test	Used to run the test of windows, procedures and classes found in an application at different levels of development. The test scenarios are automatically generated in WLanguage from the test of the application or from the test of an element. The scenarios can be modified in the code editor.
Non-regression test	Used to test the non-regression of an element between the different versions of an application. To do so, select (or create) the test scenario that will be run.

The debugger

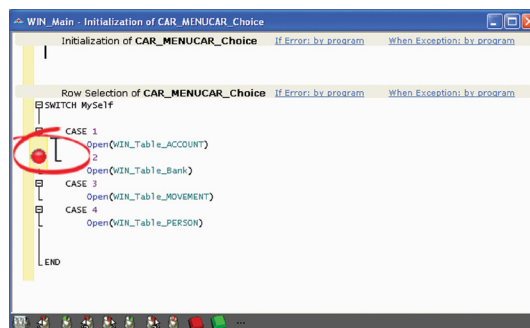
To help you optimize your source code, a debugger is supplied with WinDev. The debugger allows you to run your applications step by step, by viewing the running WLanguage code, the content of the variables, ...



Debugging the project

1. Using a breakpoint, placed in the code editor.

The debugger will be automatically started when the line preceded by a breakpoint is run. To insert a breakpoint, all you have to do is click in front of the code line: a red bullet appears.



2. Using the STOP keyword in the WLanguage code

The debugger will be automatically started when this code line is run.

3. Using the "Project .. Test mode .. Trace the project" option

The debugger is started when the test of the application is run.

4. Pressing [CTRL] + [PAUSE] while running the test of the window or project

The debugger is started during the next action performed after the use of [CTRL] + [PAUSE].

Debugging an executable

WinDev also enables you to directly debug the executable associated with the current project in WinDev. This executable can:

- be started from WinDev, on the development computer.
- be already run on the development computer.
- be started from WinDev, on a remote computer.
- be already run on a remote computer.

Debugging a component

A component is debugged from the project of the component. You can:

- Run the test of the component from the project of the component
- Run the test of the component by running another project found on the development computer. Therefore, the test of the component can be run from the project that uses it.
- Run the test of the component by running an executable found on the development computer. This enables you to run the test of the component directly from the executable that uses it.

Unit tests

Automatic tests

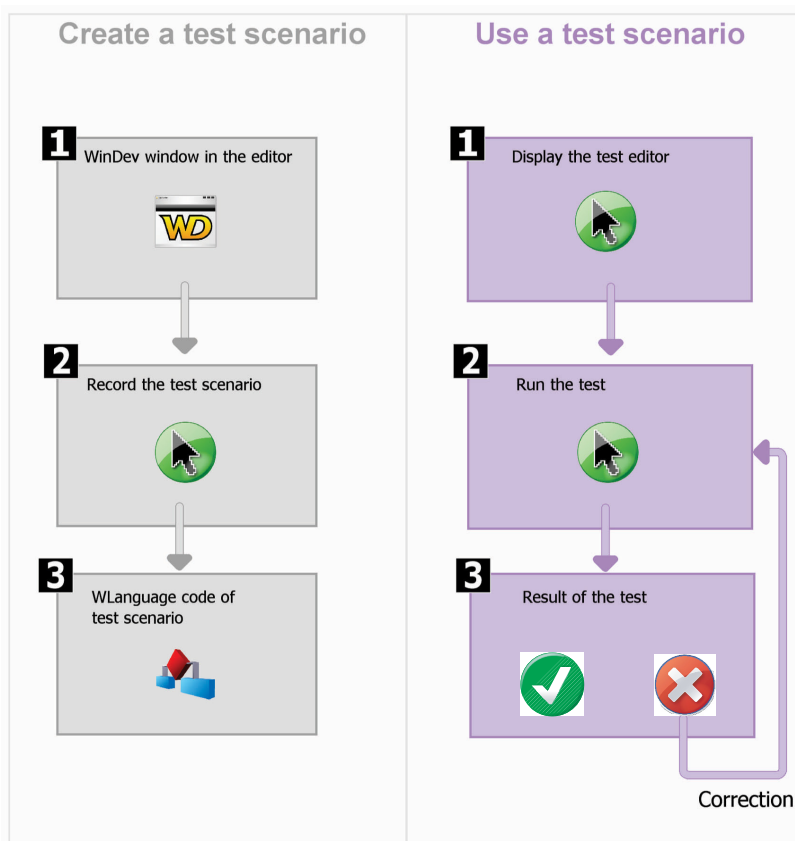
The unit tests (also called automatic tests) are used to run the test of windows, procedures and classes found in an application during the entire development. To do so, select (or create) the test scenario that will be run.


These scenarios are generated in WLanguage and they can be directly modified.

These scenarios are grouped in the test editor. The test editor analyzes the result of the unit tests and calculates the level of validation for the application.

When creating the application executable, WinDev:

- displays the validation rate of the application.
- indicates the modified elements whose test was not run.

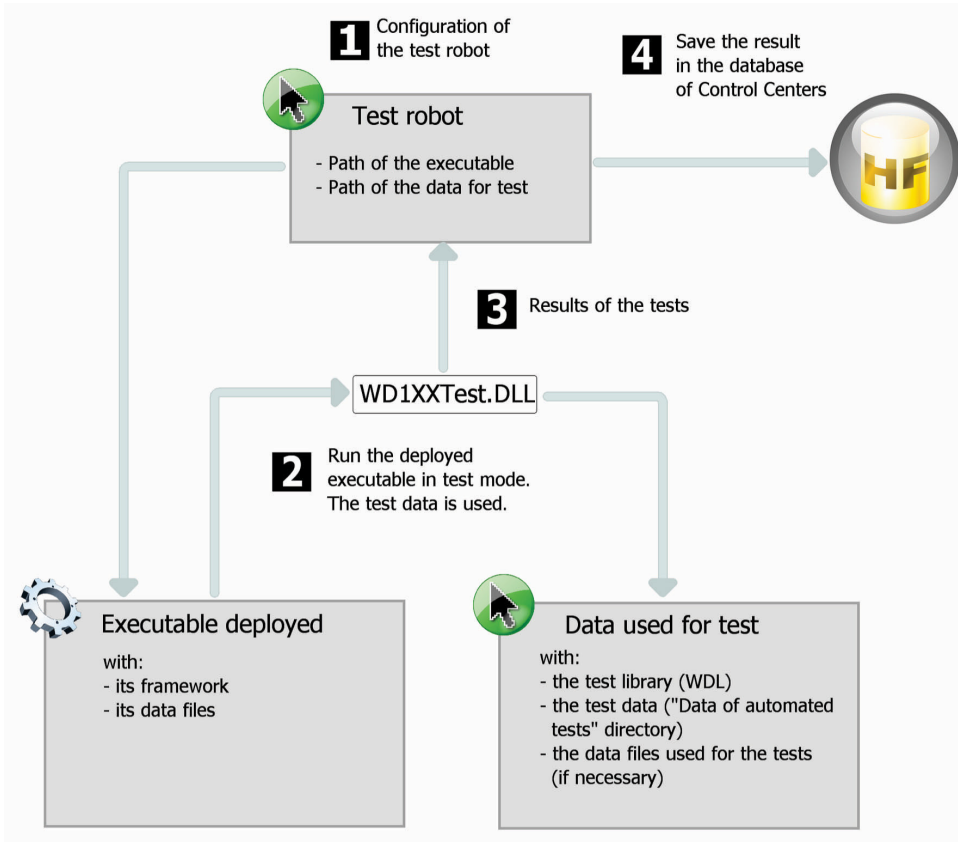


To create a unit test on the current window, click .

To create a unit test on a procedure or on a class, select this procedure or this class in the "Project explorer" pane and select "Create a unit test" from the popup menu.

Unit tests on the executable

The unit tests created by WinDev can be run on a deployed executable, via the test robot.



Test of an application in practice

1 Overview

WinDev proposes several methods to run the test of your applications:

- test of the entire project,
- test of a single window,
- test of a single query (see the "Reports and Queries" guide for more details),
- test of a single report (see the "Reports and Queries" guide for more details),
- running the project step by step,
- test of the performances of your application.
- non-regression tests and automatic tests

The test of the entire project is used to simulate the start of the application by the executable. This enables you to run the test of the entire application, even if its development is not finished yet. As soon as a problem occurs, you have the ability to start the debugger to find out and correct the problem.

Running the test of a single window allows you to run only the current window. This enables you to run the test of your project from a given window or to check the operating mode of a window as

soon as it is developed. Like for the project test, the debugger can be started as soon as a problem occurs.

The test of a single query is used to run the current query. This enables you to check the operating mode of a query as soon as it is developed.

The test of a single report is used to run the current report. This enables you to check the operating mode of a report as soon as it is developed. Like for the project test, the debugger can be started as soon as a problem occurs.

Running the project step by step enables you to start the debugger when the application is started. This solution enables you to monitor the progress of the application.

The performance test of your application or site lets you check and optimize the execution time of your application or site.

The non-regression test (or automatic test) is based on the execution of scripts. This allows you to check that, during the execution of your application ... the existing features are still supported.

2 Running the test of a project and debugging it on the development computer

2.1 Overview

Running the test of the entire WinDev project is used to simulate the start of the application by the executable. This enables you to run the test of the entire application, even if its development is not finished yet. As soon as a problem occurs in the execution of the project, the debugger can be started to identify and fix the problem.

WinDev proposes a complete set of application tests that can be run from the development computer:

- Running the test of the current project and debugging it in the editor.
- Running and debugging the executable corresponding to the current project.
- Debugging an executable that is already running on the development computer.

Note: The test of a project can be run regardless of the current element in the editor.

2.2 Running the test of the current project and debugging it

To run the test of a project from the editor, select "Project .. Test mode .. Debug the project" ([Ctrl] + [F9] key combination). The editor is automatically minimized and the project is run.

2.3 Start the executable and debug it on the development computer

Some problems can only be reproduced when running the test of the executable in real-use conditions. This test mode enables you to debug an executable once it is installed on the development computer.

In this case, the code run comes from the libraries (and not from the code found in the elements found on the computer). Therefore, the code of the current project and the code of the executable can become out-of-sync: the out-of-sync code is underlined in pink during the debug operation.

This test mode is used to run a specific executable installed on the development computer and to directly debug it.

See the online help for more details.

2.4 Debugging an executable that is already running on the development computer

Some problems can only be reproduced when running the test of the executable in real-use conditions.

This test mode enables you to find a problem that only occurs at run time (and not in test mode).

3 Running the test of a WinDev project and debugging it on the user computer

WinDev offers several methods for running the test of a project and for debugging it on the development computer. However in some cases, you may have to debug on the end-user computer (connected by network or by Internet).

From your office in London, you have the ability to debug an application running in Taiwan. The debug operation is performed without having to go anywhere, directly on the configuration of the user.

The principle is straightforward:

- The developer starts the executable on his computer and directly goes to the requested location in the application.
- In WinDev, the developer opens the project corresponding to the executable and asks to be attached to the executable run.
- The WinDev debugger is used to monitor the program execution step by step.


In this case, the code run comes from the libraries (and not from the code found in the elements found on the computer). Therefore, the code of the current project and the code of the executable can become out-of-sync: the out-of-sync code is underlined in pink during the debug operation.

