



DEVELOP 10 TIMES FASTER



PCSOFT

Remember to visit our site (www.windev.com) on a regular basis to find out 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. See the **online help**.

All product names or other trademarks mentioned in this publication are registered trademarks of their respective owners.

© PC SOFT 2011: This publication may not be reproduced in part or in full without the express consent 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** Read 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" (book + 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 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 from <http://doc.windev.com> or installed with WinDev, allows you to easily find the syntax of a WLanguage function, get help about an interface, ...

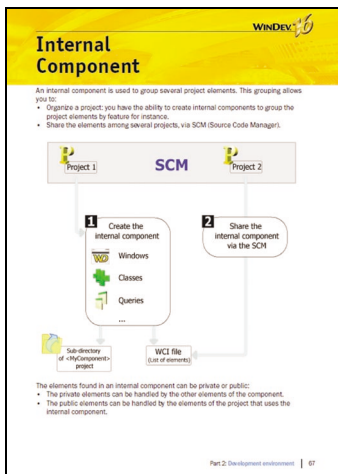
Note: If there is a difference between the guide and the online help, follow the instructions given in the online help.

We hope you enjoy getting started with WINDEV.

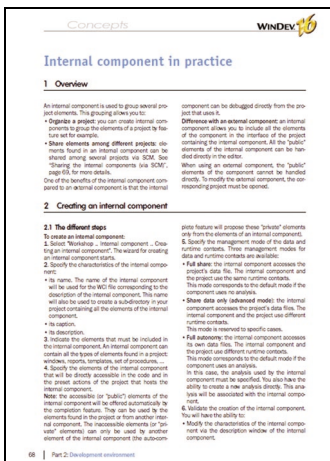
Organization of the guide

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:



Concept page



Implementation page

Table of contents

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 various types of standard controls	33
Control Templates	46
Edit: Reports	47
Report Templates	48

PART 2 - DEVELOPMENT ENVIRONMENT

The WinDev Editors	51
Project Dashboard	55
WinDev/WebDev/WinDev Mobile: 100% compatible	56
Project Configuration	57
Multiple Generation	58
Custom-folders: Organize your project	59
Source Code Manager (SCM)	60
Internal Component	67
External Component	70
Generation modes	77
Control Centers	82
Managing requirements	83
Project Monitoring Center	84
Managing the suggestions and the incidents	85
Managing business rules	86

PART 3 - DATABASES

Analysis: Structure of the database	89
The different types of accessible files	100
HyperFileSQL Classic	102
HyperFileSQL Client/Server	103
HyperFileSQL Cluster	104
HyperFileSQL: the physically created files	105
Associating the controls with the data	106
Queries	108
Embedded Queries	109
The Table/Looper controls	110
Universal Replication	112

3-Tier Architecture 113

PART 4 - APPLICATION DEVELOPMENT: ADVANCED CONCEPTS

RAD / RID 117
 The UML Models 118
 User Groupware 125
 Multilingual Applications 134
 Optimizing an application 139
 Distributing "Reports and Queries" 141
 UMC: User Macro-Code 142
 The various tests available 144
 The debugger 145
 Unit tests / Automatic tests 147
 Unit tests on the executable 148

PART 5 - SETUP

The WinDev Framework 157
 The different modes for installing an application 158
 Monitoring your applications 166

PART 6 - COMMUNICATION

Managing Emails 169
 Transferring files by FTP 171
 Managing sockets 173
 WinDev and telephony 175
 Webservices 176

PART 7 - APPENDICES

Components supplied with WinDev 179
 Available examples with WinDev 183



PART 1

Basic Concepts



DEVELOP 10 TIMES FASTER

PC SOFT

Project and Analysis

Developing a **Windows application** using WinDev relies on two main elements: the **Project** and the **Analysis**.

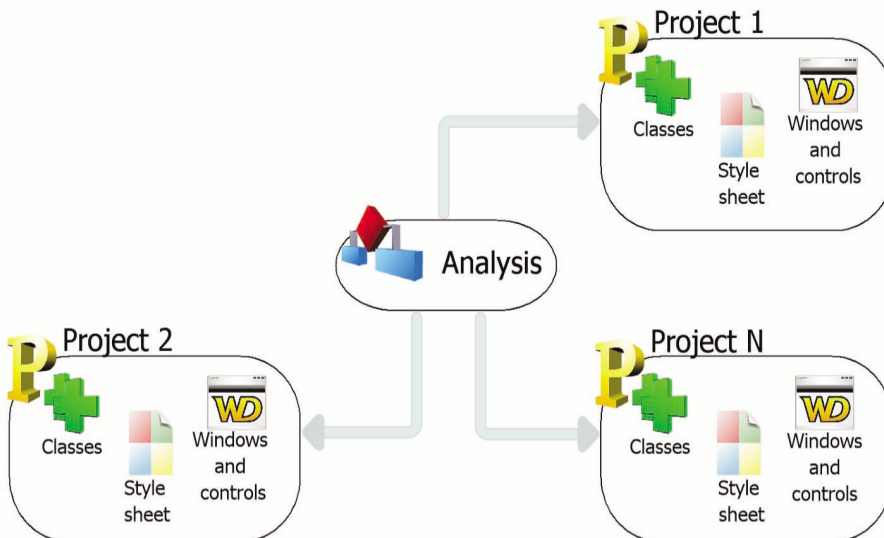
A **WinDev Project** is a set of elements: windows, reports, controls, classes, components, ... whose combination allows you to build a Windows application.

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, you will be asked several questions in order to build the requested project. All the characteristics specified when creating the pro-

ject can be modified later.

Note: The analyses are presented in details in “Analysis: Structure of the database”, page 89.

2 Creating the project

To create a project:

1. Select "File .. Project". The project creation wizard starts.

2. Specify the different options of the project:

- **the name and location.** These options cannot be modified. The project corresponds to a ".WDP" file. All the objects associated with the project will be created in the specified directory.
- **the type of generation.** Several possibilities 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 have the ability to use the developer groupware or the source code manager (SCM) to share the project.
- **the programming charter.** The programming charter is used to automatically add a prefix to the variables, the names of controls, the names of windows, ...
- the style book.

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

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

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

Notes:

- **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 automatically starts.
- 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 vision 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 fire up the most often used project elements and source code, ...

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

To display the dashboard of your project, all you have to do is select "Project .. Project dashboard".

See “Project Dashboard”, page 55 for more details.

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 elements of the project

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

To find out the list of project elements, select

"Project .. List of project elements".

This option is used to:

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

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:

- Analysis/File/Queries documentation: Contains the information about the analysis, 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 the entire 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 or 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 some 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 .. Printout areas". Some 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) in order 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).

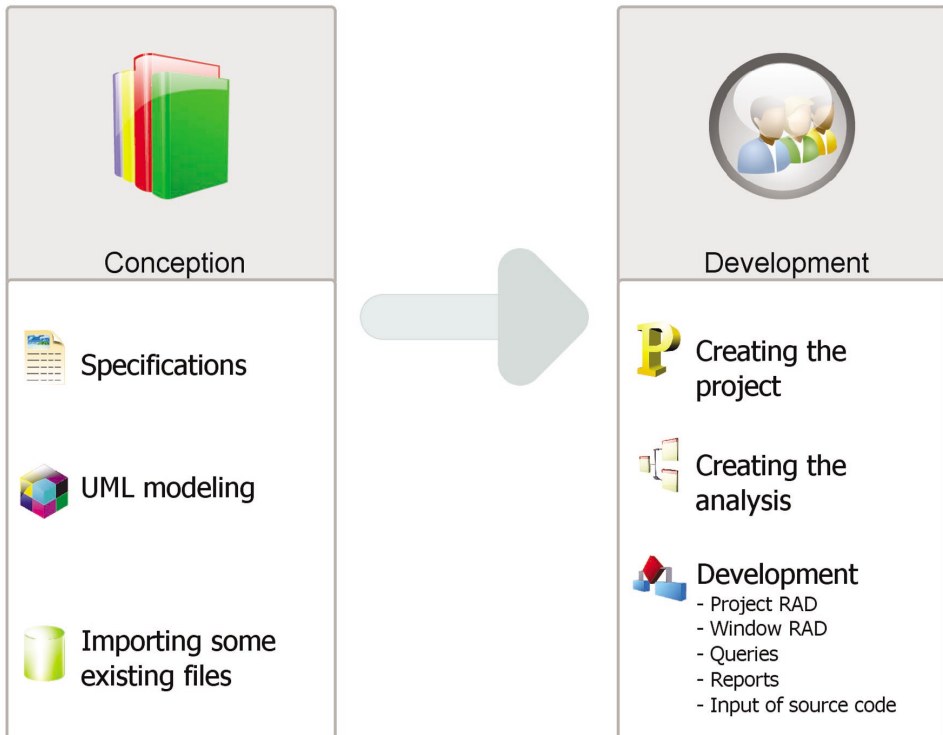
If the print format is modified (from A4 to A3 for example in the properties of the printer), the documentation pages can be adapted to the new format.

To do so:

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

Development cycle of an application

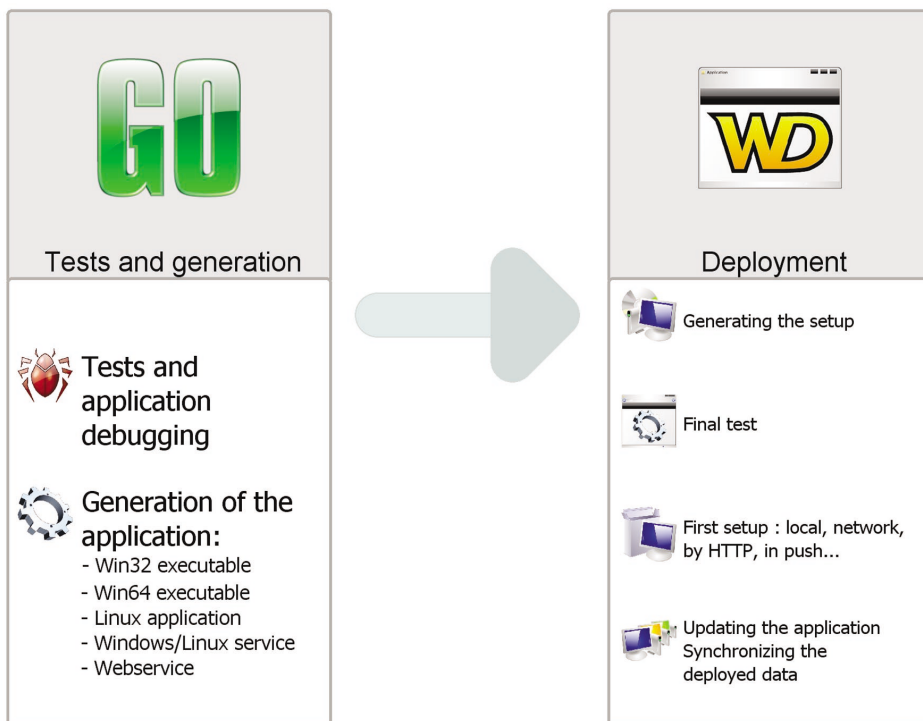
WinDev covers the entire development cycle of an application:



Details of the different steps:

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

Development step: The project and the analysis area created via specific wizards. The development can be performed 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 running automatic tests in order to guarantee the reliability of applications and the non-regression between the development steps.

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

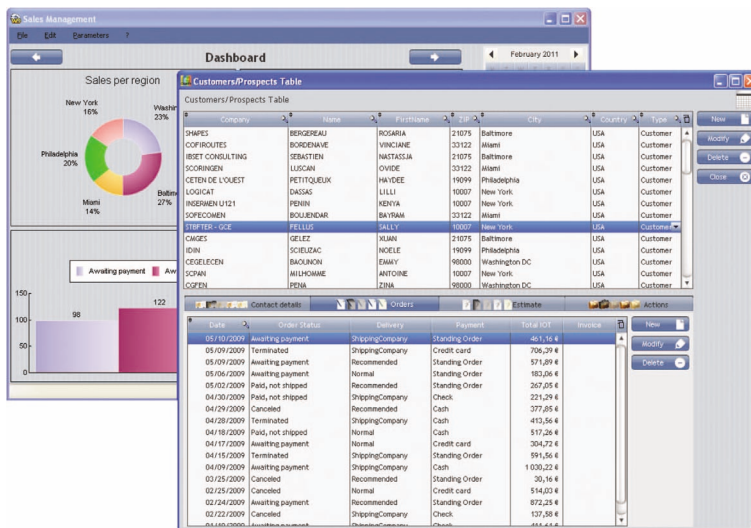
GUI: The windows

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 some 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

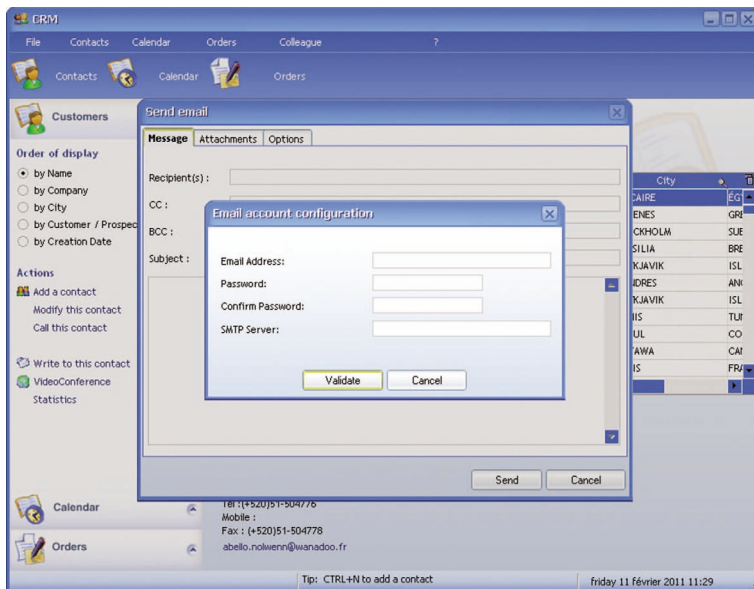
The applications created with WinDev can use several types of windows (and mix each one of these types).

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 open (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 Disable 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:

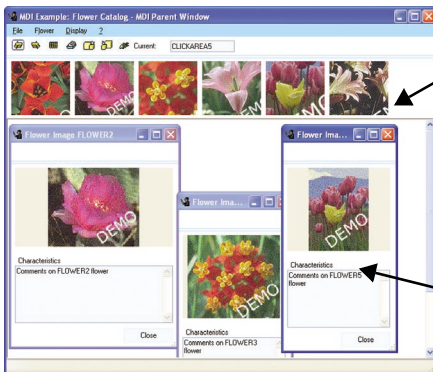
- as tool palettes,
- as dialog boxes whose modification directly affects the main window.



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.

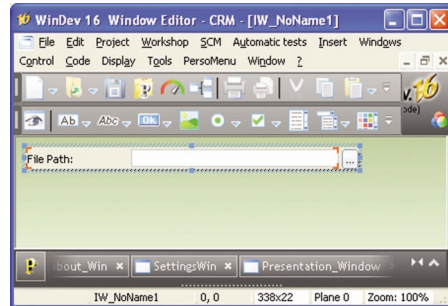
1. Creating an internal window.

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

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

All types of controls can be used in this window.

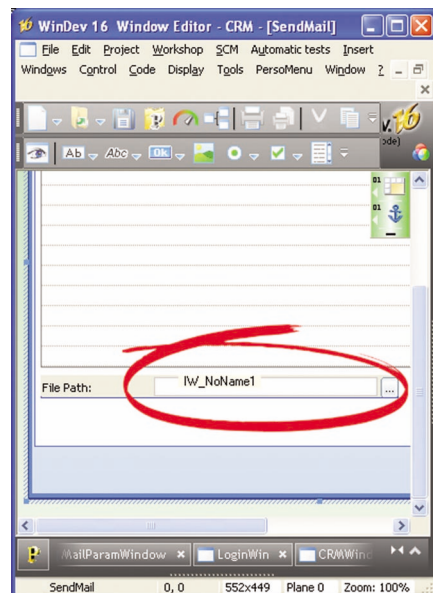
In the example, an internal window is used to select files.



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.

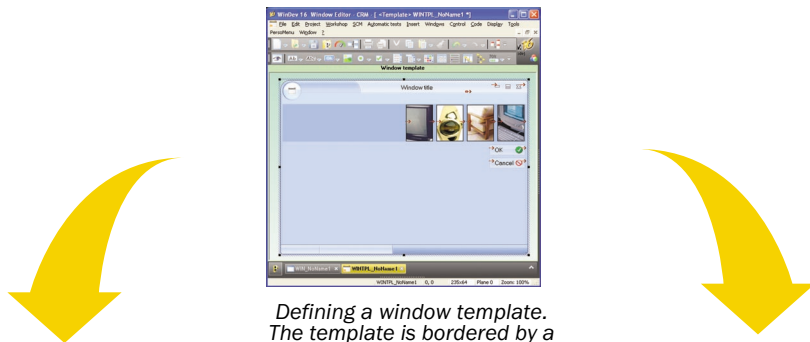


Notes:

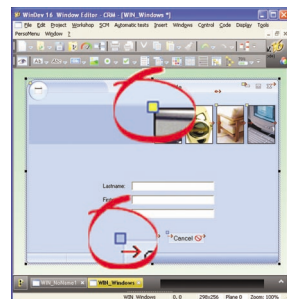
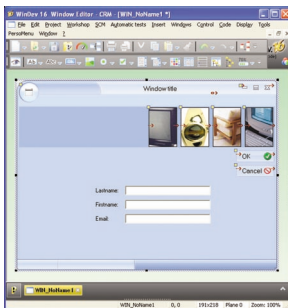
- You can also modify by programming the internal window used in the "Internal window" control.
- Limitations: The host area is rectangular and no overload is possible. To perform overloads, we advise you to use control templates.

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 bordered 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 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.

Windows in practice

The purpose of a window is to display, present or enter some information. This information can come from the data files of an 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 its code or not).
- 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 some controls and modify the characteristics of the window.

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. Select 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.

See "Controls in practice", page 38, in more details about how to handle a window's 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 the disabled windows", page 26
- "Internal window", page 27
- "Window templates", page 27
- "Processes associated with the windows", page 27
- "The menus", page 28
- "The status bar", page 32

Note: the main characteristics of windows are detailed in the online help.

The various types of controls are presented in the "Controls in practice", page 38.

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 lets you perform the following operations on windows:

- Opening a window in the editor: Simply double-click the name of the window displayed in the "Project Explorer" pane.

- Modifying the size of a window.
- Save and copy a window.
- Importing and exporting a window.
- Modifying the navigation order.

See the online help for more details.

3 Main characteristics of a window

3.1 Characteristics

A window can accept all the available types of controls.

A window can:

- be **resizable**: the user can increase or decrease the window's size. Depending on the anchoring settings defined for each control, the position of the controls inside the window can change when the window is resized (see "Anchoring controls", page 43, 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 settings for resizing windows through the 9-image mode. 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 the "Clipped window", page 25, 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, use "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. See "The popup menus", page 28, for more details.
- have a **toolbar**: this toolbar will display the help messages associated with the different controls

of the window. This status bar also lets you 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 the "Multilingual applications in practice", page 135, for more details.
- **turn gray** automatically when it becomes unaccessible. This feature is called DDW, which stands for Dim Disabled Windows. See "Dim the disabled windows", page 26, for more details.
- authorize **drag and drop** operations. For more details about drag and drop and its programming, see the online help.

Note: to force an action on a window, simply define an "Automatic execution" mode button in this window. See "Automatic execution of a button", page 45, for more details.

3.2 Description window

The description window is a window containing tabs, which groups all the configurable settings of the window.

Note: you can also view and/or modify the characteristics of a window in the modifier. See the online help for more details.

3.3 Display the characteristics

To display the description window:

- double click the window's background.
- select "Description" from the popup menu of the window (right mouse click).
- select the window then select "Display .. Selection description".
- select the window and use the [Alt] + [Enter] keyboard shortcut.

3.4 Characteristics by tab

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

See the context-sensitive help of 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 settings for the window's interface:

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

Details tab

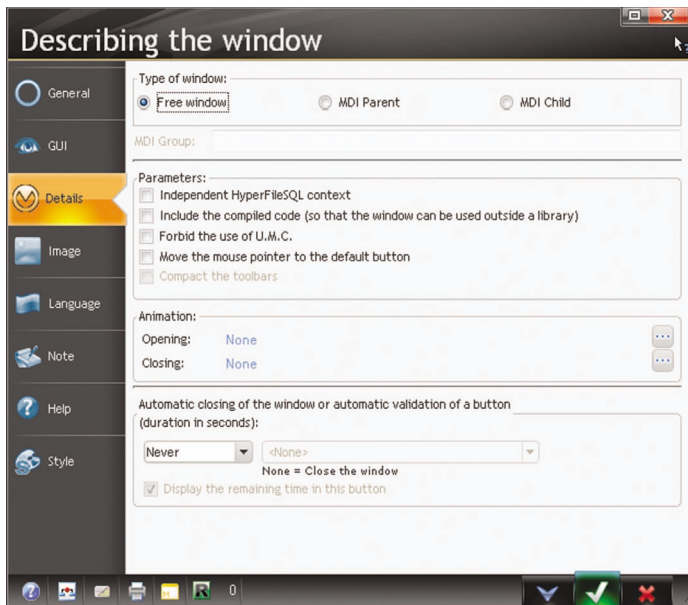
The "Details" tab is used to define the various parameters of the window:

- animation,
- automatic closing.

Image tab

This tab is used to configure:

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



Language tab

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

Notes tab

The "Notes" tab is used to detail the behavior 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 popup menu 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 various 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 window's description window.
- images in your windows. These images are used as background image of the window. Your controls are superimposed over this image. The configuration of the background is performed in the "Image" tab of the window.

Notes:

- If needed, WinDev also offers 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 found in the explorer to the WinDev window.

The options available for the image are:

- Managing multilingual images
- Display mode
- Position
- Transparency
- Symmetry
- 9-image mode
- Window clipping based on the background image

Managing multilingual images

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

To define a multilingual background image:

1. Click the icon located to the right of the background image's name.
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 100% from the left corner of the image control.
- **Centered:** the image is displayed 100% and is centered in the image control.
- **Stretched:** the image is deformed so it occupies the entire space of the image control.
- **Tiled:** the image is displayed 100% and repeated so it occupies the entire space of the control.
- **Homothetic:** the image is proportionally enlarged so it fills the entire image control.
- **Homothetic extended:** the image is proportionally enlarged so that its smallest side is entirely displayed in the image control.
- **Homothetic centered:** the image is homothetic and is centered in the image control.
- **Homothetic centered extended:** the image is stretched and centered in the image control.
- **Homothetic without increasing size:** 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 displayed in full.
- **Homothetic centered without increasing size:** the image is centered and homothetic without size increase.

In homothetic mode, "High-Quality Display" enables you to improve the quality of the image when this one is reduced from its initial size.

Position

This parameter enables you 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* property).

Transparency

The transparency options for the background image are as follows:

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

Transparency: none Transparent magenta



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

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* property).

Initial image:



The options are as follows:

Symmetry vertical



Symmetry horizontal



Symmetry horizontal and vertical



9-image mode

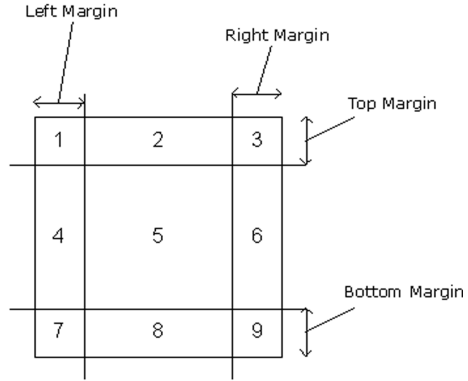
If this option is selected, the 9-image mode will be used on the window's background image.

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

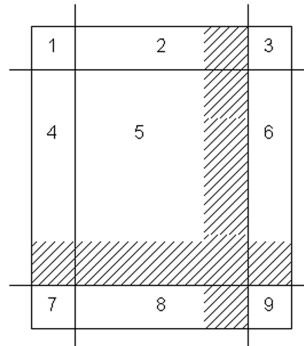
For each area, you can define whether the section of the image corresponding to the area must be displayed or not (check the name of the area to display it). If the image section is not displayed, it will be

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 enables you to manage the operating mode of each area when the window is resized.

For each area:

- by repeating the image that makes up the area,
- by stretching the image that makes up the area.

4.3 Clipped window

WinDev lets you clip windows. The windows will have custom shapes (window in the shape of 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, etc.) in order to avoid the display effects during the preview or when opening the clipped window.

The configuration of the clipped window is done in the "Image" tab of the window's description window.

5 Opening a "popup" window

5.1 Overview

WinDev lets you open "popup" type windows.

A popup window is a small-size window that opens on top of the current window. Usually this window allows the end user to make 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).

For example:



Note: here are the characteristics of a popup window:

- opened by default under the control that called it (if there is not enough space under the control, can open to the left or above of the control). The opening position can be changed from your program.
- 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 defined in the style).

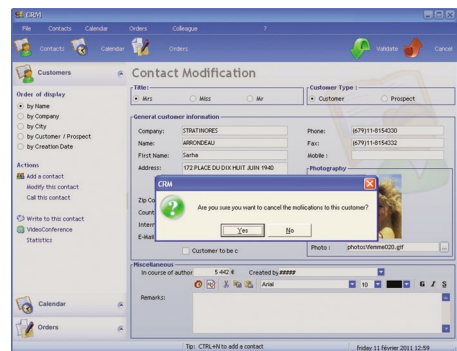
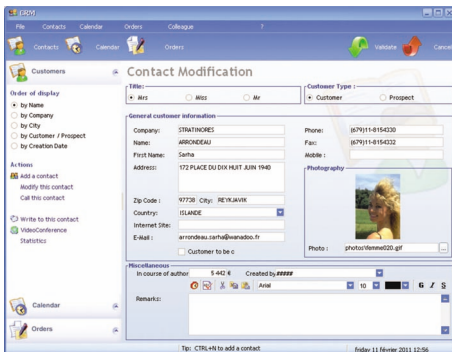
6 Dim the disabled windows

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 can be customized at each window's level.

You can also affect the DDW feature by programming.

See the online help for more details.

7 Internal window

Internal windows are used to dynamically share some part 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 some 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, some 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):

- **Declaration of globals:**
Declaring the variables global to the window
- **Initialization:**
Run when the window is opened, before focus is set. The initialization process of the window controls is run after this code.
- **Closing:**
Run when the window is closed. Note: resuming input in one of the window's controls in this code prevents the window from closing.
- **Gain of focus:**
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** has been executed from another window of the application.

This process will not be run:

- when opening the first window of the project,
- when calling **OpenChild** or **OpenSister** from the current window,
- if **Close** is called in the window's initialization code,
- if the user clicks a window from another program then re-clicks inside the initial window.

- **Loss of focus:** Executed when the window loses the focus when pointing 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** has been executed 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 window's initialization code,
- if the user clicks a window from another program then re-clicks inside 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 one of the window controls is modified (input performed by the user for instance)

- **Assigning the ..Value property (internal window only)**
Run when using the Value property in assignment on the Internal Window control.
- **Retrieving the ..Value property (internal window only)**
Run when using the Value property in read 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 The popup menus

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

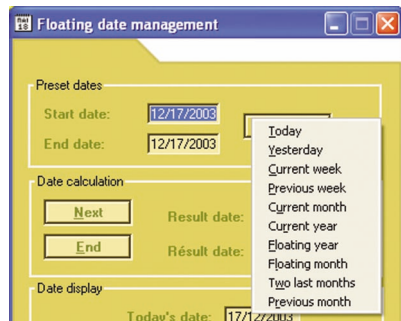
WinDev enables you to manage two types of popup menus:

- **the system popup menus** managed 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 29, for more details.
- **the custom popup menus** managed by WinDev. These menus are created in the window editor and they can be handled by programming.

WinDev 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.

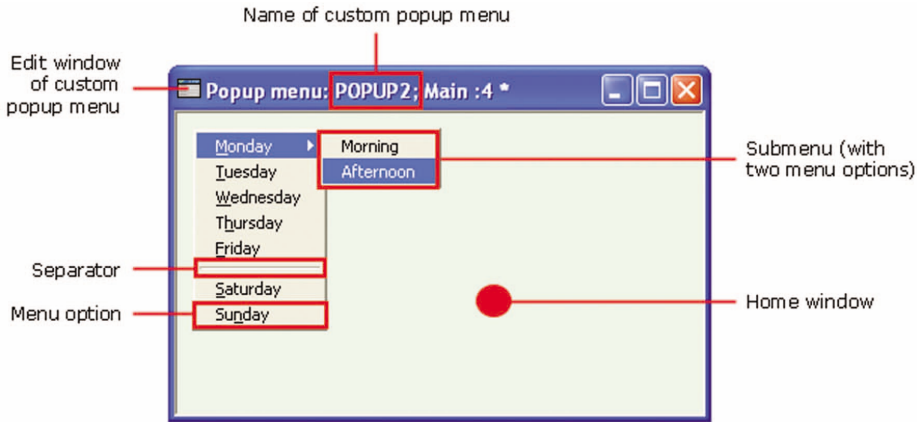
For example



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 menus, let's take a look at the vocabulary linked to

this edit window:



10.2 Available features

WinDev enables you to:

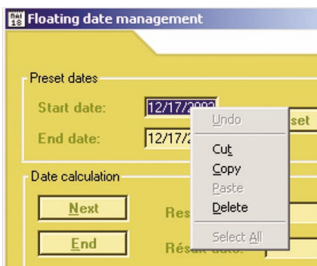
- Create a custom popup menu
- Associate a custom popup menu with one or more elements (window or controls)
- Edit a custom popup menu
- Display the description window of a custom popup menu
- Delete a custom popup menu

See the online help for more details.