See the online help for more details.

4 Running the test of a window

4.1 Running the test of the window from the editor

To run the test of a window from the editor:

1. Open the window whose test must be run.
2. Select "Code .. Run the window test" (, or [F9]). The editor is automatically minimized and the window is run.

When the test is run, all the features of the window can be run. You will have the ability to open other windows for example.

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 is required for the remote computer.

Note: to debug a WinDev application remotely, the WinDev application must include the WD170CPL.DLL library in its framework.

See the online help for more details.

4.2 Stopping the test of a window

Several methods can be used to stop the test:

- 1st method:
Close the application whose test is currently run. WinDev displays the editor that was used when the test was started.
- 2nd method:
 - Go back to the editor with the task bar or with [Alt]+[Tab].
 - Confirm that the test must be stopped. WinDev displays the editor that was used when the test was started.

5 Tracing a project

5.1 Principles for debugging

Debugging an application consists in:

- checking the operating mode of a process,
- understanding the operating mode of an existing program,
- checking the value of the variables,
- checking the operating mode of the special cases in an application.

The debugger enables you to perform these operations.

Note: WinDev also includes several trace tools (trace window, information box, ...). See "Debugging without the debugger", page 156 for more details.

5.2 Overview of the debugger

The debugger is used to trace the WLanguage programs in order to help you improve these programs. The source code run is viewed on the screen. The processes run are sorted in hierarchical order in the "Debugger" pane.

The value of the variables can be viewed:

- individually in the rollover tooltip of each variable.
- in the "Debugger" pane.

5.3 Features of the debugger

The debugger enables you to:

- find out the call stack
- view the content of the variables or expressions
- run the code step by step with ability to skip blocks.
- use conditional breakpoints
- modify the code while continuing the execution,
- ...

5.4 Debugging without the debugger

In some cases, running a program with or without the debugger may be different.

Indeed, the debugger introduces pauses in the execution of the process during which several tasks are performed by WinDev.

Therefore, the debugger cannot be used in a procedure called by a timer or in the code of a scrollbar.

Note: To find out the limits of the debugger, see the online help.

To debug these applications, you may want to follow the evolution of a value, how different procedures are called, ...

This information can be:

- displayed on the screen.
- stored in a trace file.

Caution: If the information is displayed on the screen, it must only be displayed during the tests of the application.

Displaying the information

Two tools are used to display the information:

- **the information boxes:** WLanguage function named *Info*.

Caution: The display of an information box is a locking operation.

- **the trace window:** WLanguage function named *Trace*. The trace window is displayed in the top left corner of the screen, without interrupting the execution of the program.

Managing the display of the debug information

Displaying the debug information on the screen is useful in test mode only.

Any unsuitable display must be deleted before distributing an application.

To avoid any oversight, we advise you to manage the display of the debug information via a global procedure. For example:

```
PROCEDURE MyTrace (StringToTrace)
  IF InTestMode () = True THEN
    Trace (StringToTrace)
  END
```

In this code, depending on the result of *InTestMode*, the trace window is only displayed during the test of the application.

Such procedure enables you to leave the call to the trace windows in the code of the application without any risk of displaying it on the end-user computers.

The call to the trace procedure is similar to the use of the *Trace* function:

```
MyTrace ("Customer: "+ ...
Customer.CustomerNum)
```

Creating a trace file

For the routines that take a long time to run (batch processes, ...), to check the operating mode of the program, you must keep a physical trace of the processes run (via a text file for example).

The following procedure is used to manage the display of the trace:

- on the screen (/DEBUG parameter in command line).
- in a text file (default mode).

```
PROCEDURE MyTrace(StringToTrace)
  FilePath is string
  FilePath = ...
    fDataDirUser()+...
    ProjectInfo(piProjectName)+".txt"
  File is int
  DebugMode is boolean = False
  IF Position(CommandLine(),...
    "/DEBUG") > 0 THEN
    DebugMode = True
  END
  IF DebugMode = True THEN
    Trace(StringToTrace)
  ELSE
    File = fOpen(...
      FilePath, ...
      foCreateIfNotExist+...
      foWrite + foAdd)
```

```
IF File <> -1 THEN
  DateTimeTrace is DateTime
  DateTrace is Date
  TimeTrace is Time
  DateTimeTrace = SysDateTime()
  DateTrace = MyDate..Date
  TimeTrace = MyDate..Time
  fWriteLine(File,...
    DateToString(DateTrace)+...
    " - "+TimeToString(TimeTrace))
  fWriteLine(File, StringToTrace)
  fWriteLine(File, " ")
  fClose(File)
END
END
```

Notes:

- The trace file is created by default in the data directory of the user. This file is named like the project. This file contains the information to trace during the execution of the program. The information is completed by the date and time of each "Trace". This enables you to detect a potential problem during the process.
- Sample content of the trace file:

```
01/12/2001 - 10:53:25:20
Customer name: Martin
```

6 Performance test

6.1 Overview

The performance profiler enables you to check and optimize the execution time of your application.

Its principle is straightforward:

You run the test of your application. During this test, the performance profiler keeps track of all the actions performed and the corresponding processes run.

At the end of the test, the performance profiler displays:

- the 10 most time consuming operations
- all the actions performed in the application whose test was run, sorted by duration (from the longest one to the shortest one).

You can select a process in order to analyze the reasons for its processing time in order to optimize it.

6.2 Starting the performance profiler

To start the performance profiler, select "Project .. Performance profiler".

The project is automatically run in test mode. The process to optimize can be run in your application.

To go back to the editor, all you have to do is close your application.

Then, the performance profiler displays the result of the analysis.

Note: We recommend that you use the performance profiler to optimize your application (before distributing it for example).

6.3 Reading the result of the performance profiler

The performance profiler presents the result of the analysis in two tabs:

- the "Synthesis" tab presents the 20 longest processes.
- the "Details" tab presents all the processes run during the test of the application (from the slowest one to the fastest one).

The following information is displayed for each process:

Function	Function, process or procedure run.
Total Time	Execution time of the function.
Nb of calls	Number of calls made to the function (procedure or process).
Time 1 call	Execution time of a call to the function (procedure or process).
% code	Percentage of code run outside the call to a WLanguage function or outside the call to a custom function or procedure.
Parent	Process that called the function.

Notes:

- The "Full execution" caption represents the total amount of time for running the test of the application with the performance profiler.
- The "Total for XXX Window" caption represents the total amount of time for running the XXX window (from its opening to its closing).

6.4 Choosing a process to optimize

The process to optimize is chosen according to several criteria:

- **the execution time of the process.** The longest processes must be optimized.
- **the percentage of time spent processing the function or procedure.** The higher this percentage is, the greater the number of processes that can be optimized in the code.

Note: If the process corresponds to a WLanguage function, it is fairly hard to optimize it.

7 Regression tests

7.1 Overview

Several test tools are available to guarantee the quality of your applications:

- The test mode (Go from a project or from a window) is used to immediately test a modification in your application. These tests can be saved as automatic tests.
- The WDTTest too, which allows you for instance to create validation and regression tests in WinDev.

To automate these tests and to increase the quality of your applications, you have the ability to run **automatic unit tests**. These tests are used to easily check all the features proposed by your applications.

7.2 Automatic tests

Each test contains a scenario that can be directly edited in the interface of the product. This scenario is written in WLanguage and it can be modified at any time.

These tests can be run before creating the executable in order to check the operating mode of an application.

The following elements can be checked:

- The WinDev windows.
- the sets of procedures.
- the classes.
- the WinDev executables.

Each test is associated with a WLanguage code: the test scenario. This scenario can be viewed in the code editor. The code of the tests can be modified.

The tests and associated code are not included in the executable and are not provided to the end user. The number of tests for an application has no incidence on the size of the application supplied to the users.

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

7.3 WDTTest

WDTTest is a tool used to record and automatically run test scripts of applications (for WinDev or not).

A test script (also called "macro") contains the different operations performed on an application (mouse movement, mouse click and keyboard input).

Once recorded, the test script can be run as many times as necessary. When a script is run, a result script is automatically generated. The differences between the two scripts (initial script and result script) are highlighted.

This feature enables you to check whether your WinDev applications work properly between two updates

Therefore, WDTTest is used to automate the test procedures of your applications.

Important: To successfully run a test, you must exactly use the same configuration:

- on the computer where the test is created.
- on the computer where the test is run.

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



PART 5

Setup



Express



DEVELOP 10 TIMES FASTER

PCSOFT

The WinDev framework

The WinDev framework is the set of libraries (.DLL files) required to run a WinDev application.

When creating the executable, you can choose to use:

- the common WinDev framework,
- a custom framework.

Using the common WinDev framework

When using the common framework, the libraries are installed in a common directory. These libraries are shared among all the WinDev applications installed on the same computer.

Benefits of the common Framework:

- The disk space used by the framework (about 30 MB) is shared by all the applications.
- The framework is updated once and for all for all the applications installed

Note: when using the common framework, you can replace one or more DLLs (for example, if you require a specific version).

Using a custom framework

When using a custom framework, the libraries are installed in the directory of each application. Therefore, each WinDev application uses its own version of the libraries. You also have the ability to rename the libraries when using a custom framework.

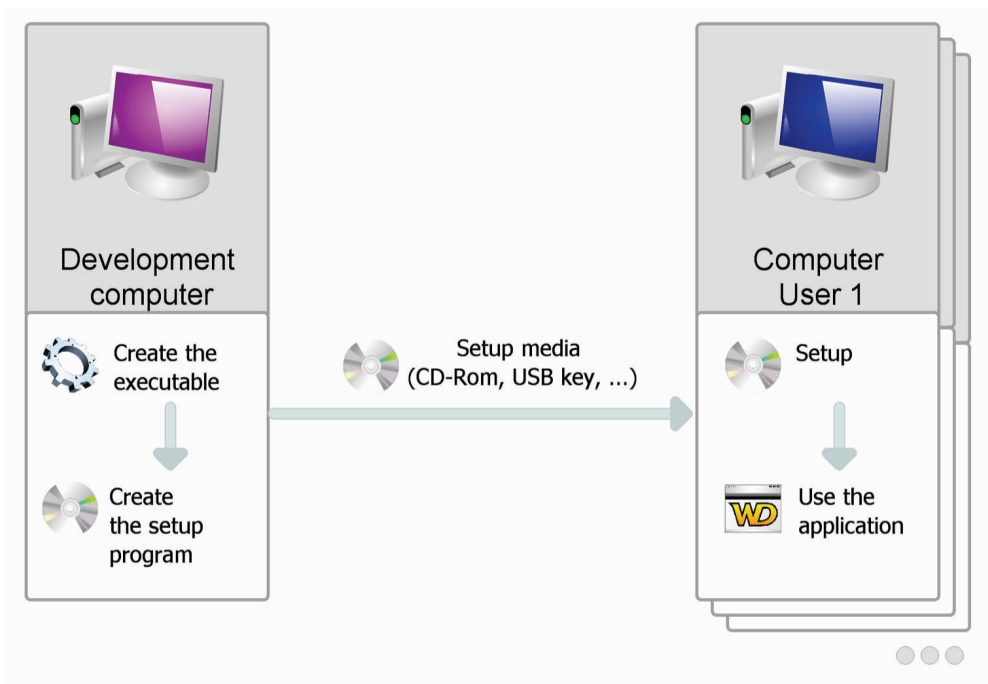
Benefits of the custom framework:

- Each application can exploit a different version of the framework DLLs.
- The framework of a given application can be updated without impacting the other applications.
- The framework can be renamed.