10.3 Default system popup menu

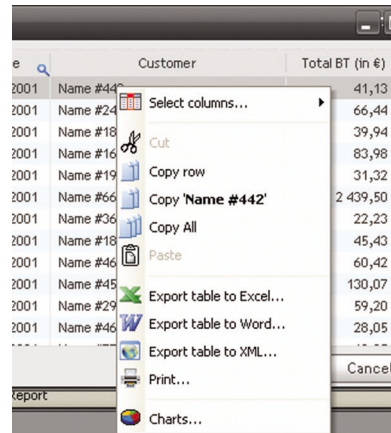
The following types of controls have a default system popup menu (displayed if no custom popup menu is associated with the control):

- an edit control:



See the online help for more details about this menu's translation.

- a table:



See the online help for more details about this menu's translation.

- an ActiveX control:



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

- an HTML control:



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

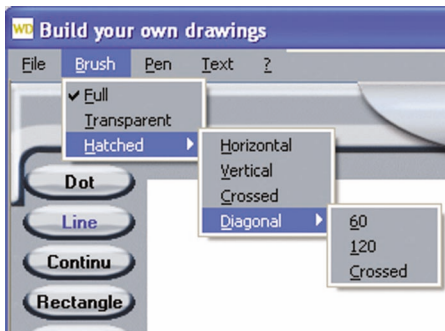
This default popup menu can be disabled from the editor as well as by programming.

11 Main menus

11.1 Overview

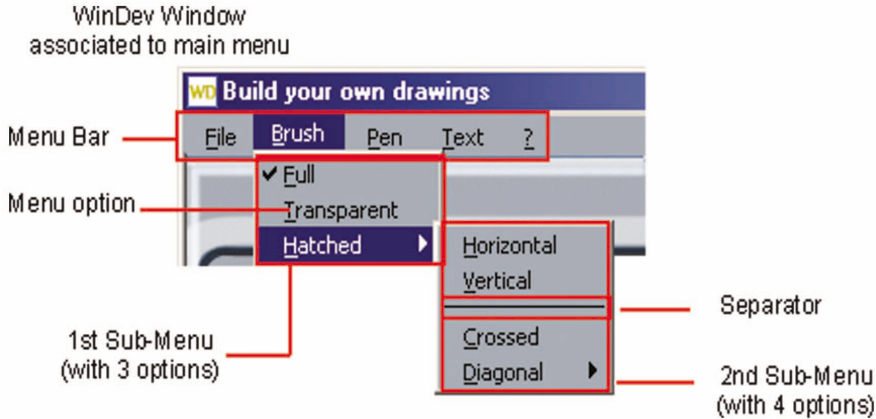
A main menu is necessarily associated with a window. This menu is displayed below the title bar of the window in WinDev. In most cases, the applications contain a menu window. This window only contains a main menu and it is used to access the other windows of the application.

For example:



11.2 Vocabulary associated with the main menus

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



11.3 Available features

WinDev enables you to:

- Creating a main menu
- Editing a main menu
- Deleting a main menu.

See the online help for more details.

Special case: Utility menu

WinDev enables you to automatically insert a "?" menu in 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 some 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 ("File .. Open").
 - select one of the menu options. The options and the sub-options are automatically displayed.
- a window 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 can also associate some WLanguage code with a menu option. Only the menu options without sub-menu can start a WLanguage process.

See the online help for more details.

12.3 Handling the menu options by programming

In the editor, menu options are the same regardless 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 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 **message with information on 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 user selects the control.
- a **special information message**. This is displayed by programming with the **Message** function.
- a **progress bar**. This progress bar is displayed by programming with the **Gauge** function.

To display several pieces of information at the same time, you can divide the status bar into several cells ("Insert a cell" option in the status bar's popup menu).

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 for each cell the information to display.

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 Caret).
- The status of the [Caps Lock] key, indicating whether the Caps Lock key is on or off.
- The status of the [Insert] key, indicating whether the [Insert] key is on or off.
- 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 instance).
- Update process: the "Whenever updated" process is automatically associated with the status bar's cell. This process is run on a regular basis. This process can contain the elements to display in the cell for instance.

13.3 Characteristics of a status bar cell

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





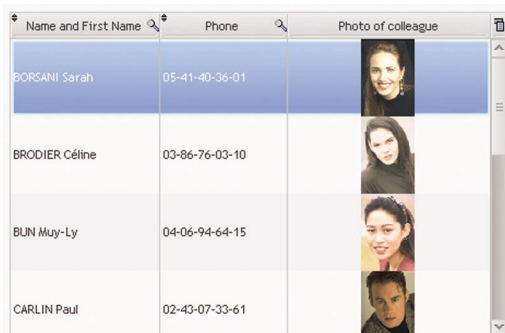



- The position of the cell in the status bar.
- The background image of the cell.
- The background color of the cell.
- The width of the cell.
- Whether the cell is in edit.
- The anchor of the cell (its operating mode when the window is resized).








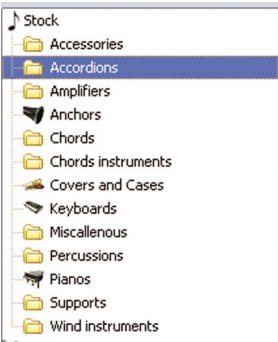


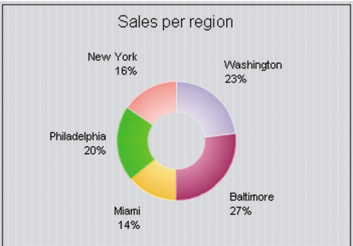

All these characteristics can be used in your programs through the properties available for status bar cells.



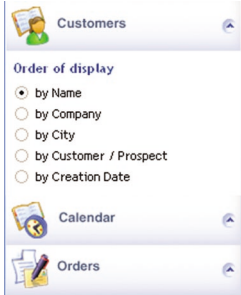



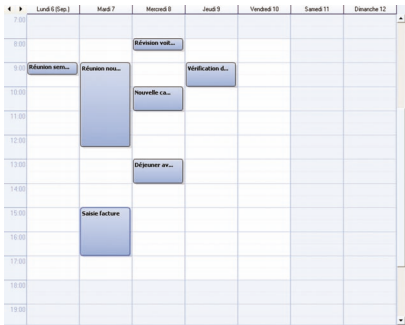

The various types of standard controls

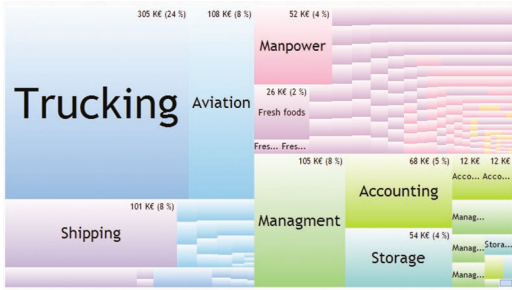

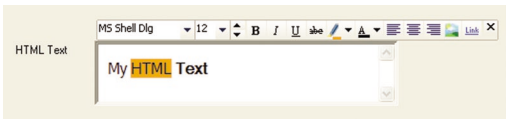




WinDev offers you a large number of controls that come with the product. Here is how to use them based on the operation you want to perform:

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

<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 file in a table (list of customers, order details, ...)</p> 	 <p>Browsing table or memory</p>
<p>Repeat some 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 some 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 printout, ...)</p> 	 Button
<p>Display some information according to a hierarchical system (directory content)</p> 	 TreeView  Treeview table
<p>Display a bar chart, a line chart, a pie chart, ...</p> 	 Graph

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

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

Controls in practice

WinDev offers a large number of controls that can easily be included in your windows. These controls can be created using the "Control" toolbar, or using 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 controls of a window

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,
- Schedule,
- Table,
- Treeview table,
- TreeView,
- Progress bar,
- Looper,
- Scrollbar,
- Slider,
- Spin,
- Tab,
- Window control internal,
- Toolbar,
- Splitter,
- Graph,
- 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 underneath the mouse cursor.
3. Click inside the window where the control will be created. 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 file item, EXCEPT:

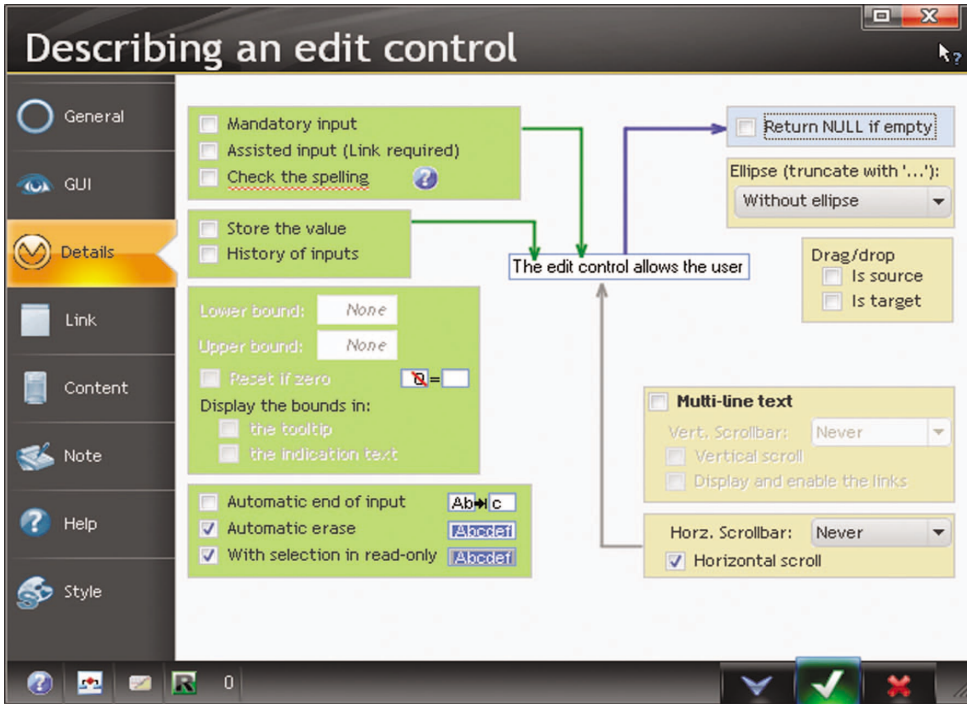
- button
- shape
- tab
- toolbar
- Xaml
- Carousel
- Cube
- HTML control

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

2.1 Display 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 .. Selection description".
- select the control and press [Alt]+[Enter].
- select "Display .. Options .. Modify the options" ([F12] key), 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 context-sensitive help of 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 for the control (caption, input mask, ...).

GUI tab

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

- Control's initial state when opening the window
- Visibility of the control
- Control's size
- Alignment, ...

Details tab

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

- input settings
- Drag and Drop, ...

The content of this tab varies a lot based on the current control's type.

Link tab

This tab is used to select the item 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 "Note" tab is used to describe the detailed behavior 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 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 on demand), if a help file has been defined for the window
- as a help page. The *WHelp* function will need to be used to display the help page.

Style tab

The "Style" tab is used to define the style of the 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 control handles

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* are used to 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.

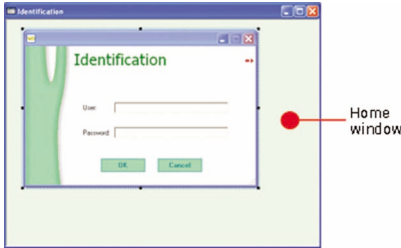
- The *gray handles* indicate the first selected control during a multiple selection. This control will be the reference control.

3.2 Available operations

The window editor enables you to:

- Selecting one or more controls (with the lasso for example)

- Grouping selected controls
- Moving a control
- Modifying the caption of a control
- Displaying an advanced tooltip when a control is hovered by the mouse cursor, containing: the control's name, its position, its size, its anchoring mode, its initial state.
- Using 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 enables you to use the content of the control in programming (like a variable).
- display new controls in a window when resizing this window.

4 Control alignment

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

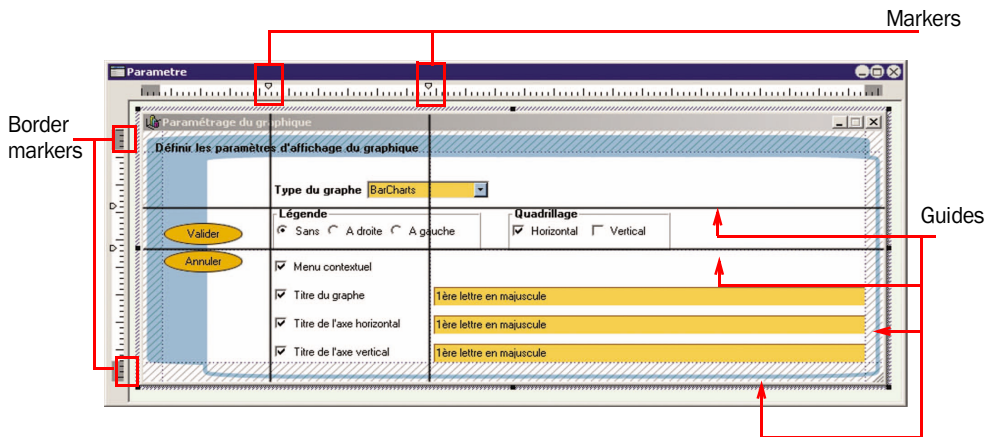
- the rulers.
- the grid.
- the real-time interface checker (automatically proposed when positioning a control).
- the advanced interface checker.
- the alignment options.
- the customizable zoom.

4.1 The rulers

In the window editor, you can display alignment rulers.

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

For instance, in the case of a window:



Displaying the rulers

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

- guides, allowing you to align and organize the controls inside the windows.
- border guides, used to define a margin of the same 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 control you are moving 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 enables you to apply presentation rules to the current window or to the selected controls according to the Windows standard: control alignment, standardization of the button size, ...

4.5 The alignment options

To align several controls, WinDev proposes:

- a preset alignment (available via the drop-down menu "Control .. Alignment..." or in the toolbar for control alignment).
- a custom alignment ("Control .. Alignment .. Custom alignment"). The custom alignment enables you to use some 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 on 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 the size of controls, the reference control can be:
 - the first selected control (the one with the white handles)
 - the last selected control
 - the largest selected control
 - the control 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 operations of click and double-click on the controls.
- the options for control selection.
- the position of the window at opening.