The different modes for installing an application

Standard setup

The standard setup consists in creating a setup program that will be installed on each user computer via a setup media.

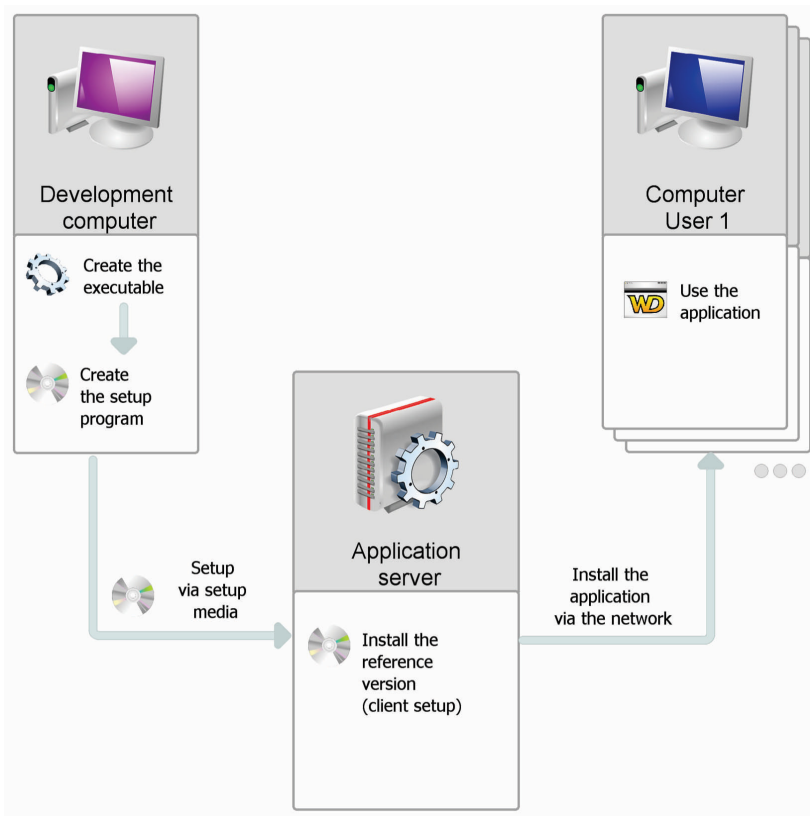


This setup mode is compatible with:

- the data files in HyperFileSQL Classic format installed locally on the user computer,
- the data files in HyperFileSQL Classic format installed on a network and accessible from the user computer,
- the data files in HyperFileSQL Client/Server format installed on a server.

Network setup

The network setup consists in creating a setup program that will be installed on a server. All the user computers that want to install the application will start the setup program via the network.



If an update is available on the application server, the application update will be proposed when the application is started on the user computer.

A network setup allows for a granular control of the user computers (versions deployed, mandatory updates, disconnection of users, etc.)

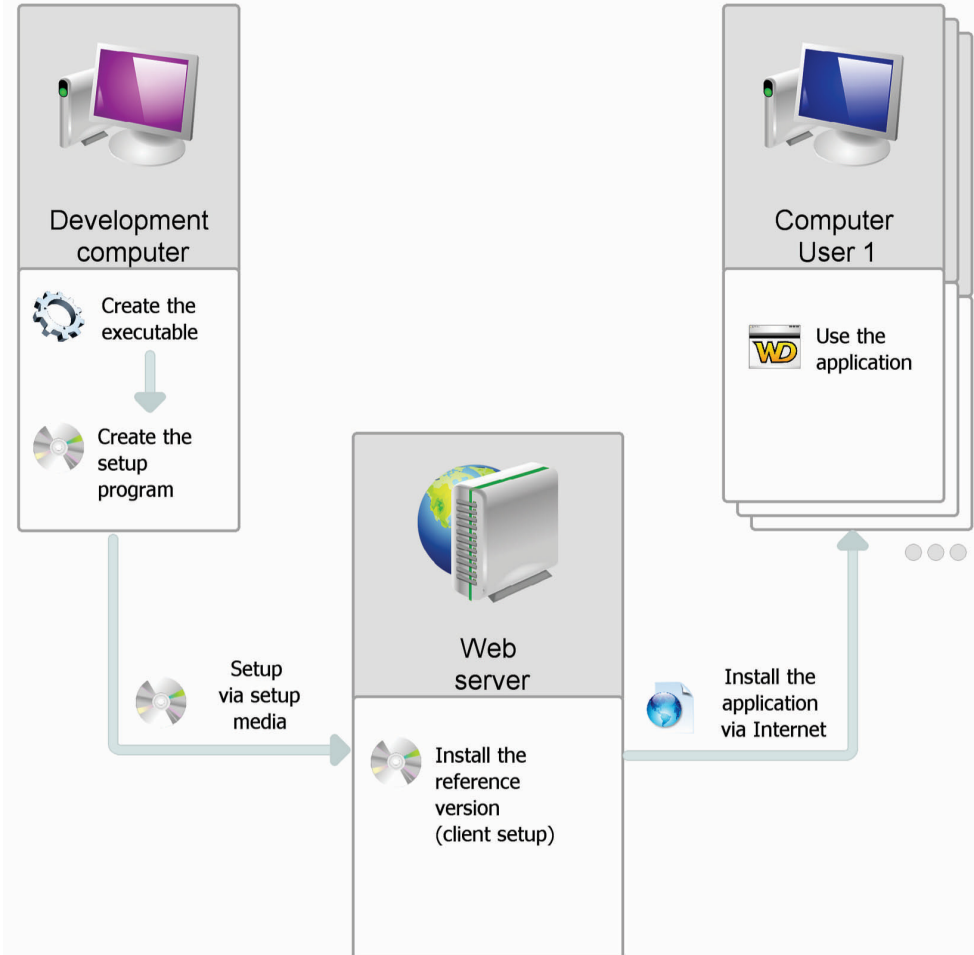
This setup mode is compatible with:

- the data files in HyperFileSQL Classic format installed locally on the user computer,
- the data files in HyperFileSQL Classic format installed on a network and accessible from the user computer,
- the data files in HyperFileSQL Client/Server format installed on a server.

HTTP setup

The HTTP setup consists in creating a setup program that will be installed on a Web server.

All the user computers that want to install the application will directly start the application setup via an HTTP URL.



If an update is available on the Web server, the application update will be proposed when the application is started on the user computer.

Multi-site setup

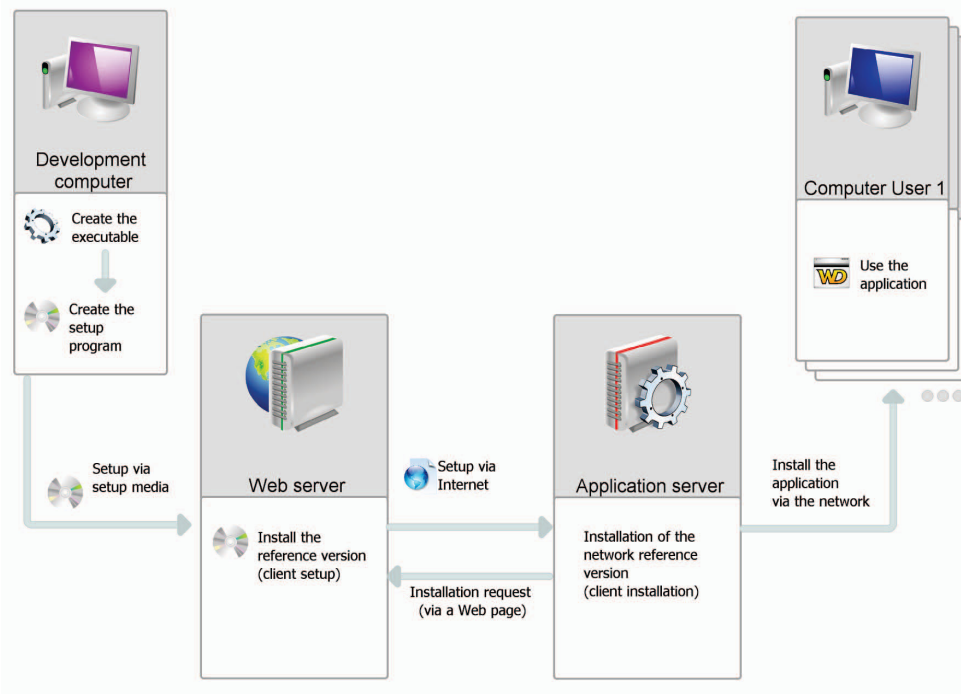
The multi-site setup allows you to install an application on several different sites (a site can represent a subsidiary for example).

The multi-site setup consists in creating a setup program that will be installed on a Web server.

All the sites that want to install the application will start the application setup via an HTTP URL. In this case, the setup program will be installed on a server of the site.

All the user computers that want to install the application will start the application setup via the network.

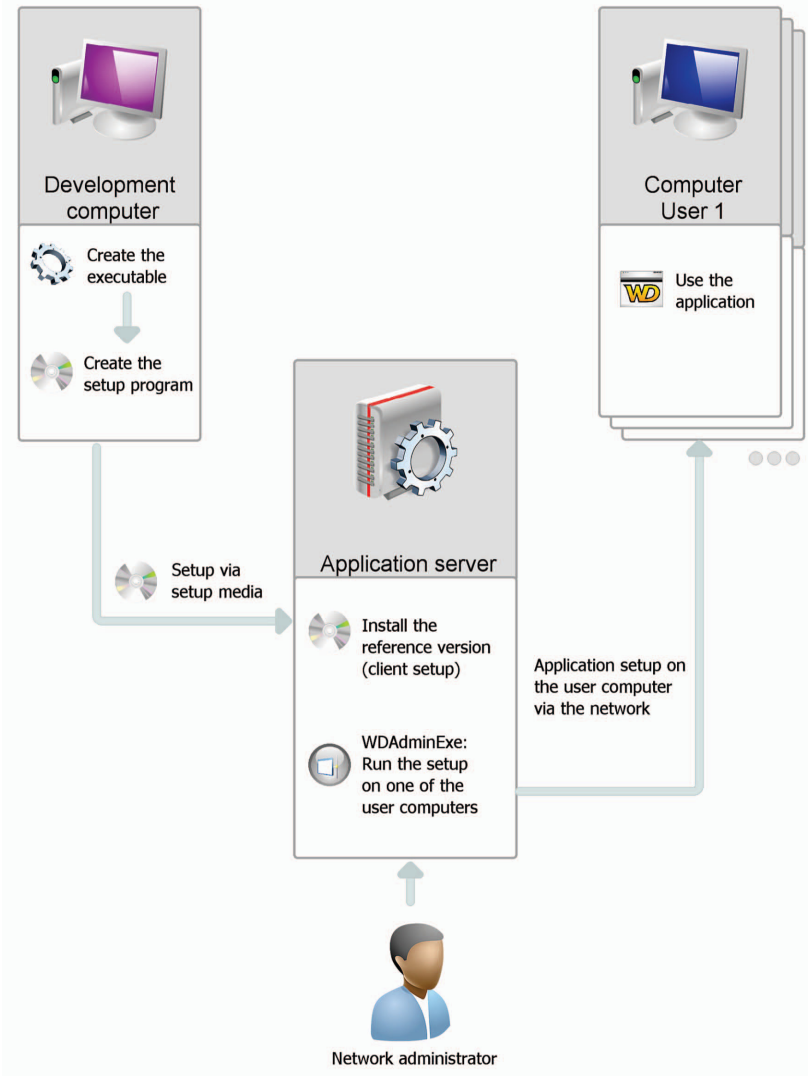
This setup combines the benefits of the network setup and HTTP setup by allowing a large-scale deployment as well as a granular control of the local setups.