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

"Control .. Edit options .. Visible" lets you make a control visible/invisible in the window editor.

5.2 Locking/unlocking a control

"Control .. Edit options .. Locked" lets you lock/unlock a control.

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

All other operations are possible on a locked control.

5.3 Framing or not a control

"Control .. Edit options .. Frame" lets you frame a control or not.

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

6 Anchoring controls

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

The management of control anchoring includes two parameters:

- **control anchoring:** this parameter lets you define the changes to apply to the control based on the size changes of the window. 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:** this parameter is used to manage the percentage by which the control is moved or stretched. This parameter enables you 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

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 "Detail" tab of the description window of the control ("Display .. Selection description").
 - via "Anchor" from the popup menu (right mouse click) of the control.
3. In this screen, choose the type of anchor 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 is enlarged. To make the control move slower or faster, use the anchor rate to the right.



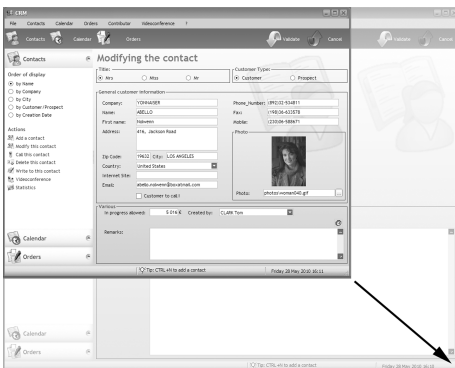
Centered in width: The control remains centered in width in the window regardless of the window width.



Height: The control stretches to the bottom when the height of the window is increased. 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 to the right and to the bottom when the window is enlarged. 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.



the size of anchored controls is automatically modified.



Height and Right: The control stretches to the bottom and moves to the right when the window 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 available only 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 to 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 is enlarged. 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 width. 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.



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

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

5. Validate.

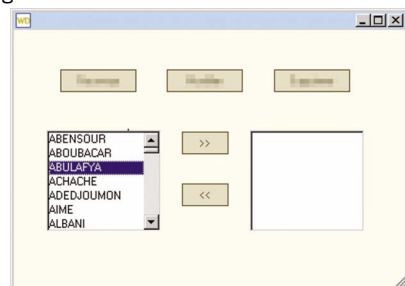
If the option "Display .. Option .. Show anchors" is enabled, the anchor signs automatically come up in the control (red arrows).

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

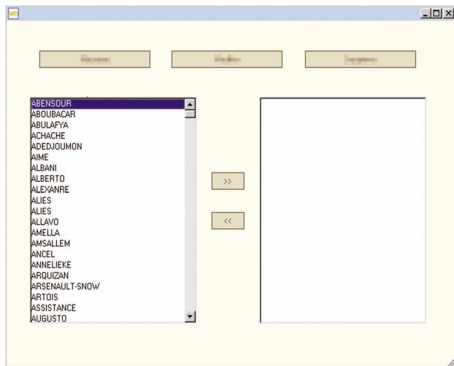
6.2 Example for configuring the control anchors

Here is a detailed example, illustrating the anchoring possibilities for controls in windows.


Original window:




Larger window:



To obtain this result:

1. Select the [<<] and [>>] buttons: these controls must be centered by height and width ().

2. Select both lists: these controls must be anchored to the right, in height and in width (options "to the right", "in height" and "in width" selected).

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 upon a window, define a button in the window as "Automatic Execution". The principle is easy, if no button has been used in the window after a given period of time, the process of the "Automatic Execution" button is automatically run. At run time, the amount of time is decremented automatically. When it reaches zero, the button's code is automatically executed.

7.1 Implementation

To automatically run a window button:

1. Define which button your window must run automatically.
2. In the window description ("Details" tab of the description window), specify:
 - the delay before automatically executing 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 that an application's button be run automatically after a given period of time.

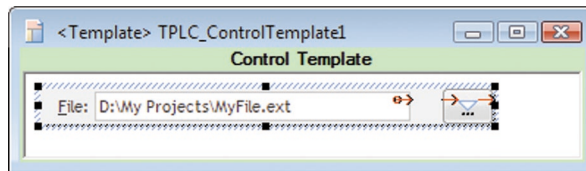
Control Templates

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

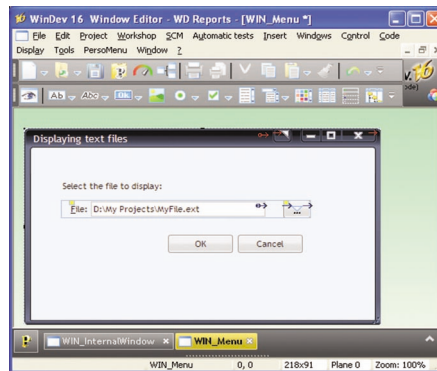
The modifications made to a control template are automatically applied to all the windows that use this template.

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 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: 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, ...

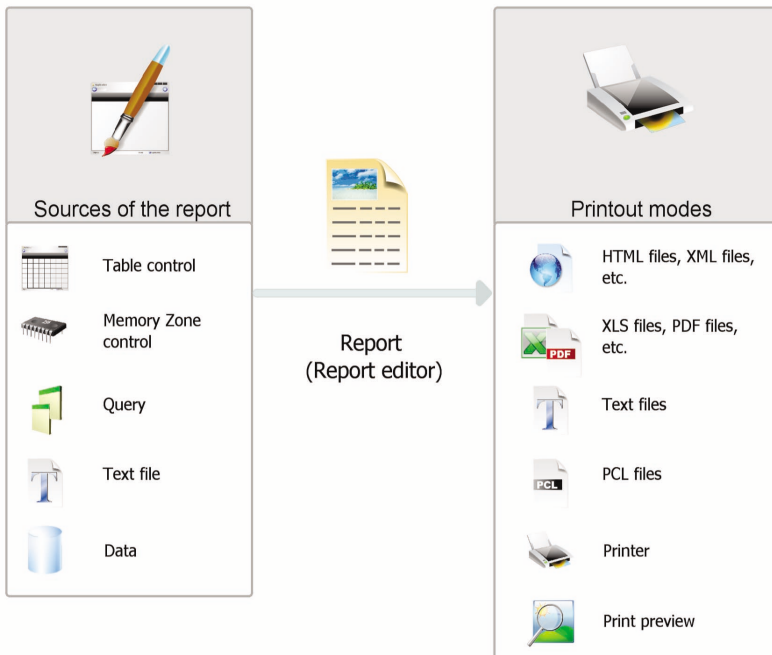
A report can be used to summarize and synthesize data.

You can:

- group 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:

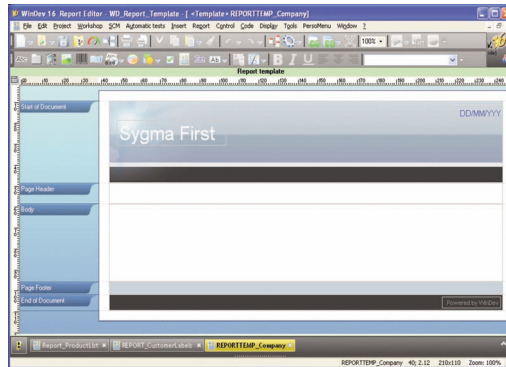
- the data to print comes from a data source (file described in an analysis, HyperFileSQL view, query, memory zone or text file).
- the report groups, sorts and organizes the data.
- the report can be displayed on the screen, saved in an HTML file, a RTF file, ... or printed.



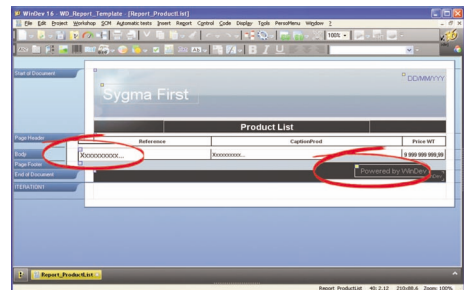
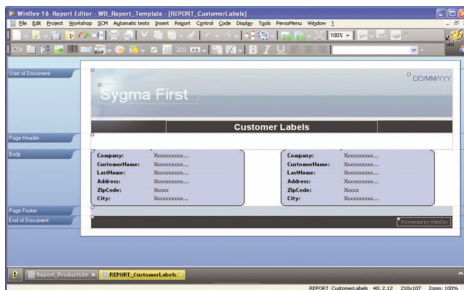
Report Templates

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

The report templates allow you to easily standardize the layout of your reports.



Defining a report template in the report editor.



Using the template in different reports.

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

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

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.



PART 2

Development Environment



DEVELOP 10 TIMES FASTER



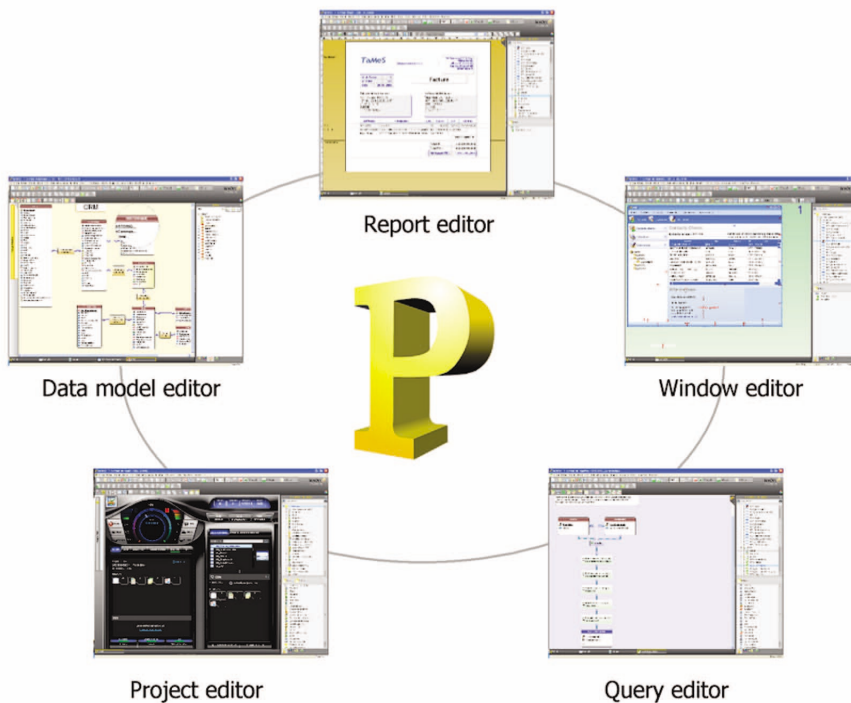
PCSOFT

The WinDev Editors

To handle a project, WinDev offers several integrated editors:

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

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



User-friendly editors in practice

1 Introduction

WinDev is built around a set of editors, adapted to the requirements of the developer, which operate in a single environment:

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

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

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

2 The various editors

2.1 Project editor

The project editor includes two main features:

- it enables you to quickly view all the elements found in a project (windows, reports and queries).
- it enables you 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.
- Project management via a comprehensive dashboard.
- Creation and visualization of 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 main 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 used 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 in practice", page 119, for more details about building a UML model.

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 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.

See "Report and query guide" for more details.

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 query creation 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 "Windows in practice", page 21, for more details about creating windows.

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 the windows and controls.
- "Drag and Drop" allowing you to copy, paste or move the controls from one window to another one.
- Skin templates of windows and preset control styles.
- Presence of several icon catalogs allowing you to associate 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 some reports to print.

See "Report and query guide" for more details.

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 report creation without writing a single code line.
- "Drag and Drop" allowing you to copy, paste or move some 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 as report background.
- Creation of multi-column labels.
- Ability to edit a report in HTML format (to publish it on Internet for instance) 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 correction help.
- 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 modification of code history with ability to rollback.
- Ability to open several code windows (to perform code comparison for instance).

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

See the "WLanguage" manual for more details.

2.8 Help editor

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

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

Let's see some characteristics of 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:

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



In Project Manager mode, the dashboard is used to provide 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 requirement management.

WinDev/WebDev/WinDev Mobile: 100% compatible

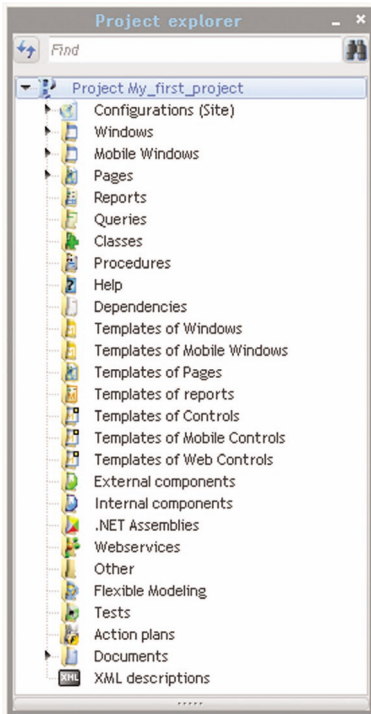
The created projects are often multi-target projects.

For instance, for an ERP system intended for 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 shareable among the WinDev, WebDev and WinDev Mobile projects.

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

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 where it was created, a wizard is displayed, allowing you to create a **project configuration** specific to the product used.

For instance, if a WinDev project is opened by WebDev, you will have the ability to create a project configuration named "WebSite" 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 instance. Clicking a WebDev page from the project editor of WinDev opens the WebDev page (WebDev must be installed on the computer).

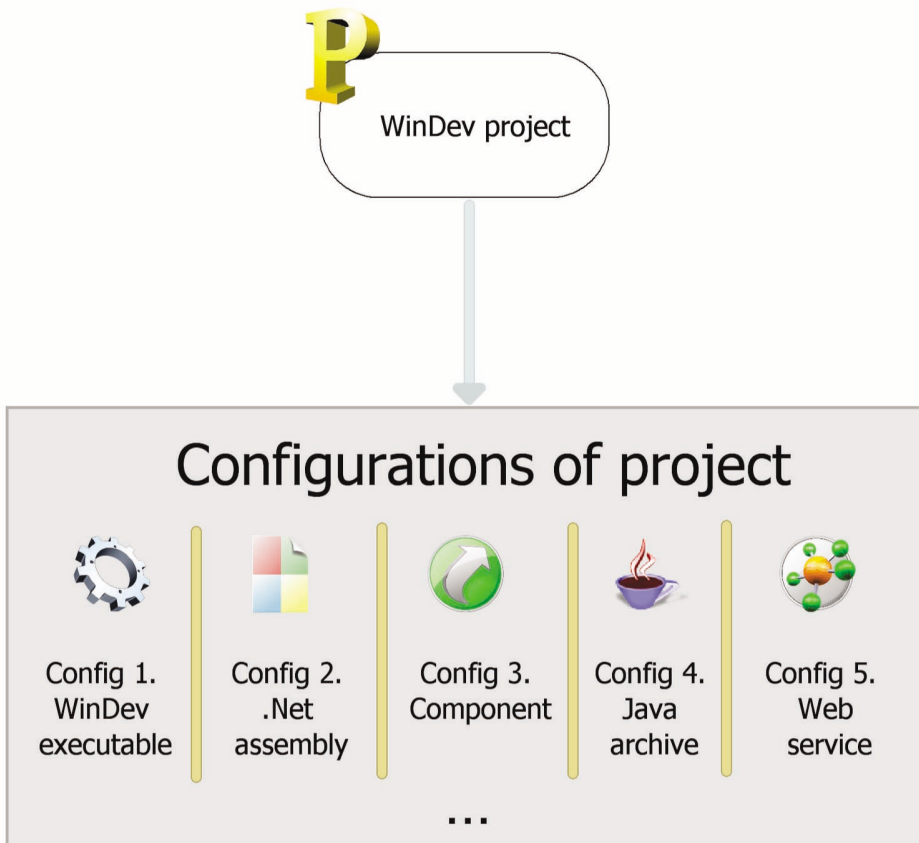
Project Configuration

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

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

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

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



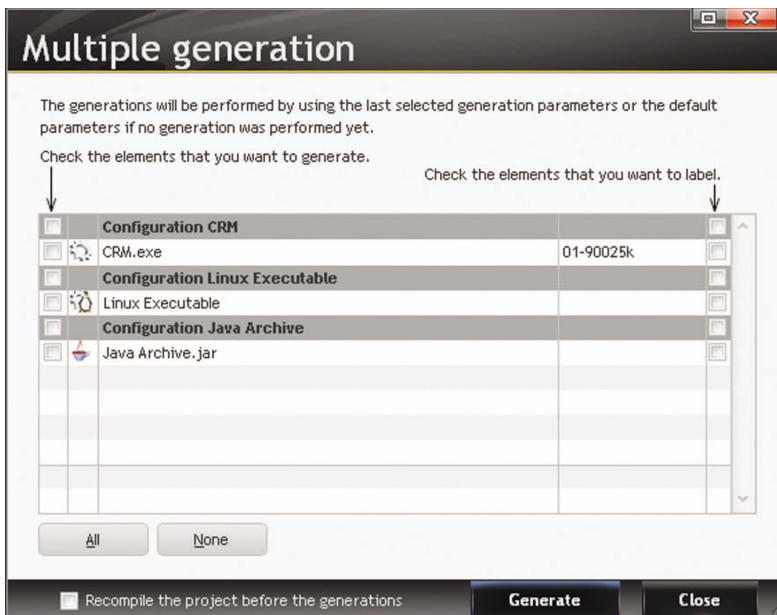
The multiple generation allows you to generate all the project configurations (or some of them) in a single operation.

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 executables, libraries, Webservices, .Net assemblies, ...

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

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



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

Custom-folders: Organize your project

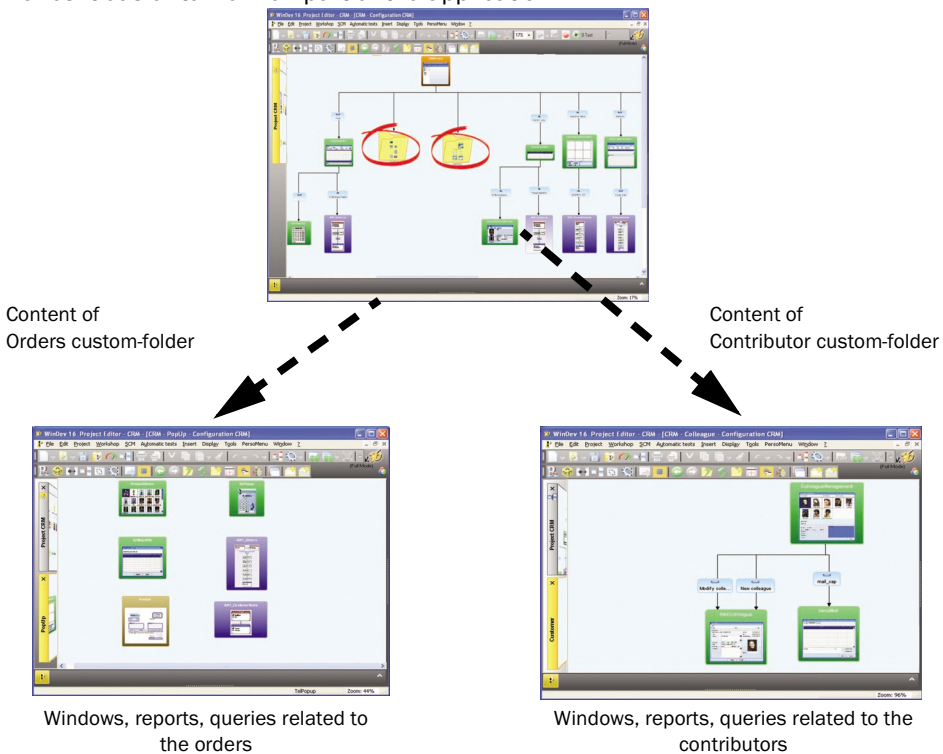
All the elements included in a project are listed in the Project Explorer. By default, the Explorer organizes the elements according to their type: windows, reports, classes...

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

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

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

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



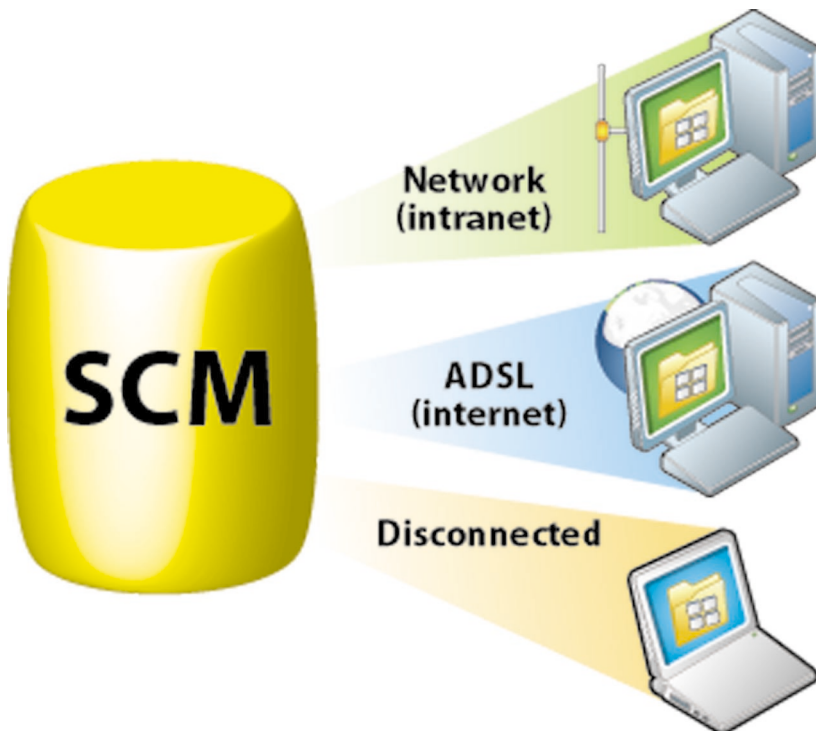
Notes:

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

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.

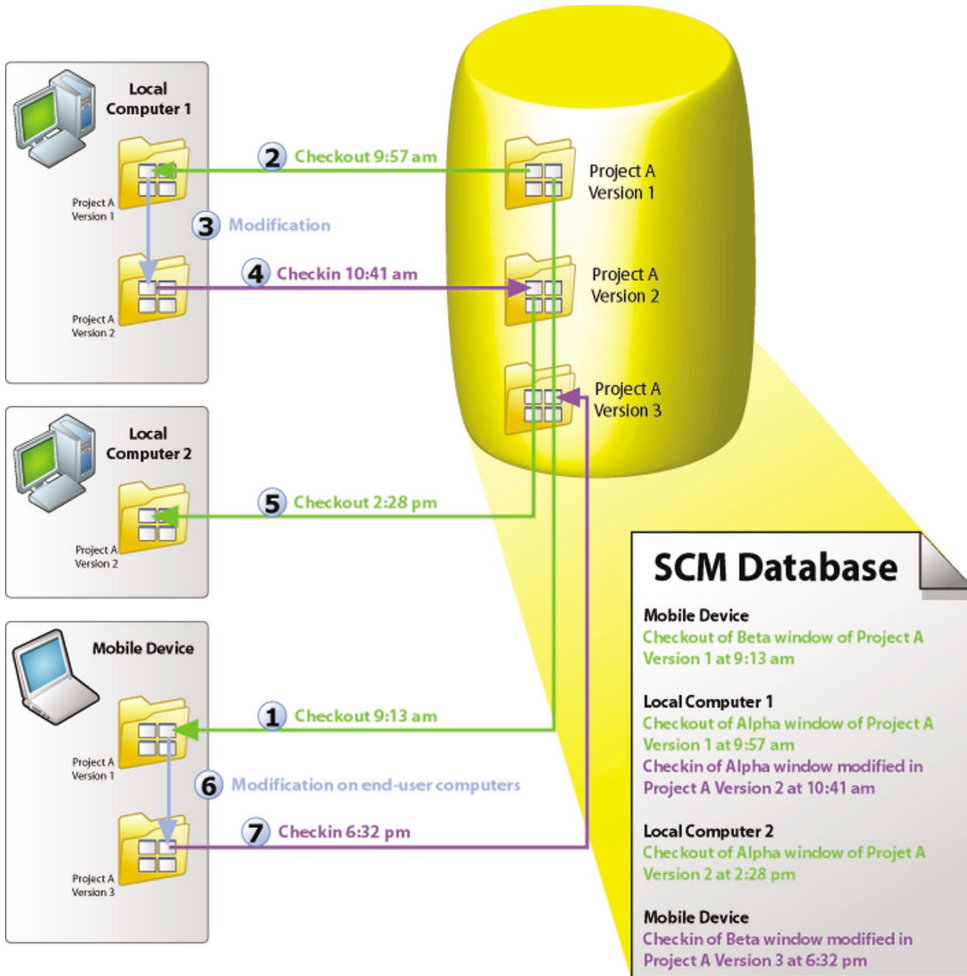


The elements found in SCM can be shared:

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

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 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 for more details (Keyword: "Developer groupware").

1 The Source Code Manager

1.1 Overview

WinDev has a new feature for teamwork management: the **Source Code Manager** (also called SCM).

Completely integrated in the environment, the **source code manager (SCM)** is used to:

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

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

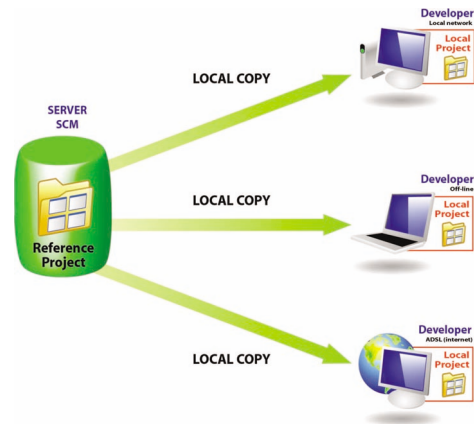
SCM uses a SCM database: procedures, classes, windows, reports, components, analyses, ...

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 (train, 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.

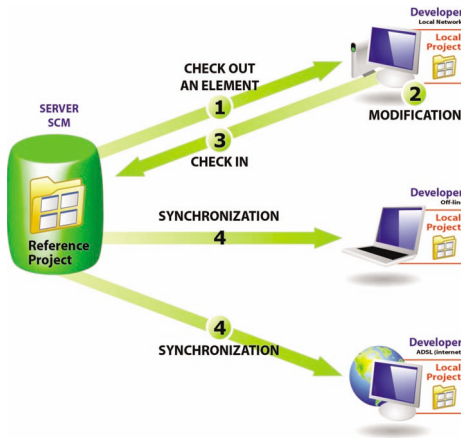
Each developer who uses the Source Code Manager gets a local copy of the project.

Use

To work on a project element (window, page, ...), 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 user 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 drives storing redundant 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 "secure" zone and use a firewall.

2 Creating a SCM database

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 where 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 SCM users must have the proper 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 source 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 source database be created?

The SCM database must be created once only.

WinDev lets you create a source 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 advise you to perform backups of the SCM database on a regular basis. These backups can be performed using the SCM administrator.

3 Configure the project 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 to SCM

To add a local project to SCM:

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

Note: if you have not already created a source database yet, click the "Install 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 Hyper-

FileSQL 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. Click the "Create" button.

3.2 Sharing resources

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

Sharing is an important SCM option. Various methods are available for sharing. See the online help for more details.

4 Working with SCM

4.1 Options of a project affecting SCM

Several options enable you 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 executables, libraries, components, ...**
When creating an executable, a library and so on, this option is used to display the list of checked-

out elements in order to check in and get the latest version of the elements. Therefore, the executable, the component or the library can contain the most up-to-date elements.

By default, the executable and the library are generated with the project elements currently found on the local computer.

- **Check out/Check in the project automatically:**
This option is used to automatically manage the "project file". If this option is checked, the project file is checked out only if the action performed requires it. Once the action has been performed on the project, the project file is automatically checked in.
This option allows you to not have the "Master/Guest" management on the project. This option can also be enabled via "SCM .. Manage the project .. Manage the project checkout automatically".

4.2 Checking out an element

The different checkout modes

SCM offers two modes for checking out the project elements:

- **the standard mode:** if you display a SCM element that is not checked out, a panel indicates that 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 attempt to modify an element that is not checked out, SCM automatically proposes to check out this element. Once the checkout is validated, the element can be modified.

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

To change the checkout mode, select "Tools .. Options of WinDev" and select the "General" tab. "Check out the elements during the first modification" enables you to switch all the projects opened from the current software to the automatic mode.

Opening a project element to modify its characteristics

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

1. Check out the element from the Source Code Manager.

2. Select the checkout mode of the element:

The checkout mode can be:

- **exclusive:** no one will be able to check out this element until it is checked back in. The element can be checked out for test only.
 - **For testing:** the element can be modified but the modifications cannot be checked back in.
 - **multiple:** the element can also be checked out by other developers. 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, select "Check in" from the popup menu of the element (in the project diagram or in the "Project Explorer" pane).

When checking in an element, a screen comes up where the following actions can be performed before the element is checked in:

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

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

4.4 Project management modes

Two project management modes are available with SCM:

- Management of the project in Master/Guest mode
- Management of the project in automatic mode

Master and guest

The Source Code Manager distinguishes between 2 types of users:

- the master: the master is the user who put 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's 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 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 in.