If an update is available on the Web server, the application is automatically updated on the application server.

Push setup

The network setup consists in creating a setup program that will be installed on a server. The setup will be performed from the server, by the network administrator on the user computers selected by WAdminExe.



A Push setup can be updated either in Push mode, or as part of a network setup.

Deploying a WinDev application

When developing an application, one of the important steps consists in deploying this application on the user computers.

WinDev optimizes this deployment by offering a large number of choices:

- in the setup media.
- in the tools used to create the setup program.

- in the available setup modes (single computer or network).

This chapter briefly describes the different types of setup proposed by WinDev and their specific features.

See the online help for more details.

1 Setup: Summary table

1.1 Setup media

Depending on the setup medium used (Internet, in-house network, ...), WinDev proposes several modes for generating the setup.

To optimize your setup, choose the recommended generation mode.

		Modes for generating the setup proposed by WinDev			
		Self-extracting executable	Directories, to the size of diskettes	Single directory	CD DVD
Setup media	Internet	X (recommended)			
	CD/DVD			X	X (recommended) Autorun possible
	In-house network	X		X (recommended)	X

1.2 Tools for creating the setup program

Several tools are available for creating the setup program, for maintaining it and for customizing it.

	Wizard	WDINST, setup editor	Customizing WDSETUP
Creating a standard setup program (in French and in English by default)	X (recommended)	X	
Maintaining a setup program	X	X	
Customizing a setup program (the program that is started on the user computer)		X	X (multilingual setup for example)
Start mode	"Workshop .. Create the setup procedure"	"Tools .. WDINST"	WDSETUP project available in the "Examples" sub-directory of the setup directory of Win-Dev.

1.3 The different types of setup

Two types of setup are available in WinDev:

- **Stand-alone setup:** recommended for the applications installed on a single computer. The setup program installs the application on the user computer.
- **Network setup with management of users:** recommended for the applications installed on several computers.

The setup 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 evolves on the server.

2 Creating the setup program

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 custom options are not available.

- **Setup via a MSI package**

A standard setup that uses the features of the integrated Windows installer is performed.

- **Setup with automatic update (network setup)**

This setup enables you to:

- manage the automatic update of the application installed on the network.
- find out 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 installing 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", ...) are also installed on each user computer.

- **Setup with automatic update (network setup in PUSH mode)**

In addition to the network setup, this setup 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.

2.1 Creating a setup program

To create a setup program allowing you to:

- **deploy a WinDev application**, you can:
 - use the wizard for creating the setup program directly from WinDev ("Workshop .. Create the setup procedure"). This wizard enables you to create the different types of setup programs.
 - use WDINST (setup editor) ("Tools .. WDINST-Setup Editor"). This editor enables you to customize the setup program.
- **provide other files**, you need to use the setup creation wizard started from the WDINST setup editor ("File .. New" from WDINST).
Reminder: To start WDINST, select "Tools .. WDINST - Setup editor".

2.2 Installing a WinDev application

Standard, compact or MSI setup

To install your application, all you have to do is start the setup program of the application on the user computer. The files required by the application are installed in the specified setup directory.

To install an update of your application, all you have to do is run the setup program of the update on the user computer. The application files modified by the update are replaced.

Setup with automatic update

To install your application:

1. Start your setup program on the server. The files required by the reference application are installed in the specified setup directory. The reference application corresponds to a client setup program.
2. Run the client setup program from each user computer. The files required by the client application are installed in the specified setup directory.

To install an update of your application:

1. Start your setup update on the server. The files of the reference application modified by the update are replaced. These replaced files can be saved.
2. When the application is started from a user computer, the application update is automatically proposed.

Setup with automatic update in PUSH mode

To install your application in PUSH mode:

1. Start your setup program on the server. The files required by the reference application are installed in the specified setup directory. The reference application corresponds to a client setup program.
2. The program for managing the network applications is automatically started.
3. Select the network computer on which the setup must be performed.

To install an update of your application in PUSH mode:

1. Start your setup update on the server. The files of the reference application modified by the update are replaced. These replaced files can be saved.
2. Start the program for managing the network applications.
3. Select the network computer on which the update must be performed.

Monitor your applications

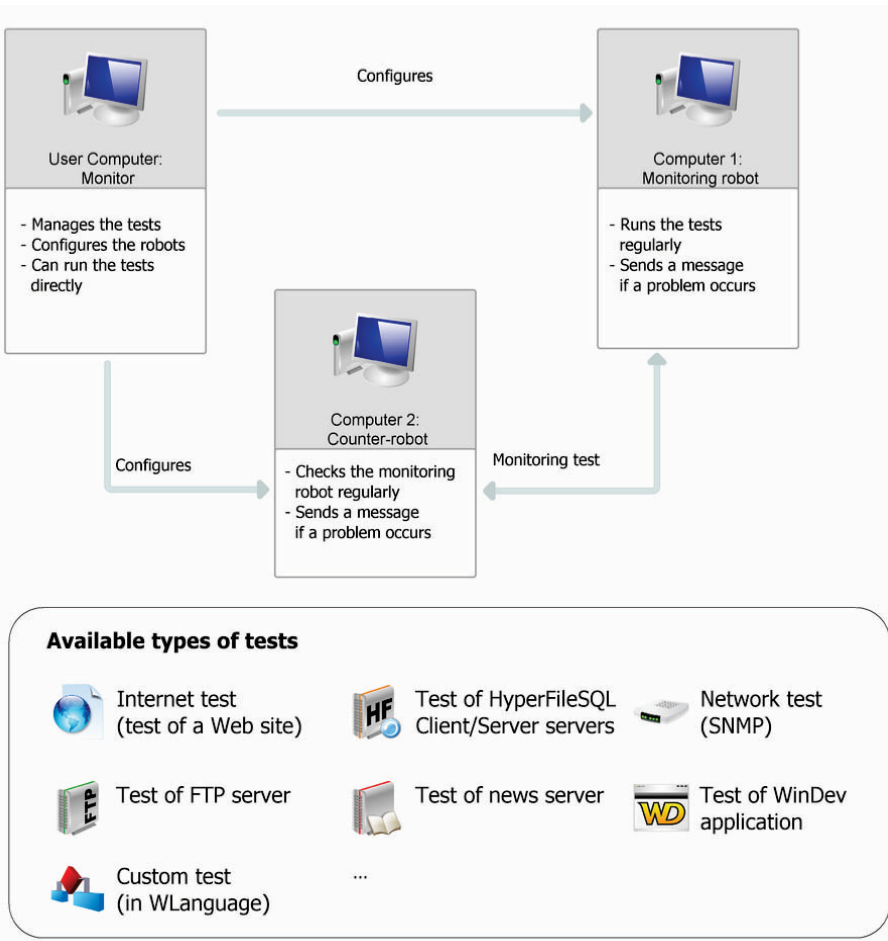
An application depends on several external parameters: Web site, HyperFileSQL server, network, ... To optimize the management of incidents, WinDev proposes to use a monitoring robot.

Made of three executables started on different computers, the monitoring robot is used to run different tests: check the Internet connection, network tests, ...

If problems occur when running a test, different methods can be used by the monitoring robot to inform you:

- Message sent in the PC SOFT messaging (WDBAL).
- Email
- Run another program.

Furthermore, a sound alert can be implemented on the monitor.





PART 6

Communication



Express



DEVELOP 10 TIMES FASTER

PCSOFT

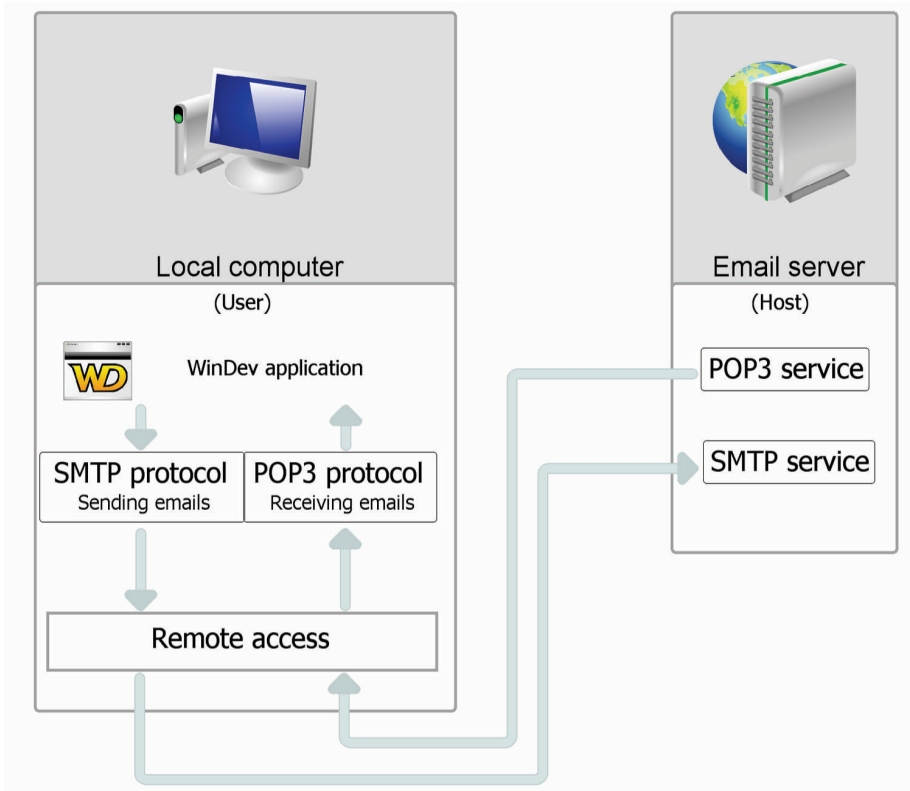
Managing the emails

WinDev proposes several methods for managing the emails:

- the POP3 or IMAP protocol and the SMTP protocol (most common method)
- the "Simple Mail API API (also called SMAPI or Simple MAPI)"

POP3 or IMAP and SMTP protocols

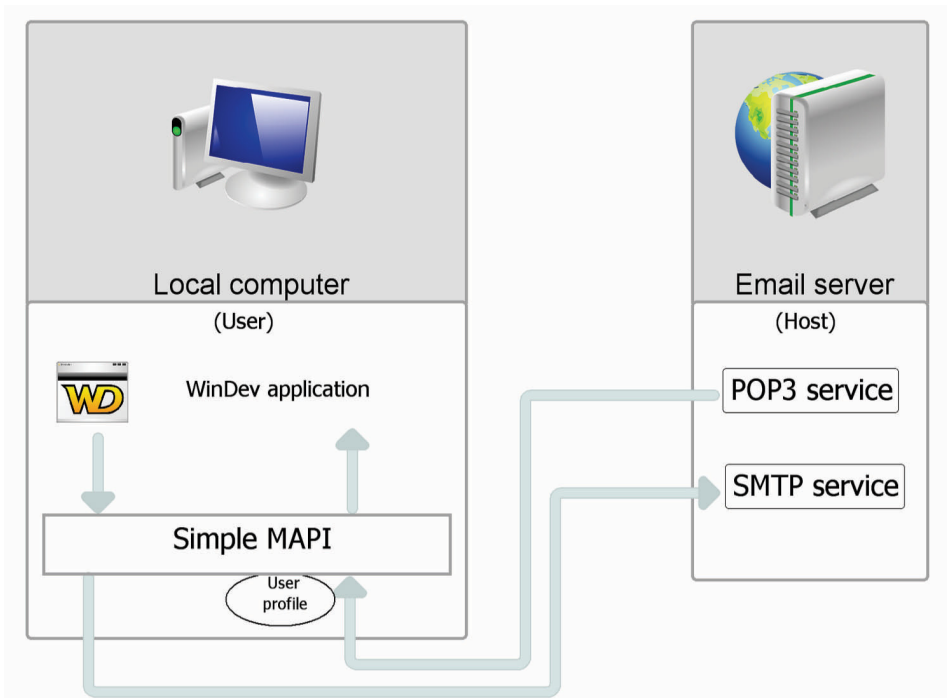
These protocols are the standard Internet protocols for email management. They are standardized by the RFC 1939, 3501 and 5321 documents. These protocols allow you to directly communicate with the server, available at your ISP.



- POP3 (Post Office Protocol) is a protocol for reading emails.
- IMAP (Internet Message Access Protocol) is a protocol for reading emails remotely.
- SMTP (Simple Mail Transfer Protocol) is a protocol for sending emails.

"Simple Mail API (also called SMAPI or Simple MAPI)"

This mode for email management is used by most Microsoft applications, and specifically MS Exchange 4.



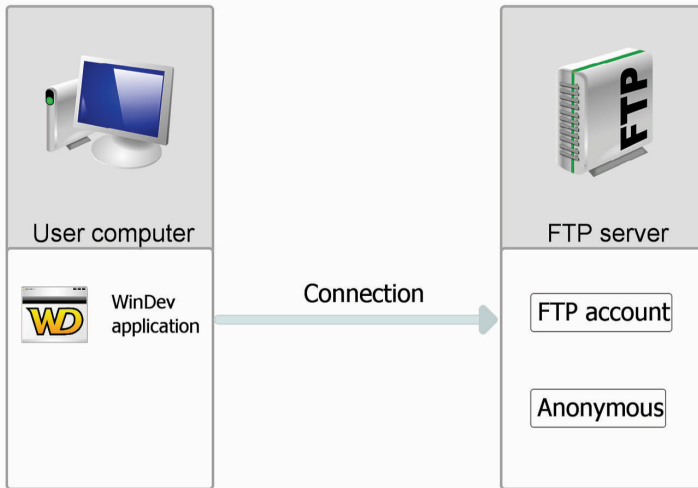
Transferring files by FTP

The FTP (File Transfer Protocol) is a protocol used to transfer files from a site to another remote site. This protocol is used to exchange files via TCP/IP or Internet.

Principle

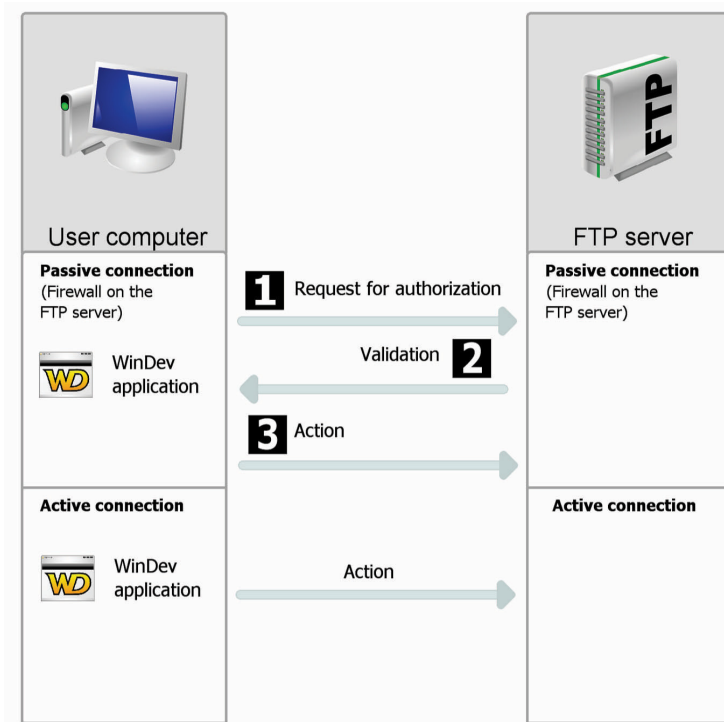
To handle the files found on an FTP server from a WinDev application, you must:

1. Establish the connection to an FTP server (**FTPConnect**).

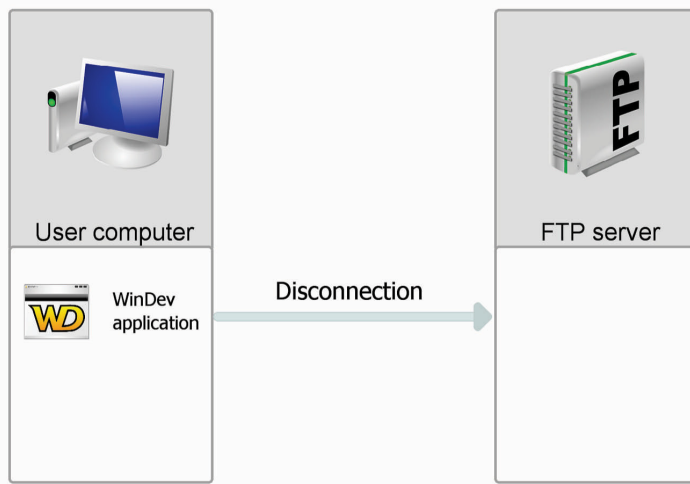


2. Send and retrieve files (according to the connection mode). For a passive connection, the WinDev application must request the authorization from the FTP server before any operation is performed on the server files. It also enables you to:

- find out the characteristics of the files found on an FTP server: attributes, size, ...
- handle the files found on an FTP server: creation, deletion, ...
- list the files of a directory found on the FTP server by running a procedure used to perform a process for each file found.



3. Close the connection to the server (**FTPDisconnect**).



Managing the sockets

WinDev proposes several functions used to perform an advanced management of sockets.

A socket is a communication resource used by the applications to communicate between computers regardless of the network type.

This communication mode can be used, for example, to establish a communication between computers connected by Internet.

Different possibilities

A WinDev application can manage the sockets according to different modes:

- **Client WinDev application:** the application connects to a standard server and exchanges data via a socket.
- **"Simplified server" WinDev application:** the WinDev application is a server, exchanging information via a socket with a single client computer.
- **"Standard server" WinDev application:** the WinDev application is a server, exchanging information via sockets with several client computers.

Principle of a client application

A client application of a socket server connects to a standard server in order to exchange information via a socket.

Example: A WinDev client application can connect to a standard news server on Internet.

Step 1: Connecting to the server

To connect to a server socket, use **SocketConnect**. This function is used to establish a request for connecting to the server. The socket is identified by its port and by its address.

Step 2: Exchanging data

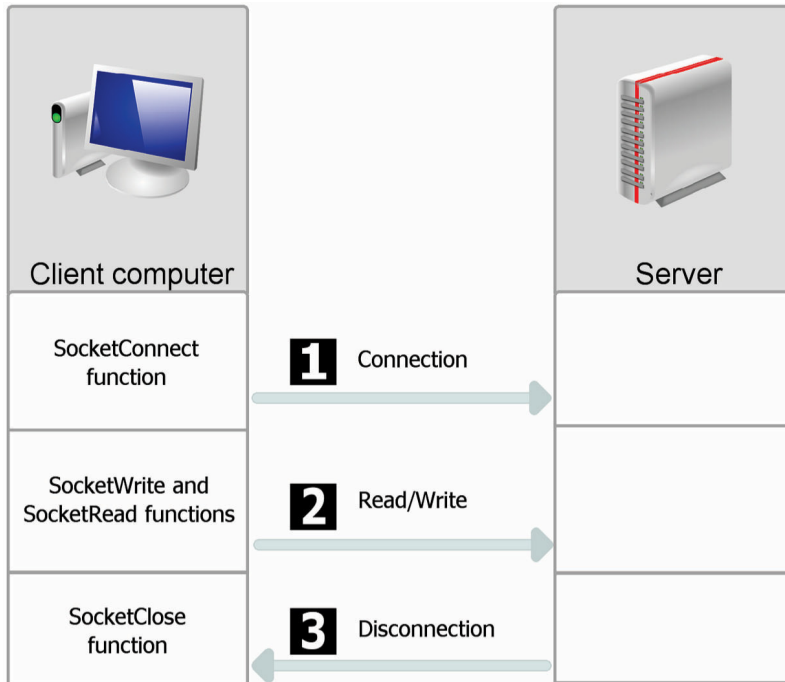
Once two computers have connected their socket, a communication channel is established between these two computers. These two computers can read and write character strings on the socket.

To read and write on the server socket, the WinDev client application must use **SocketRead** and **SocketWrite**

Step 3: Ending the communication

To end the communication, close the socket from the client computer with **SocketClose**

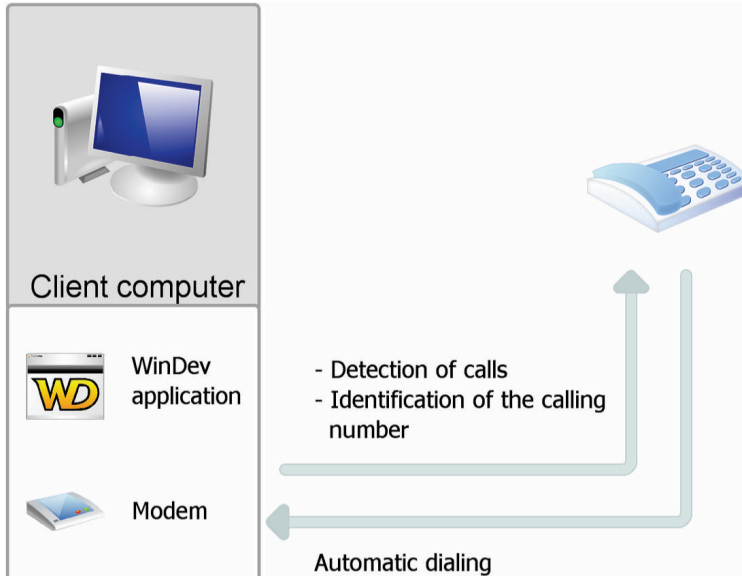
The different steps can be represented as follows:



Note: the SocketXXX functions can also be used to manage secure sockets according to the SSL protocol. To do so, you must use **SocketCreateSSL** and **SocketConnectSSL**.