The automatic mode allows you to not have to manage 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 dif-

ferent 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 various projects included in the source code manager.

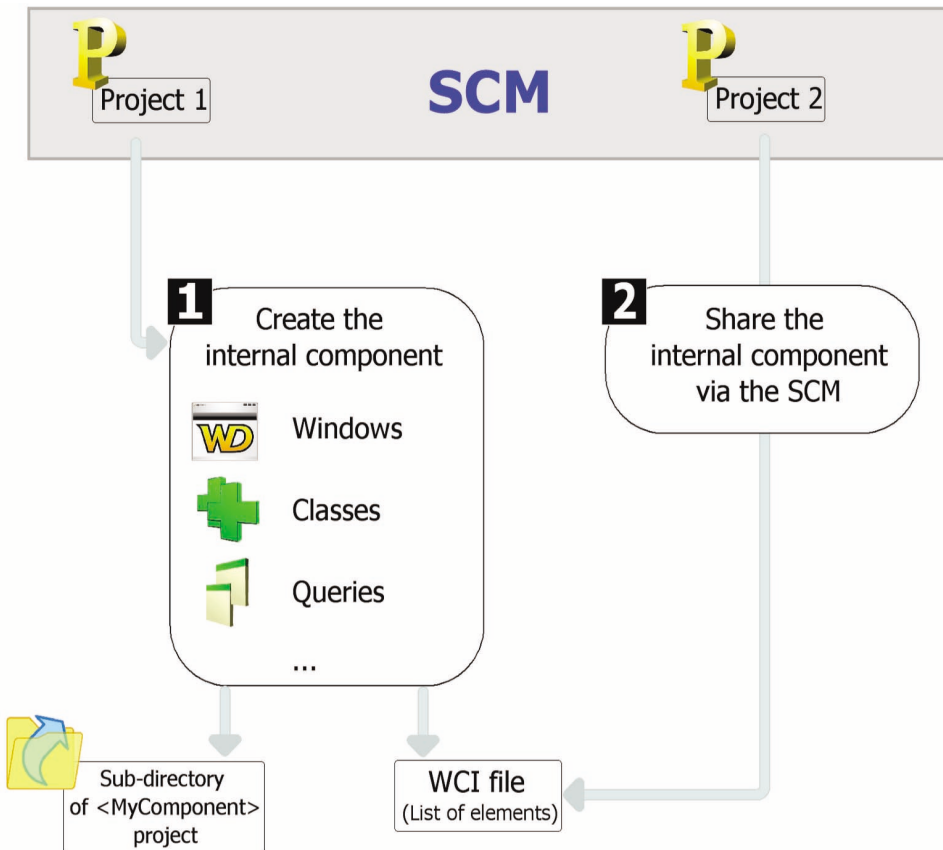
It allows you to:

- manage the SCM databases (creation, connection to a SCM database)
- manage branches
- manage the files and the directories found in a project of the SCM database (add, delete, rename files and directories)
- manage the different files found in the SCM database (check-in, check-out, share, and so on).
- start some tools (options, maintenance, and so on).
- restore a project version.
- see the history and the version differences in the source.
- cancel checkouts (in administrator mode)
- purge, back up, restore a database, ...
- add files of any type to 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 instance.
- 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 can create internal components to group the elements of a project by feature set for example.
- **Share elements among different projects:** elements found in an internal component can be shared among several projects via SCM. See "Sharing the internal components (via SCM)", page 69, 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.

Difference 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 .. Creating an internal component". The wizard for creating an internal component 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, templates, set 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 offered automatically by the completion feature. 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 auto-com-

plete feature will propose these "private" elements only from the elements of an internal component).

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

- **Full share:** the internal component accesses the project's data file. The internal component and the project use the same runtime contexts. This mode corresponds to the default mode if the component uses no analysis.
- **Share data only (advanced mode):** the internal component accesses the project's data files. The internal component and the project use different runtime contexts. This mode is reserved to specific cases.
- **Full autonomy:** 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 has no declaration code of globals. A set of procedures can be used to initialize the internal component.

2.2 Internal component and analysis: case of the 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 SCM)

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

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

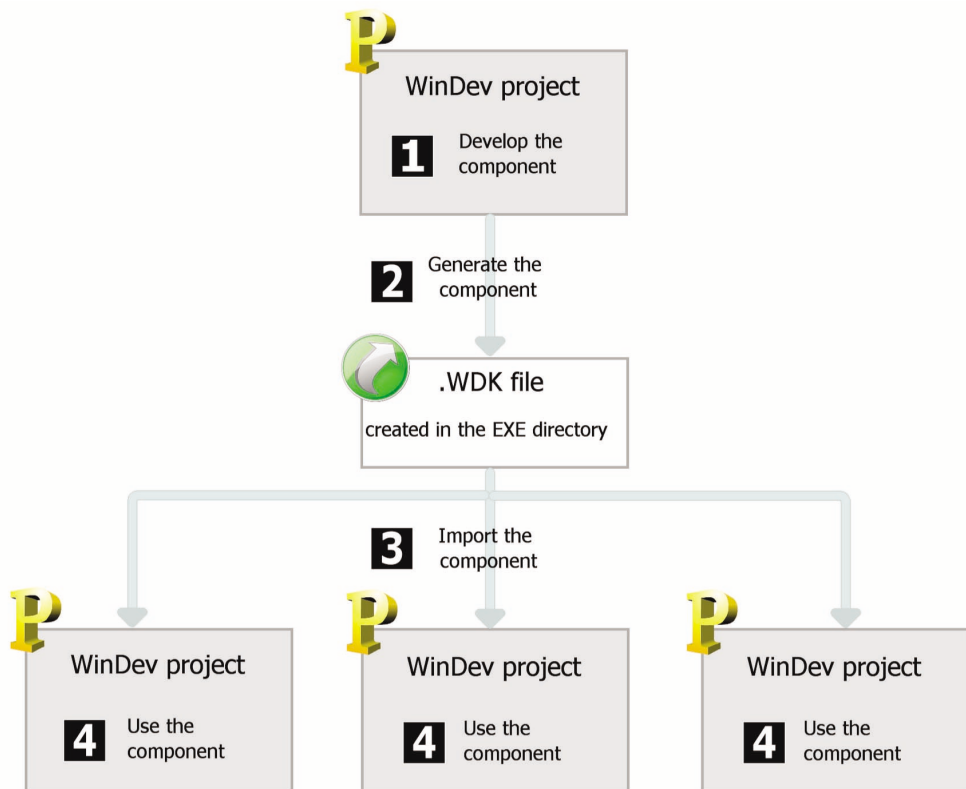
- create the internal component from a project found in SCM. The internal component will be automatically found in SCM.
- import a project containing one or more internal components into SCM. The internal components will be automatically included in SCM.

External Component

An external component is a set of 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 the 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 needing to have access to the corresponding source code (if it hasn't been distributed as well). The external component will then be added to 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 the emails
- ...

Notes:

- To optimize the organization of your projects, you have the ability to use internal components. See "Internal component in practice", page 68, for more details.
- In the rest of this chapter, the term "component" means "external component".

A **component can be distributed to other WinDev developers** (for free or for a fee). 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's elements. The developer can build a component from a WinDev project linked to a WebDev project and/or 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 features of the component, ...). A short documentation (to be completed) is automatically generated and associated with the component.

Note: some component examples are provided with WinDev. These components can be accessed 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. It can then be included in 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, 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 elements of the component. 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 (when developing applications only). This file is required to include the component in a project. This file must not be supplied in the setup program of the application that uses the component.
<Component Name>.WDO (optional file)	Optional file File in text format containing the list of additional elements supplied with the component (data files, .INI files, ...). See "The WDO file", page 76).

1.3 What is included in a component?

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 reused. This help is generated from the code comments.

Note: By default, if a component uses an analysis, some files, ... the HyperFileSQL management functions of WLanguage 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 a component

A component must be created from a project containing all the elements required for the component to operate. *Therefore, we recommend 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 then be distributed and reused.

3 Distributing a 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 directly providing the necessary files.

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

- **component setup procedure** (with a setup procedure of the component identical to the one used for applications).

Example: This distribution mode is recommended if the component is intended to be sold or distributed on a large scale, with regular upgrades.

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, etc.), you need to create a <Component name>.WDO file after generating the component. This file contains the list of external files (data files, text files, and so on) 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 operate. The files are copied into a specific directory. This directory can be found on a network server for instance.

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, and so on) as well as <ComponentName>.WDO. <ComponentName>.WDO contains the list of files that must be supplied with the component. See the "The WDO file", page 76, for more details about this file.

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, and so on) used by the component.

- the setup of specific tools (MDAC, ODBC driver for HyperFileSQL, etc.)
- 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 using "Workshop .. External component .. Create the setup procedure of a component".
 - from the list of components generated from the project ("Workshop .. External component .. List components generated from this project" "Setup" button).
2. In the different planes of the wizard, 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 the setup program of an application, the necessary files are created in the INSTALL directory of the project.

4 Using a component in an application

A component can be reused at anytime in any WinDev application: you just need to know the directory of the component's files and import it in the project. The component can also be imported via SCM.

When a new version of a component is available, simply install the new files of the update into the setup directory of the component (depending on the setup mode used). See "Distributing a component", page 72, for more details about the various setup modes for a component.

If the component was published and imported via SCM, updates are offered 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 operate are automatically detected.

5 Deploying an application that uses a 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 end-user computer:

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

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 operate.
- the executable of the application and the files required for the application to operate.
- the component and the files required for the component to work.

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

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

1. Recompiling the host project

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

- New features have been added to the component and 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 to the component but they are not required for running the application.

6 Modifying a component

6.1 Overview

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

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

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

6.2 The different types of compatibility

The management of the compatibility is linked to the versions of a component. Indeed, if some 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 used (.WDK file) to compile applications using this component must always be greater than or equal to the current version used in deployment.

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

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

7 Advanced component settings

7.1 Automatic documentation

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

WinDev simplifies the creation of this documentation by offering:

- **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 in the code of the component's 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

```

Global Declarations of Form_Customer_WI
// Description of in/out parameters for
//
//Syntax :
//Form_Customer_Window ({<sMode>[,<nCus
//
// This procedure opens the customer wi

PROCEDURE Form_Customer_Window({sMode="Cf
//
// This procedure opens the customer wi

SWITCH sMode
CASE "CREATION" :
    
```

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

When is the documentation generated?

The component documentation is generated when generating the component for the first time ("Workshop .. Component .. Generate a component").

When generating this documentation:

- the comments found in the code are used to create the documentation.
- if there is no comments, 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: each time the component is generated, you can generate the documentation associated with the component again ("Regenerate" option in the "Component history" window).

Caution: if you use the "Regenerate" button, the modifications performed in the generation wizard will be lost.

How do I access the documentation of a component?

The **general overview 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 including a component in a project ("Workshop .. External component .. Import a component into the project").
- at anytime when double-clicking the icon of the component found in the "Wizards, Examples and Components" pane or via the "Description" option of the component's popup menu.
- at anytime, in the description of the components included in the project ("Workshop .. External component .. List of components imported into the project"). The "Description" button lets you get some information and the documentation of the component.

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

- by double-clicking this element (from the "Project" pane or the project graph) or by pressing [F2] from the code of this element.
- directly from 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 element by programming (as any other project element).

Note: However the code of this element is not visible.

- If the element is not accessible, the user will not even know that this element exists.

Caution: depending on the declaration mode of the project's elements (class, Set of procedures, etc.), the elements that are accessible or not may vary. The table found on the next page summarizes the various ways you can access an element depending on 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:

<Component-Name>.WDK	Contains all the elements that must be redistributed (windows, reports, ...) with the component
<Component-Name>.WDI	Interface of the component. This file contains: - a help for using the component when it is checked 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 need to add the following in the project's EXE directory:

- A <ComponentName>.WDO file: This file contains the list of external files (data files, text files, and so on) 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 located in a specific tree. 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 the standard text editor of Windows, Notepad.

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

This ".WDO" file may contain:

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

tory of the current directory

- .\FileName.xxx to specify the current directory
- ..\FileName.xxx to specify the parent directory
For example: \SelectorComponent\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 a component in an application", page 73, for more details.

Example: the "Postal 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 where the component was created. 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 you do not use a setup program for the component, you must provide:
 - the WDK file
 - the WDI file
 - the WDO file
 - all the necessary files referenced in the WDO file.
- If you use a setup procedure for the component, the WDO file will be created automatically when creating the component's setup program.
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 Windows applications, WinDev allows you to generate a large number of other types of projects.



Applications

The applications are the most often used generation mode. Applications built by WinDev can target the following runtime platforms: Windows (32 bits or 64 bits) and Linux.



Services

WinDev can be used to generate some services for Windows (in 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 without any user interface.



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 some stand-alone libraries that can be used by other applications as well as corrective patches for an application that is already deployed; this enables you not to have to reinstall the full application for a minor correction.



External components

The external components are application bricks allowing you to share one or more specific features 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 a XML Web service) can be generated from a WinDev project. A Webservice exposes a set of functions (one or more sets of procedures) and make them accessible through the Web (or a private network) by using the HTTP and SOAP standard 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 Microsoft .Net platform. 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 virtual Java machine, regardless of its operating system. This enables you to build some applications that operate in Windows, Mac OS or even Solaris for example.

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

Creating the executable in practice

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


The executable program (".EXE" file) can be run directly: you do not need to start WinDev to run the

application.

Once the executable is created, you can deploy the application 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 creating executables starts.

2. Specify:

- the **general options** of the executable (name, icon of the executable program, splash screen, etc.).
- 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; only one instance will be open at the one time. If the developer tries to run another instance of the program, the instance already running will be activated.
- the **library's use mode**. This library is optional. It can be created in a separate file (".WDL" extension) or added to the executable. Including the library in the executable program presents the benefit of having only one file to install or update, but it increases the size of the ".EXE" file.
- the **error message** that will be displayed when a programming error occurs.
- whether to accept User Macro-Code(UMC). If UMC is active, 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 executable directory.
 - the languages supported by the executable.
 - the **files found in the library** (if there is one).
 - whether to include the components used in the project to the executable.
 - **the location of the user groupware's files** (only if your application manages the user groupware).
 - the **location of the data files** to use by the executable.
 - the **version information**. This information will be displayed in the "Version" tab of the executable's property window (accessible from Windows explorer). The **version number** can be **incremented** automatically each time the executable is created.
 - the operating mode of the WinDev framework (using the framework specific to the application or using the framework common to all the WinDev applications installed on the computer).
 - the manifest to use (when running on Windows Vista only).
 - whether the executable must be saved in the reusability center.
 - whether a **project backup** needs to be done.
3. Validate the executable's description. The executable is automatically generated based on the provided information. The executable can be run directly after it has been created.

Java application in practice

1 Overview

WinDev enables you to generate Java applications without even knowing the Java language. As with 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 website of the database publisher).