WinDev and telephony

WinDev allows you to easily manage incoming and outgoing phone calls via the telephony functions. These functions are used to manage a voice box, an answerphone, ... in a WinDev application directly.



Required configuration:

To be able to use the telephony features, you must have:

- a modem.
To save and play recordings, this modem must have voice support.
To identify the caller, the modem must support Caller ID.
- the Tapi 2.0 protocol.

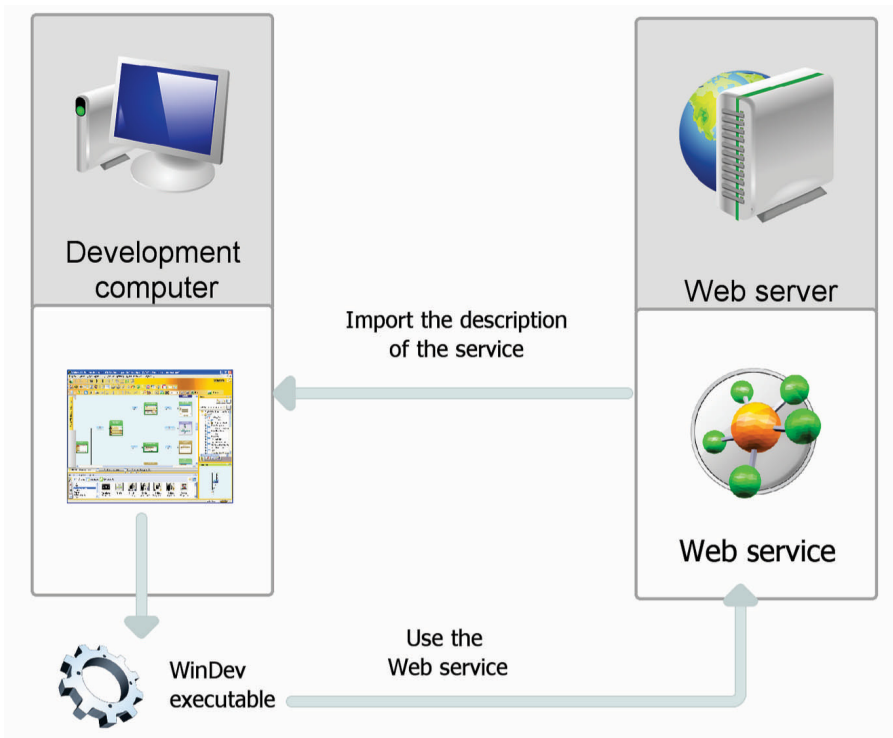
Note: This technology can be used on any system, however you must:

- install the service pack 4 (or later) if the system used is Windows NT4
- upgrade the system if the system used is Windows 95 (can be downloaded from <ftp://ftp.microsoft.com/developr/tapi/tapi2195.zip>). (note: address valid at the time this documentation is being written)

The Webservices

WinDev allows you to easily use the Webservices (or XML Web services). A Webservice is an application available on the Web and that is accessed by using the SOAP protocol (Simple Object Access Protocol).

Any WinDev application can use an existing Webservice. To do so, you must know the address of the Webservice and import its description into the application.



Note: you can also expose one or more sets of procedures of a WinDev application in the form of a Webservice. To do so you must create a project configuration dedicated to this use, generate the Webservice and deploy it on a WebDev Application Server.



PART 7

Appendices



Express



DEVELOP 10 TIMES FASTER

PCSOFT

Components supplied with WinDev

WinDev is supplied with several components. These components are supplied with their source code and with an example.

The list of the main components supplied with WinDev is as follows:

Component Name	Description
Agent	Used to manage an agent.
Assisted directory input	Enables you to implement the assisted input for a directory path or for a file path in an edit control.
Balloon tip	Enables you to manage the display of a BalloonTip (advanced tooltip, displayed above the taskbar).
BankHolidays	Enables you to: <ul style="list-style-type: none"> - display a calendar and select a date - manage the public holidays in a calendar
CalcHFStat	Used to keep the HyperFileSQL index statistics updated without programming.
Client socket	Simplifies the use of the functions for socket server. This component manages the connection to a server. All you have to do is manage the dialog with the server.
Countries of the world	Contains the list of all the countries of the world as well as their characteristics: <ul style="list-style-type: none"> - ISO code of each country - list of regions, states or countries making up a country (including the time difference for each state, region or country) - time difference (GMT) - main currency of the country - main languages - flag of each country - capital - dialing code
Did you know it	Enables you to display tips and tricks from different sources (preset tips in the code or from a file).
DisplayImage	Used to write a note in an image (screen shot for example). You have the ability to insert some text, lines, etc.

Email assisted input	Provides assisted input when the user saves an email.
EventLog	Used to: <ul style="list-style-type: none"> - List and read the events saved in the event log of Windows. - Add an event into the event log of Windows.
Extended Table	The extended tables (also called "TableView") are used to display hierarchical information in a table. These extended tables are often used in messaging software to list the messages and the answers as well as the date, the time and the author of the messages.
Fax Management	Used to manage printing and sending faxes (operates in Windows 2000 and Windows XP only).
Floating help	Used to display temporary information to the user without interrupting his work.
Gantt chart	Enables you to create a Gantt chart from a task list.
GraphicList	Used to manage a graphic list in an internal window.
Last documents	Enables you to propose a menu option used to open the last documents opened in the application.
Limits	Used to limit an application by different methods: <ul style="list-style-type: none"> - Time limitation (the application can be used during a 30-day period) - Limitation to a single computer (the application can be run only on a single computer) - Limitation on the number of users (the application cannot be run by more than n users)
Login Management	Centralizes the management of users and passwords.
Managing Logs	Used to define, for an application, a configuration for the runtime logs: <ul style="list-style-type: none"> - number of log files that must be stored, - maximum size for each log file, - information that must be logged. This example simulates different processes to observe the management of log files (.wlog files).
Photocopier	Enables you to photocopy documents via a scanner and a printer.
Pickers	Groups all the pickers available in WLanguage. Each picker automatically stores the last path selected in the registry. This enables the user not to re-browse all the folders to find the files and/or the directories that he wants to select.

Pocket 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 tactile devices, TabletPC in Windows XP, interactive devices, ...
Post emails	Used to: <ul style="list-style-type: none"> - create preset messages for the next outgoing emails - create contacts and lists of recipients - create messages and send emails
Print spooler	Defines the list of less used printers in order to perform one or more prints.
Printout model	Enables you to manage the printout profiles.
RTF search	Enables you to perform any type of search in an RTF control.
Scanner	Enables you to digitize a document via a user-friendly interface.
Server socket	Simplifies the use of the functions for socket server. This component manages the server startup and the acceptance of client connections. All you have to do is manage the dialog with the client applications.
Simplexe	Used to perform calculations from the Simplexe algorithm.
SMSs	This component is used to send SMSs from a simple Internet connection.
SQL connection	Establishes the connection to the database, via the information given by the user when displaying a window for connection setting.
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.
Text extraction	This component is 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.
Visual Memory zone	Used to view the entire content of a memory zone in a window. Can also be used to modify and add values into the memory zone.
WD Communication	Used to establish a communication between two remote applications. These two applications can be: <ul style="list-style-type: none"> - On remote computers. The communication is performed via sockets. - On the same computer. The communication is performed via the Windows messages.
WD Explorer bar	Used to manage an address bar similar to the address bar of the explorer found in Windows Vista.

<p>WD Glossary</p>	<p>Used to organize and store the written expressions regularly used by a customer.</p> <p>This allows you to propose an access to a "Glossary" that can be easily customized by the end user. All the glossary entries can be organized in tree-like categories.</p>
<p>WD Search engine</p>	<p>Used to index documents (text, videos, sounds, ...) according to an identifier managed by the user (can come from a HyperFileSQL data file,...) and according to keywords (text found in a document, file name, ...).</p> <p>It can also be used to perform searches on the documents that have been indexed and to generate statistics about indexing (unused keywords, number of documents per keyword, ...).</p>
<p>WD Tactile Screen</p>	<p>This component is used for interfacing a touch screen that can replace the standard keyboard. This keyboard enables you to use a WinDev application on a computer not equipped with a keyboard.</p>
<p>WD WIFI</p>	<p>Used to:</p> <ul style="list-style-type: none"> - list the network cards available on the current computer - list the available WIFI networks - find out the quality of the WIFI signal - get information about an existing WIFI connection

Two sub-directories are specific to each one of these examples:

- the "<ComponentName>-Example" sub-directory contains an example of project that uses the component,
- the "<ComponentName>-Source" sub-directory contains the project of the component.

Additional components are supplied with the Technical Support Newsletter (LST) or they can be downloaded from our site (www.windev.com).

Examples supplied with WinDev

The examples supplied with WinDev are intended to help you learn the features of WinDev.

Their source code is presented in details.

These examples are found in the "Examples" sub-directory of the setup directory of WinDev and they can be opened directly from the "Wizards, Examples and Components" pane.

Different types of examples are supplied with WinDev:

- Full examples: these examples correspond to full applications that can be used without any adaptation.
- Educational examples: these examples illustrate a specific feature.

Let's see the features of some examples supplied with WinDev.

Full examples

CRM	Enables you to manage the customers, orders and estimates as well as a task list and a schedule.
Sales Management	This application is made of 5 projects used to manage all the aspects of sales management (customers, suppliers, products and stock).
WD Active .Net Directory	Enables you to view the content of an Active Directory.
WD Answering Machine	Uses the telephony features to manage an answerphone. This answering machine can be interrogated and configured remotely via a menu.
WD Burner	Used to burn audio CDs and data CDs, DVDs or Blue-Ray disks.
WD Burner Lite	Used to burn data CDs, DVDs or Blue-Ray disks.
WD Dictaphone	Proposes the features of a tape recorder. It is used to record a sound in "WAV" format and to play it back.

WD DOTNET WMI	<p>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.</p> <p>It gives you the ability to list the devices installed on the computer (networks, video cards, processors) as well as the applications (from the operating system to a "regular" application).</p>
WD Euro Calculator	Presents the multi-currency conversion via a calculator.
WD File synchronization	<p>Used to synchronize two file directories.</p> <p>Once synchronization is performed, the destination directory is identical to the source directory. The comparison is performed on the presence and on the date of the files.</p>
WD FTP File Transfer	<p>Enables you to transfer files:</p> <ul style="list-style-type: none"> - from an FTP server to the local computer, - from the local computer to an FTP server.
WD FTP Threads	Used to simultaneously transfer several files found in the same directory via FTP.
WD FTPProgressBar	Presents the use of an internal window to display an advanced progress bar.
WD Generating data	Generates test data for your applications. This data is stored in data files external to the application. To handle these data files, there is no need to associate an analysis with the project.
WD Holidays and Flex Time	<p>Full management of the holidays and flex time for the company employees.</p> <p>Each user can display his own dashboard presenting the status of his holidays and flex time.</p>
WD HTML Album	Used to manage an HTML photo album.
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 consulted.
WD Internet Search	Used to control different search engines on Internet (Yahoo, Alta Vista, Voilà, Lycos, Excite, Nomade, Google, Euroseek, etc.). The result of the search is displayed via an ActiveX.
WD LDAP Explorer	Used to view the content of any LDAP directory and to modify the LDAP data.
WD Live Messaging	Presents the functions for managing the sockets.

<p>WD Loan</p>	<p>Enables you to simulate loans and to:</p> <ul style="list-style-type: none"> - calculate the amount of monthly payments according to the amount of the loan - calculate the amount that can be borrowed according to the amount of a monthly payment - calculate the interest on investment from a monthly deposit <p>For each case, you have the ability to print the depreciation schedule corresponding to your parameters.</p>
<p>WD Magnifier</p>	<p>Used to zoom a section of the screen via a magnifier. The zoom value can be multiplied by 8.</p>
<p>WD Mail</p>	<p>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.</p>
<p>WD Manager of .Net tasks</p>	<p>Used to list the applications and processes via the .NET functions. Enables you to perform different processes on the applications.</p>
<p>WD Managing a computer fleet</p>	<p>Used 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.</p>
<p>WD Managing Contacts</p>	<p>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 found in Outlook</p>
<p>WD Managing orders</p>	<p>Simplified management of orders/invoices.</p> <p>Used to:</p> <ul style="list-style-type: none"> - create/modify/delete a product, - create/modify/delete a customer, - contact a customer by email, - show the history of the actions performed for a customer, - place an order, print an order form, - invoice an order, print an invoice, automatically email an invoice in PDF format to the customer.
<p>WD Messaging Client</p>	<p>Messaging that uses the RPC features of WinDev for the dialog with the message server. This example is associated with the "WD Messaging server" example.</p>
<p>WD Messaging Server</p>	<p>Messaging server that must be used with the WD Messaging Client example.</p>

WD Multimedia	<p>Enables you to read and play:</p> <ul style="list-style-type: none"> - videos (AVI, MPEG,...) - sound files (MP3, WAV), - audio CDs, - Flash animations.
WD Organizer	Used to synchronize the appointments in your Outlook, Lotus Notes and Google calendars.
WD Outlook	<p>Enables you to read and write in the Outlook folders:</p> <ul style="list-style-type: none"> - Messaging - Calendar - Contacts - Tasks. <p>This example is intended to operate with the standard version of Outlook supplied with Office.</p>
WD Password	Presents a "box" of passwords. It can be used to manage the passwords created when using Internet sites but also in applications or in everyday's life (codes for credit cards, etc.) .
WD POP3Proxy	This example presents a POP3 proxy. In this example, the proxy is used to automatically archive the emails in a HyperFileSQL database.
WD Puzzle	Explains how to use the Drag&Drop functions as well as the functions for handling images in WLanguage.
WD RSS Reader	Presents the use of classes for reading the RSS streams
WD RTF editor	Used to edit text with RTF formatting
WD Scanner - Twain	<p>Used to:</p> <ul style="list-style-type: none"> - Preview a section of a document - Digitize a document (GIF, JPEG, BMP format) - Digitize a document and copy it into the clipboard - Digitize a document and print it
WD Schedule	Used to create a schedule containing click areas.
WD Shipping costs	Calculates the shipping costs for a package.
WD Telephony	<p>Uses the telephony functions of WinDev to:</p> <ul style="list-style-type: none"> - Dial a phone number - Detect and identify the incoming calls
WD Video conference	<p>Enables you to start a conference with a remote user.</p> <p>Used to send a file to a recipient and to establish a dialog with him.</p>
WD Video surveillance	Used to broadcast the images recorded by a Web camera via a network (Internet, Intranet, etc.).

WD ViewTIF	<p>Small image editor developed in WinDev for the images in TIF format.</p> <p>WD ViewTif is used to open an image and to display the different pages of the image</p> <ul style="list-style-type: none"> - in MDI mode (in several child windows), - in slide show mode.
WD WebCam	<p>Presents the integration of a Web Camera in a WinDev window.</p> <p>You can:</p> <ul style="list-style-type: none"> - View the image broadcasted by the camera - Perform an immediate capture - Capture the scene broadcasted as a Windows animation ("AVI" file)
WD Zip	<p>Used to create and read the compressed archives in "WDZ" format (WinDev ZIP).</p>

Educational examples

WD Advanced input mask	<p>This example presents the advanced input masks:</p> <ul style="list-style-type: none"> - Format of positive and negative numbers in a numeric edit control - Appearance of negative numbers in a numeric edit control - Appearance of the value 0 in a numeric edit control - Regular expression used to prevent from entering characters other than 1, 2, 3, 4, 5 and 6. - Regular expression used to "control" the input of a French registration number.
WD Advanced reports	<p>Presents the different types of reports. Used to illustrate:</p> <ul style="list-style-type: none"> - the internal reports, - the chart control, - printing in a RTF, HTML or PDF file.
WD Advanced Rotation	<p>Used to perform the rotation of an image from any point that will be used as rotation center.</p>
WD Age Pyramid	<p>Explains how to create an "Age pyramid" chart from two histograms.</p>
WD Agent	<p>Includes an "agent" in a WinDev application via an ActiveX.</p>
WD Alarm	<p>Displays an alarm message in the title bar of the active window (regardless of the application). The message can be associated with a sound alarm.</p>
WD Animated Menu	<p>This example proposes different menu animations.</p>

WD Animated	<p>Presents the main animations that can be performed in an Image control.</p> <p>This example proposes:</p> <ul style="list-style-type: none"> - animations without code, - programmed animations.
WD BarCode	<p>Used to create products and to associate them with an existing bar code via a bar code reader. These products can be printed as labels.</p>
WD BGInfo	<p>This example explains how to write information on the background of a computer screen (IP address, name of the computer, today's date etc...).</p>
WD Bluetooth	<p>This example presents the use of the Bluetooth functions:</p> <ul style="list-style-type: none"> - to list the accessible Bluetooth devices - to list the services proposed by a device - to connect to a Bluetooth device - to send a file to a device via the "OBEX" functions of WLanguage.
WD Calculations on date and times	<p>Performs several calculations on the dates and times via procedures:</p> <ul style="list-style-type: none"> - Calculate the duration between two moments (dates and times) - Calculate a sum of durations - Calculate an average of durations
WD Carousel	<p>This example presents the use of the Carousel control of WinDev. It can be handled like a ListView control.</p>
WD ChainedList	<p>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.</p>
WD Changing Skin Templates	<p>This example presents a solution allowing the users to dynamically change the skin templates.</p>
WD Characters	<p>This program, powered by WinDev, is used to insert some special characters into an RTF control and to view the ASCII and ANSI codes</p>
WD Chart	<p>This educational example presents the use of the chart control. The available charts are: Semi-circular, Donut, Pie, Line, Scatter, Histogram, Radar, Area.</p>
WD Click area	<p>Manages the click areas on non-rectangular shapes.</p> <p>When the map is clicked, the selected region is displayed at the bottom of the screen.</p>
WD Click on Chart	<p>This example proposes a solution used to offset the sections of a Pie chart via a simple click.</p>

WD Clickable map	<p>Presents a cartography application based on a MICHELIN map (authorization #9905251).</p> <p>In the left section of the window, the entire map is displayed with gridlines. When one of the cells found in the gridlines is clicked, the corresponding region is drawn in the right part of the window.</p>
WD Clickable reports	<p>The example is used to list the customers and to print this list of customers.</p> <p>During the print preview, a click performed on the name of the customer opens the form of the selected customer.</p>
WD Closing Windows	<p>Enables you to:</p> <ul style="list-style-type: none"> - Close a Windows session - Stop and restart the computer
WD Coloring Search	<p>This example explains how to highlight the sought words in the results of a full-text search.</p>
WD Controlling Excel	<p>Used to control Excel via OLE Automation. All the Excel functions can be used.</p>
WD Controlling IE	<p>Used to control IE via OLE Automation. All the IE functions can be used.</p>
WD Controlling Open Office Database	<p>This example proposes a class used to easily handle an Open Office database without having to handle any OLE objects.</p>
WD Controlling OpenOffice Calc	<p>This educational example explains how to control the OpenOffice.org spreadsheet. Indeed the workbooks opened in OpenOffice.org can be created, opened and used from a WinDev application.</p>
WD Controlling OpenOffice Writer	<p>This educational example explains how to control OpenOffice Writer. Indeed the documents opened in OpenOffice Writer can be created, opened and used from a WinDev application.</p>
WD Controlling Outlook	<p>Used to control Outlook via OLE Automation. All the Outlook functions can be used.</p>
WD Controlling Spreadsheets	<p>This example explains how to control the OpenOffice Calc spreadsheet and the Excel spreadsheet with a generic code.</p>
WD Controlling word processor	<p>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.</p>
WD Controlling Word	<p>Used to control Word via OLE Automation. All the Word functions can be used.</p>
WD CSGroupware	<p>Presents the user groupware with a HyperFileSQL Client/Server database.</p>
WD Curve Smoothing	<p>Allows you to smooth a "line" chart by using an algorithm of "cubic splines".</p>

WD CustomCalculations	Proposes a solution to allow a user to enter a calculation formula in a table column.
WD Data Binding	The "WD DataBinding" example explains how a control can be linked to a WLanguage variable.
WD Data encryption	Presents the different methods for encrypting the data (character strings or text files).
WD Detecting errors	Presents the module for assisted management of HyperFileSQL errors: - the default operating mode, - the customization options.
WD Device Opening	This example is used to add or delete an application into/from the automatic opening system for the removable devices by programming.
WD Direct printout	Used to print on the dot-matrix printer connected to the parallel port of your choice.
WD Directory Observer	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 DndTVTable	This example presents the use of programmed Drag & Drop in a treeview table.
WD Drawing lines	Enables you to draw the line of a function with parameters (mathematical function). You have the ability to use preset functions or to create your own functions.
WD Drawing on Chart	This example is based on an internal component used to draw elements on a chart control: min/max areas, various areas, flags, plots and monitoring trackers,...
WD Draw	Highlights sections of a scanned document or annotates the images via the drawing functions of WLanguage.
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 a database can be duplicated on a server.
WD Dynamic compilation	Presents: - The dynamic compilation of WLanguage code - The execution of dynamically compiled code - The process of the possible runtime errors

WD Evaluation period	Enables 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"> - the test of a window - the test of a global procedure
WD Export Table	Exports the content of a table to: <ul style="list-style-type: none"> - Word - Excel - The Windows clipboard - A text file
WD External file	Explains how to handle the files not described in the analysis.
WD External Language	This example presents the use of windows developed with WinDev in an application in C++ (Visual Studio 2005) and MFC.
WD Fax	Enables you to send faxes containing text in RTF format.
WD Graphic objects	Uses the object-oriented programming. This example is an editor of graphic objects. You can: <ul style="list-style-type: none"> - move, resize the objects - perform selections with the lasso or with the mouse - cut/copy/paste the objects - delete/duplicate the objects - save the created documents and print them.
WD Graphic Popup	Several software for image editing enable you to select a shape to draw and to choose a thickness for this shape via an advanced graphic popup combo box. This type of graphic popup combo box offers a real-time preview of the shape with the selected thickness. The example explains how this type of interface can be included in a WinDev application.
WD GraphicDrawing	This example presents the use of a drawing to generate an interactive interface. It is used to display objects in image format, to resize them, to move them, to distort them.
WD Grouping task lists	Enables you to group all the tasks distributed among several projects.
WD Handles	Enables you to manage the sizing handles or the moving handles of a graphic element.