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 developed with WinDev, the Oracle J2SDK must be installed on the development computer (version 1.4.2 or later required). A version is available on the setup CD of WinDev. This program can also be downloaded from the Oracle Web site.

2 Features of a Java project

2.1 Creating a Java project

A Java project is created as any other WinDev project. To do so, select "File .. New .. Project". The wizard for project creation starts. In one of the first screens, the wizard enables you to choose the type of generation for your project. All you have to do is select "Java" and follow the instructions given by the wizard.

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

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

- The window RAD or the full application RAD is

1.2 Why develop a Java application with WinDev?

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

This way you can share your network applications.

1.3 Principle for generating a Java application with WinDev

As mentioned above, you do not have 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 WLanguage functions are converted to Java automatically.

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

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 for deploying the Java application.

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

2.2 JAVA RAD


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

RAD takes into account all the requirements for generating Java code (controls, processes, functions available for 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 window creation wizard is offered.
- JDBC: you can then specify the parameters for connecting to the database using the JDBC driver.

2.3 Test of a Java project in the editor

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

To run the Java application, you need to run the ".Jar" file located 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 about the WinDev elements that are not available in a Java application.

Note: as mentioned in the "What is a Java application?", page 80, you need to install the Sun J2SDK on the development computer to compile your project in Java.

2.5 Creating a Java archive

WinDev enables you to:

- use Java compiler to create and compile the Java files needed for 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, the corresponding Eclipse project can be generated.

To create the Java archive simply start the executable's creation ("Workshop ... Java .. Generate Java archive"). A wizard helps you to create your java archive then.

2.6 Deploying a Java application

As 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 setup creation wizard, select "Workshop .. Create the setup procedure".

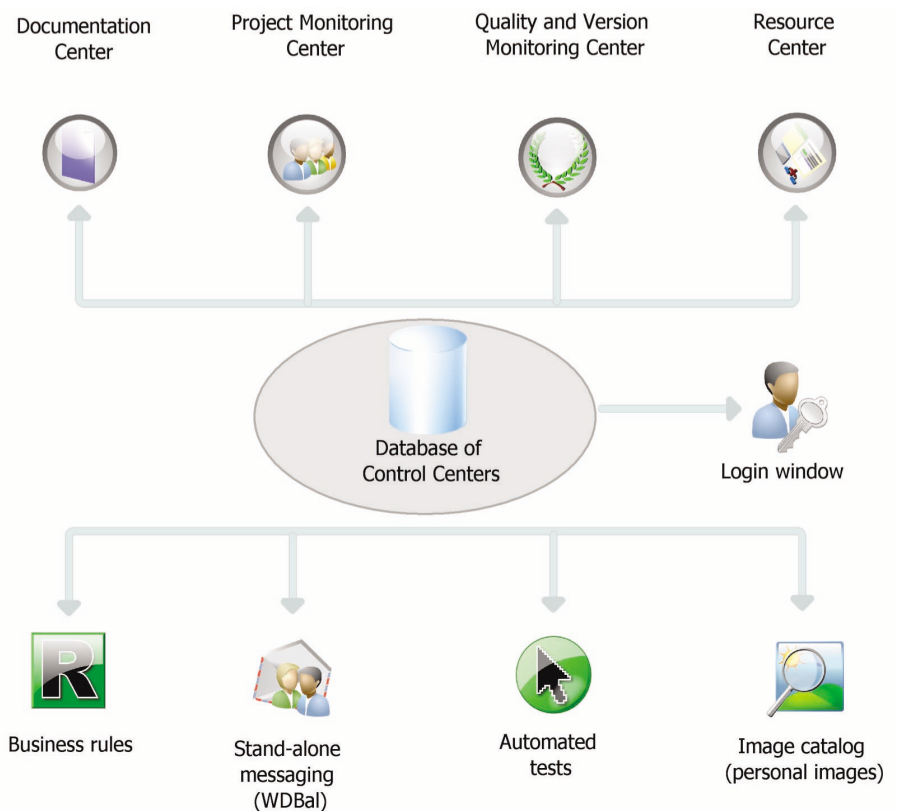
See the online help (keyword:"Java") for more details.

Control Centers

To manage the life cycle of your development projects, WinDev offers you various Control Centers. The Control Centers make up a set of tools meant to handle:

- a project's requirements (needs 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

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



As soon as WinDev is installed, it will offer:

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

Managing requirements

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

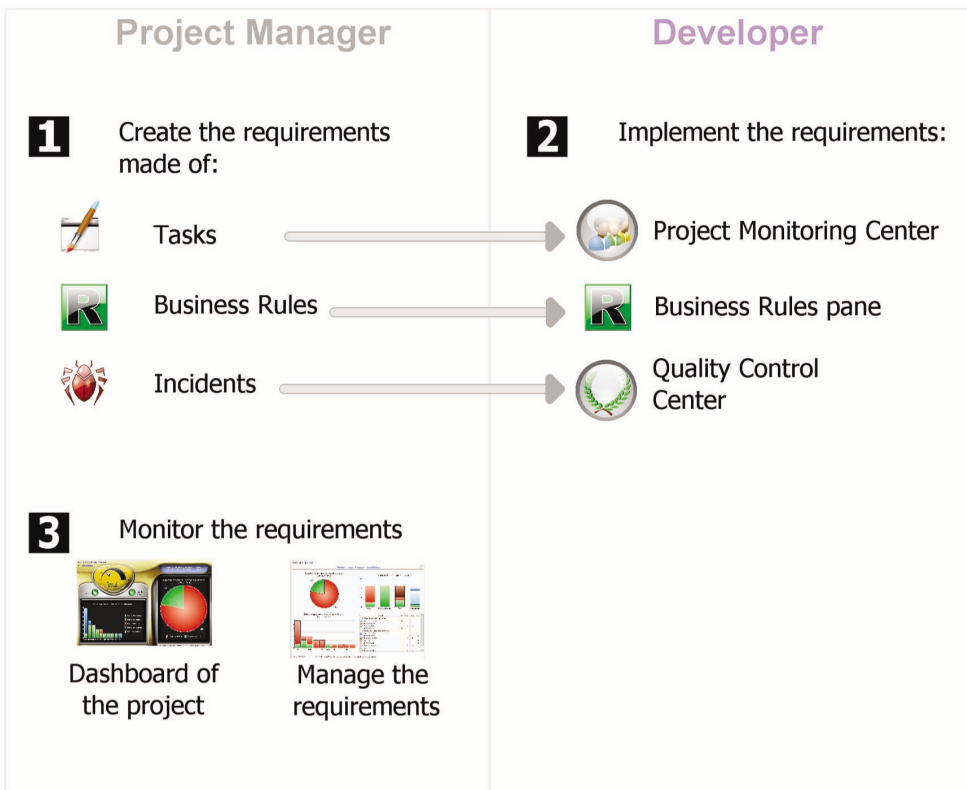
The Control Centers allow a project manager to manage the requirements of a development project.

To do so, you must:

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

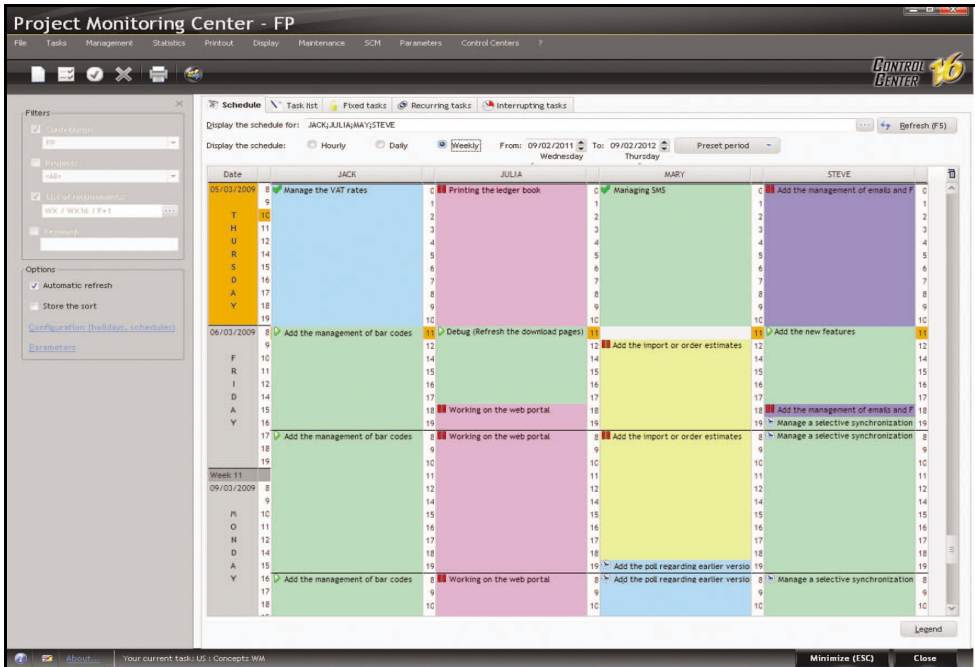
Each developer performs the different tasks assigned to him.

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



Project Monitoring Center

The Project Monitoring Center allows the different 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.

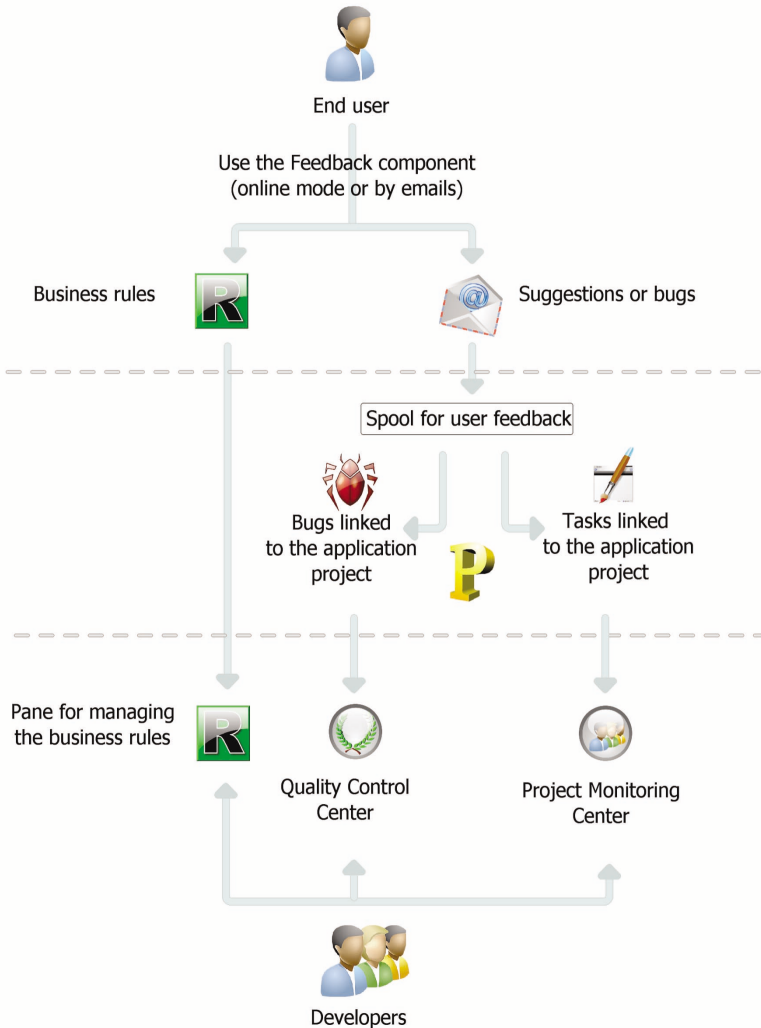
When the relevant project is opened, the Project Monitoring Center requests or indicates the current task. As soon as a task is completed, all you have to do is indicate that this task is over and specify the new task.

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 the incidents

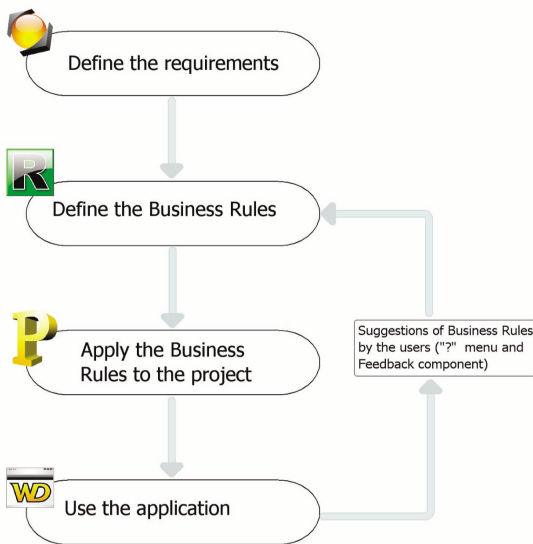
The Control Centers also enables you to manage user feedback about your applications. If your application contains an automatic help menu ("?" menu with "Send suggestions..."), the users of your application will be able to easily send you suggestions, incidents or business rules.



Note: in case of a fatal error during the application's execution, an error report can be generated automatically and sent to the developer using the "Feedback" component provided with WinDev.

Managing 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 commercial 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 associated with the project are directly displayed in the "Business rules" pane of the development environment. This pane presents the number of project elements to which the business rules apply and the percentage of rule currently implemented.



PART 3

Databases



DEVELOP 10 TIMES FASTER

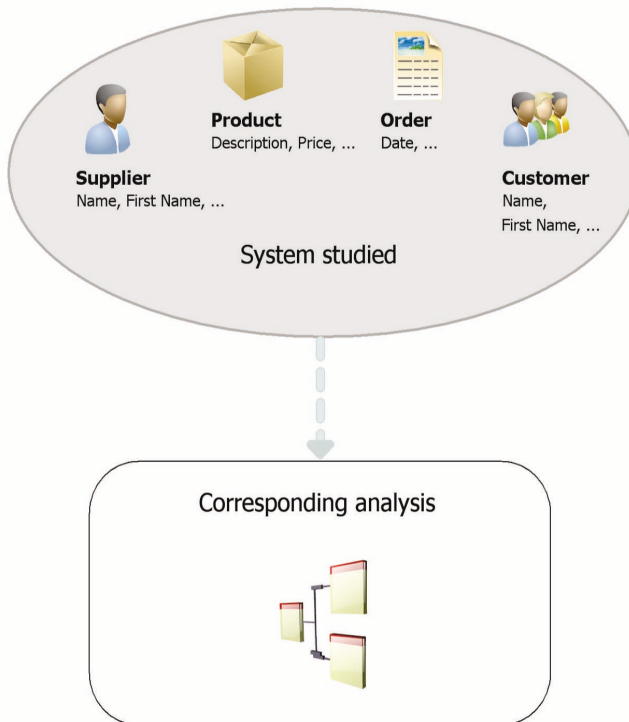


PC SOFT

Analysis: Structure of the database

When some data files are used by a WinDev, WebDev or WinDev Mobile project, 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 is used to easily create an analysis.