<p>WD Handling shapes</p>	<p>Used to:</p> <ul style="list-style-type: none"> - Draw rectangles, circles and diamonds - Move the created elements - Select the color of the elements to draw - Capture the double-click performed on an element to find out its characteristics - Call a popup menu on an element - Select colors by clicking a shape
<p>WD Handling the Organizer control</p>	<p>Example for using the Organizer control. The following features are used:</p> <ul style="list-style-type: none"> - Creating an appointment, - Saving an organizer, - Restoring an organizer
<p>WD Highlight</p>	<p>Used to highlight the control currently edited by changing its style. 3 types of highlighting methods are available:</p> <ul style="list-style-type: none"> - Displaying a border around the control, - Changing the background color, - Changing the font.
<p>WD Hotkey</p>	<p>Used to define your own keyboard shortcuts.</p>
<p>WD HSL</p>	<p>Enables you to define a color via its HSL components (Hue, Saturation, Lightness).</p>
<p>WD HTML Export</p>	<p>Manages the external files. Also explains how to handle the files in HTML format.</p>
<p>WD Importing HTML pages</p>	<p>Saves an HTML page found on a Web site. This page is analyzed in order to import all its dependencies (images, applets, etc.).</p>
<p>WD Input checks</p>	<p>Checks the data entered by the user (name, company, city, zip code, email address etc.) in a form window.</p>
<p>WD JAVA calls</p>	<p>This example presents the call to the methods of the classes of a JAR archive via the WLanguage function named JavaExecuteFunction.</p> <p>This example also explains how to fill a memory table (same operating mode in Java and in Windows)</p>
<p>WD Java Chart Designer</p>	<p>This example explains how to create charts via the drawing functions.</p> <p>Three types of charts are presented: histogram, scatter, line.</p>
<p>WD Java Client</p>	<p>This example is a Java application developed in WinDev. This example presents:</p> <ul style="list-style-type: none"> - the Hxxx commands supported in Java on HyperFileSQL (the Hxxx commands are also available in Java on MySQL and SQL Server) - the controls with file links

WD Java Controls	<p>Presents some controls supported in Java:</p> <ul style="list-style-type: none"> - the "progress bar" control - the "splitter" control - the "listview" control - the preset actions - the Vista style via the Vista skin template supplied with WinDev.
WD Java Drawing	<p>Presents the use of the drawing functions of WLanguage for the generation of a Java application.</p> <p>The following topics are presented in this example:</p> <ul style="list-style-type: none"> - the creation of graphic shapes - the creation of text in the graphic areas
WD Java HSL	<p>Enables you to define a color via its HSL components (Hue, Saturation, Lightness) and to generate a Java archive.</p>
WD Java Sockets	<p>This example presents the use of the "socket" functions and it is used to manage a .jar.</p>
WD Java Threads	<p>This Java educational example explains how to use the threads. Critical section, semaphore, all the main points for multi-thread development are presented here.</p>
WD Java TreeView Table	<p>Used to develop Java applications without having to know Java.</p>
WD Java XML	<p>This example presents the use of the XML functions in a WinDev Java project.</p>
WD JavaMail	<p>Enables you to create an applet that can be used in an Internet page.</p> <p>This applet is used to read and send emails.</p>
WD Jump Lists	<p>JumpList in Windows 7</p> <p>A JumpList is a menu accessible from the Start menu or from the application icon in the taskbar.</p>
WD Keystrokes	<p>Enables you to detect and identify the keyboard key that was pressed.</p>
WD Lasso	<p>Uses a selection lasso in an image control to perform a zoom in a preview control.</p>
WD License	<p>Used to display an important text to a user. It can be the text of the license displayed to the user during the first execution of a program for example. The text must be entirely read, the choice for validating the window is enabled only when the display control is entirely scrolled.</p>
WD Lock	<p>Explains how to handle the functions used to automatically lock an application.</p>
WD Mailshot	<p>Used to send a "mailing" by Internet.</p>

WD Managing the Java errors	This educational example explains the operating mode and the implementation of the mechanism for managing the WinDev errors.
WD MIME Extraction	Enables you to extract the attachments found in an email.
WD Multi-windowing	Used to understand the principle of multi-windowing in WinDev.
WD Multitouch	This example uses an internal component that gives you the ability to use the Windows Touch features in a WinDev application.
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 Number in words	Used to convert a numeric value into letters. Then, this conversion is displayed in a "Check letter" report.
WD OLE DB	Used to access external databases via OLE DB.
WD OPC	Shows the implementation of the OPC protocol in WinDev.
WD OpenGL	Used to interface OpenGL and WinDev. OpenGL is a software interface used to draw three-dimensional objects (3D).
WD Organization Chart	This example is used to create organization charts.
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 other elements found on the screen. This example explains how to create this type of interface in WinDev.
WD PC Info	Used to find out the language of the computer on which the application is run.
WD PDF image	This example uses the principle of report with programmed data source to propose the creation of a PDF document that groups a selection of images.
WD PDF Viewer	Presents the management of a PDF in an Image control of a window.
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 Phonetic	Used to perform a phonetic search on the name of a person. This search is used to find a record whose exact spelling is not known.

WD PowerPoint Generation	<p>This example explains how to generate a PowerPoint presentation from:</p> <ul style="list-style-type: none"> - the data found in a WinDev application such as the data found in tables charts .. - and from a base PowerPoint document (.pptx).
WD Print Parameters	<p>Illustrates the use of iParameter.</p>
WD Printing TreeViews	<p>Used to print a TreeView. This example uses a "TreePrint" class that can be easily re-used in any treeview control</p>
WD Process	<p>This example presents the use of the functions for managing the memory processes of WinDev (list of processes, memory used, lists of DLLs used, localization).</p>
WD Progress bars	<p>Used to create different types of progress bars.</p>
WD Pure Menu	<p>This example proposes a menu that can be customized via a control template.</p>
WD Queries	<p>Enables you to perform calculations, to pass parameters or to print the result of a SQL query on a HyperFileSQL data file.</p>
WD Reflexive link	<p>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.</p>
WD Regular expressions	<p>Presents two methods for using the regular expressions:</p> <ul style="list-style-type: none"> - checking the input format - extracting the different elements respecting the input format
WD RemoteControl	<p>Enables you to implement the remote control of a WinDev application.</p>
WD Reports	<p>Presents the different data sources of a report:</p> <ul style="list-style-type: none"> - Report based on a memory table - Report based on a memory zone - Report based on a view - Report based on a query - Label report with bar codes - Report based on a text file (automatic or programmed) - Report while passing parameters
WD RTF Conversion	<p>Explains how to transform a text in RTF format (with formatting) into a "rough text" (to perform processes) or into HTML format (to be used in an email or in an Internet site for example)</p>
WD SAP	<p>Used to access your SAP data from your WinDev applications via the SAP functions.</p>

WD Screen saver	Used to manage a screen saver via the management of timers and Windows events.
WD Search by composite key	Presents the use of filters and the use of composite keys in order to optimize the search times on a HyperFileSQL data file.
WD Search Control	This example explains how to manage an edit control used to perform "Google" FullText searches in your database.
WD Sending keys	Illustrates how keyboard sequences can be sent to an application with the WLanguage functions.
WD Server trigger	This example presents the use of the 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 SNMP	This example displays information about the specified network computer: IP address of the computer, its MAC address, its system ... This example interrogates the computer to get this information via the SNMP functions of WLanguage.
WD Soap Client	This example presents the use of the SOAP functions available in WLanguage
WD SOAP Server	Presents the use of the SOAP functions available in WLanguage. Presents two main topics: - Running a SOAP query - Retrieving the result of a SOAP query
WD Speedometer	A "speedometer" is a progress bar that can look like a revolution counter, a kilometer counter. This educational example presents the circular progress bars.
WD Spelling checker	Enables you to start the spelling checker of Word from a WinDev application. To do so, this program establishes an OLE connection with Word.
WD StaticMap	This example presents the interaction between a WinDev application and the Google Maps API service in order to display a geographical, political or road map of a given location.
WD Statistics	Presents a set of statistical calculations.
WD Stock chart	Presents the features of the chart designer for the stock charts. WinDev manages the following types of stock charts without programming: BarCharts, Japanese Candlestick, Line, Histogram, Min/Max.

WD Syntactic Coloring	This example proposes a base class used to perform a syntactic coloring on a text.
WD SystemAPIs	Presents the use of the Windows APIs.
WD Table Breaks	Explains how to use the "breaks" in a table.
WD Tables and Combos	Enables you to select a product in a combo box and to display the information in a table. The description of each product is in RTF format and it is illustrated by an image.
WD Text Files	Manages the external files (creation, deletion, modification of the content of an external file).
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 Thread Pool	Implements a thread pool. To illustrate this feature, the threads are materialized by soap bubbles that rise inside the window. When a bubble touches a side of the window, it is destroyed along with the thread that was associated with it.
WD Timeline	Proposes an internal component used to handle a 'Timeline'. A timeline is used to display in a linear way a set of events that occurred over a given period of time.
WD Trace Socket	Used to trace a communication via socket, on a given address, on a given port.
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 TreeView Duplicates	This example is used to handle the duplicates in a treeview via the WLanguage functions named TreeID and TreeSelect.
WD TreeView Table	The treeview tables are used to associate a treeview with a WinDev table. This example presents the use of the TreeView Table control.
WD TreeView	Presents the use of the TreeView controls.

WD Trigger	Used to enter the expenses for the contributors of a company. Two modes are available: - View with ability to perform some modifications (read/write) - Read-only The triggers are used to check the access to the files according to the selected mode.
WD Universal Replication	Explains how to synchronize the data of different sites by using the universal replication.
WD User groupware	Presents the operating mode of the User Groupware.
WD Using aliases	Used to handle the same physical data file via several aliases.
WD Using DotNet classes	Enables you to use the .NET classes in WLanguage. Used to: - analyze a character string, - configure the dialog boxes displayed.
WD Using DOTNET	Enables you 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 Voice commands	A WinDev application already responds to the voice when the user dictates in an edit control or spells out the caption of a button. You also have the ability to program a specific voice command to perform a specific action.
WD Webservice Client	This example presents the use of the WebServices
WD Webservice Server	This example presents the use of the WebServices.
WD Who is locking	Signals to the users of a HyperFileSQL database on a network who is locking a record.
WD Windows Drag and Drop	Proposes an image catalog. The images can be added into the catalog by Drag and Drop from the Windows explorer.
WD WindowsUpdate	This example explains how to manage the updates available on Windows Update.
WD Wizard	Manages a "Wizard" window.
WD XLS Class	This educational example proposes a first approach for handling the advanced types (xlsDocument, xlsColumn, xlsRow and xlsCell).
WD XML	Enables you to import an XML file in HyperFileSQL format and to export some files in XML format (eXtensible Markup Language).

Additional examples will be available for download from our site (www.windev.com).