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

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

The analysis

1 Overview

When a project uses some 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 an analysis (or LDM)", page 90.

- **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", page 95 and in "Generating the LDM from the CDM", page 96.

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 creating analysis 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 into an analysis

The analysis files 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 data file format for instance).

The following paragraphs explain how to create a data file.

To create a new data file:

1. Select "Insert .. Data file". The wizard for creating a data file starts.

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

3. Specify:

- **the logical name of the data file.** This name will be used to handle the data file.
- **the caption of the data file** that briefly describes the purpose 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 contains 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 data 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 WinDev native accesses (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 system

file of the computers where the application will be deployed support the NTSF format (Windows 2000, NT or XP).

Caution: If this option is checked, the data 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 data file. The description window of the items found in the data file is automatically opened. It enables you to describe the items of the data file.

To import a preset description of data file:

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 afterwards.
4. Select the items of the data file that must be kept. These items can be modified afterwards.
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 the data files" or "Structure of files .. Items".

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

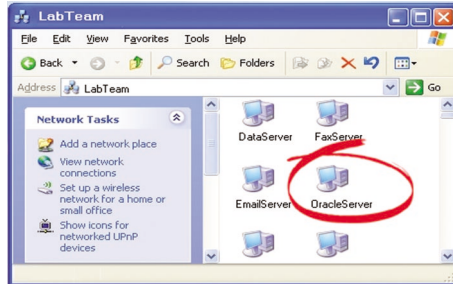
Solution 1: from the data model editor

1. Select "Insert .. Data file".
2. Check "Use the data files from an existing database" and select the database type.
3. Specify the data format used by your project. You can keep the current format or migrate the data files to the 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 data files whose description must be imported and validate.
7. The imported data file is automatically inserted into the current analysis.

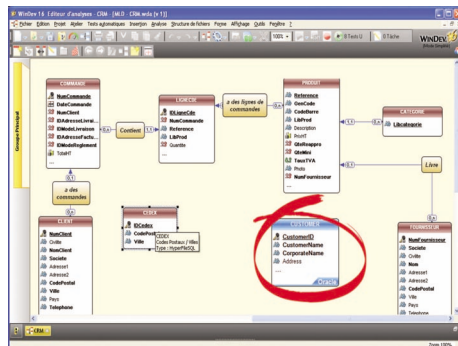
Solution 2: From Windows explorer

"Drop" the data file description from the 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 of the items found in the data file is displayed.
2. Click the first empty row in the table of items.
3. Specify the name, the caption and the 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 metatypes offered by WinDev. To do so, click the "+M" button on the right of the table. The list of available metatypes is displayed.

2.4 Creating a link

Different types of links can be created between data files. See "Characteristics of the links defined in an analysis", page 92, 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 cardinalities between the two links:
 - 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 define some advanced cardinalities, check "Display the advanced cardinalities" and answer the questions asked.
 5. Enter the caption of the link by briefly describing the subject of the link.
 6. Specify the keys to link.
 7. Specify the integrity rules. These rules are used to ensure the integrity of data 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 data file and member data file

For a link between 2 data files, there is an **owner data file and a member data file**:

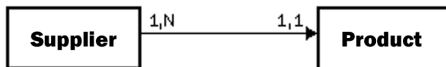
- the owner data file is the owner of the key.
- the member data file is a member of the analysis files containing a copy of the key.

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

This type of link is represented as follows:



3.2 Cardinalities

Cardinalities are used to count the links between data files.

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

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

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 "only one", the cardinality is 1,X.

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

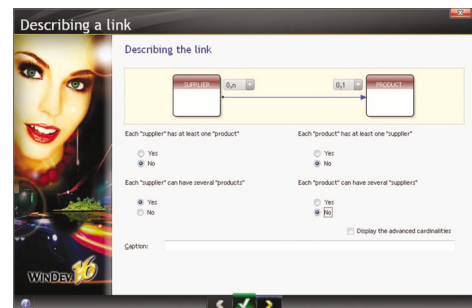
This answer provides the second part of the cardinality (**maximum cardinality**):

- if the answer is "only 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.

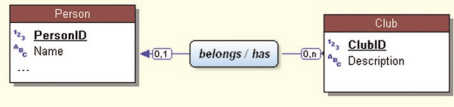
To be clearer, these two questions can be broken down into four questions. For example, to describe a link between the Supplier file and the Product file:



- 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 only be a member of one sport club. This person does not have to belong to a sport club.
- **Cardinality 0,N:** A club can have nobody or several people as members.

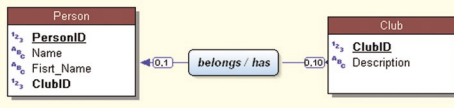
3.3 The advanced cardinalities

The cardinalities can be defined more precisely. We talk about advanced cardinalities.

These cardinalities are used to exactly specify the minimum cardinality and the maximum cardinality. To define the advanced cardinalities, check "Display the advanced cardinalities" in the link description ("Insert .. Link").

Example of advanced cardinalities:

This example presents two different cardinalities:



- **Cardinality 0,1:** A person can only be a member of one sport club. This person does not have 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 data file, the corresponding records are also deleted from the member data files,
- if a record is added into a member data file, a corresponding record exists in the owner data file,
- if a record is modified in the owner data 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.

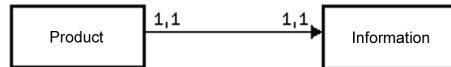
3.5 The different types of links

Several types of links can be found between the data files:

- parallel
- optional
- complement
- shared
- complex

Parallel link

For a parallel link, *each record* of a data file (Product) is linked to *a record* 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 data file is Product, the member data file is Information.

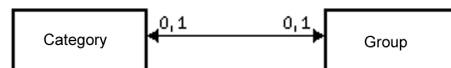
Note: This type of link is rare because the two data files can be combined into a single data file.

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

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

Optional link

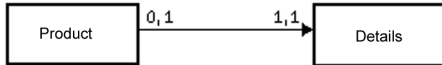
For an optional link, *each record of a data file* (Category) is associated *with no record or with a single record* in another data file (Group). *Each record* in the other data file (Group) has *no or only one record* associated with the first data file (Category).



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

Complement link

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



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

To manage a complement link, the key of 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 data file is Product, the member data file is Details.

Shared link

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

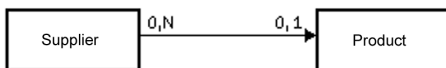
To manage a shared link, the key of the Supplier file is copied into the Product file. It becomes a multiple key to increase the speed of integrity check.

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

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

- Shared link with a 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

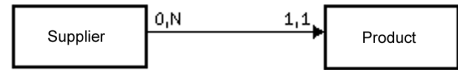
Shared link with a 0,N - 0,1 cardinality



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:

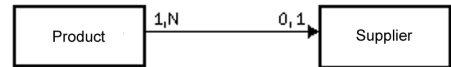
- an owner may have no member (a supplier may have no product).
- each member has a single owner (each product has a single supplier).

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:

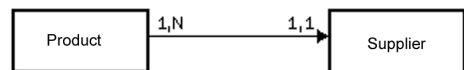
- each owner has at least one member (a supplier has at least one product).
- a member may have no owner (a product may have no supplier).



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 data file, called **relation data file**.

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

The relation data file can also contain some 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 data 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:

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

WinDev enables you to easily perform the following operations on a data file found in the data model editor:

- Duplicate/Copy a data file
- Delete a data file
- Rename a data file

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

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

See the online help for more details.

5 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 CDM creation 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** that briefly describes the purpose of the CDM.
 - **whether the CDM must be 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 your 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 briefly describes the purpose 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.
 - **the notes** that describe 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 the LDM is generated from the CDM, this name will be the logical name of the created file if this association contains at least one property or if it belongs to a complex link.
 - **the caption of the association** that briefly describes the purpose of the association.
 - **the notes** that describe 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 description window of the property is displayed.
2. Click the first empty row in the property table or

click the "+" icon.

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 .. Draw 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.
- Move 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 properties** (of the CDM) are kept 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 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 new, the corresponding item is created.
- if the property already exists, 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 files, the files and the items of the LDM are updated according to the above-mentioned method. 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 the entities has been modified, the cardinalities are 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 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 files. This generation is used to:

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

As long as the analysis (the LDM) has not been generated, the description of the analysis (the 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 .. Manage the Versions").

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 rectangle indicates the generation number of the analysis; the caption on the right indicates the generation date.

The "small rectangles" displayed between two versions indicate:

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

To see the details of 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 into the specified directory. An independent analysis corresponding to the selected version is created. This option is selected by default.
 - overwrite the current analysis.
3. Validate. The restore operation is performed.

Caution: The data files corresponding to an analysis whose version number is greater than the restored version cannot be opened anymore. In this case, you must also restore the files corresponding to the version of the restored analysis or delete the existing files in order to recreate them.

9.2 Canceling the last generation

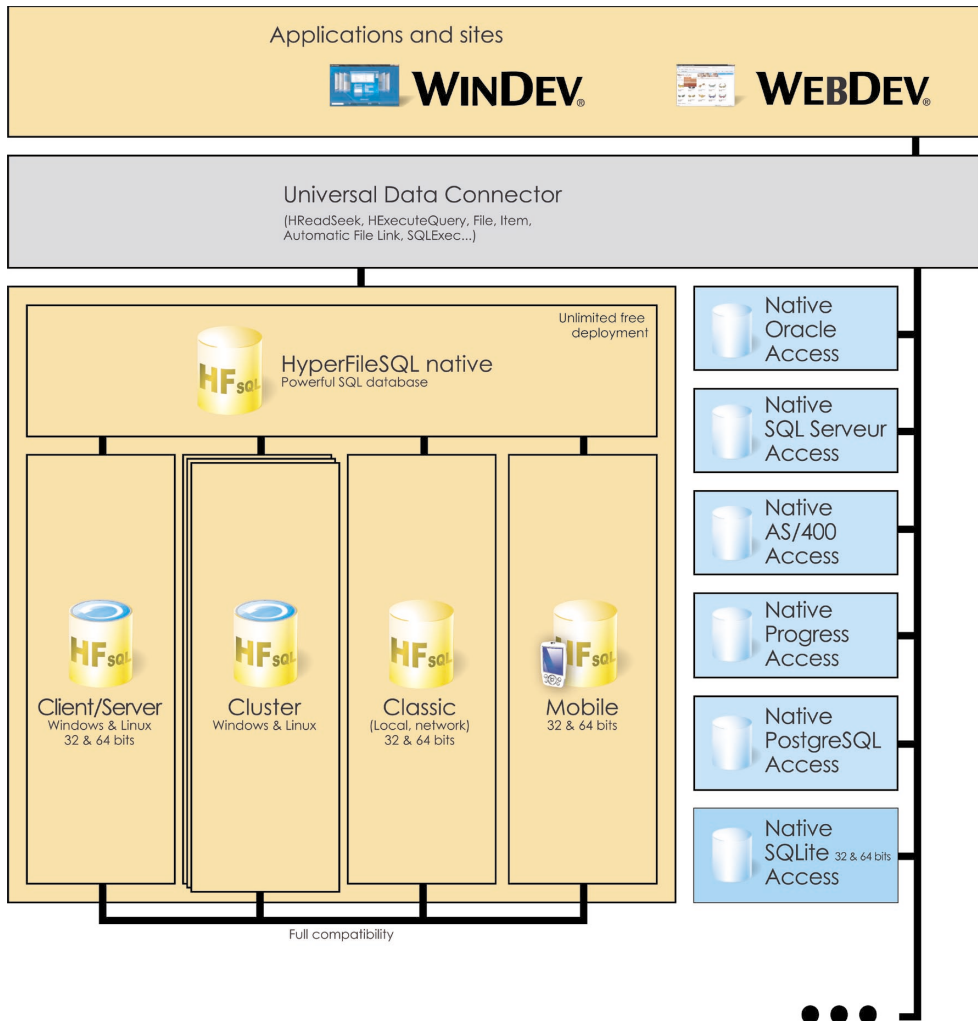
Canceling the last generation of the analysis is used to restore the analysis to its status before the last generation. The modifications performed since are not applied.

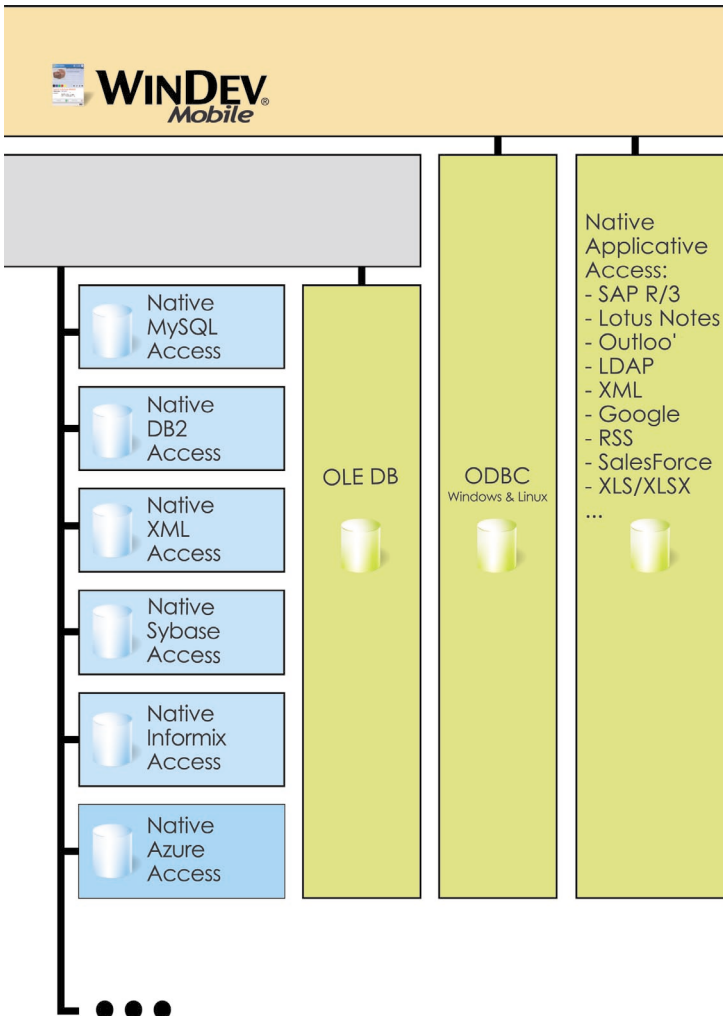
9.3 Resetting the version number to 1

You can reset the generation number of the analysis to "1" ("Analysis.. Reset the analysis version 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 easy access to most of the databases on the market.



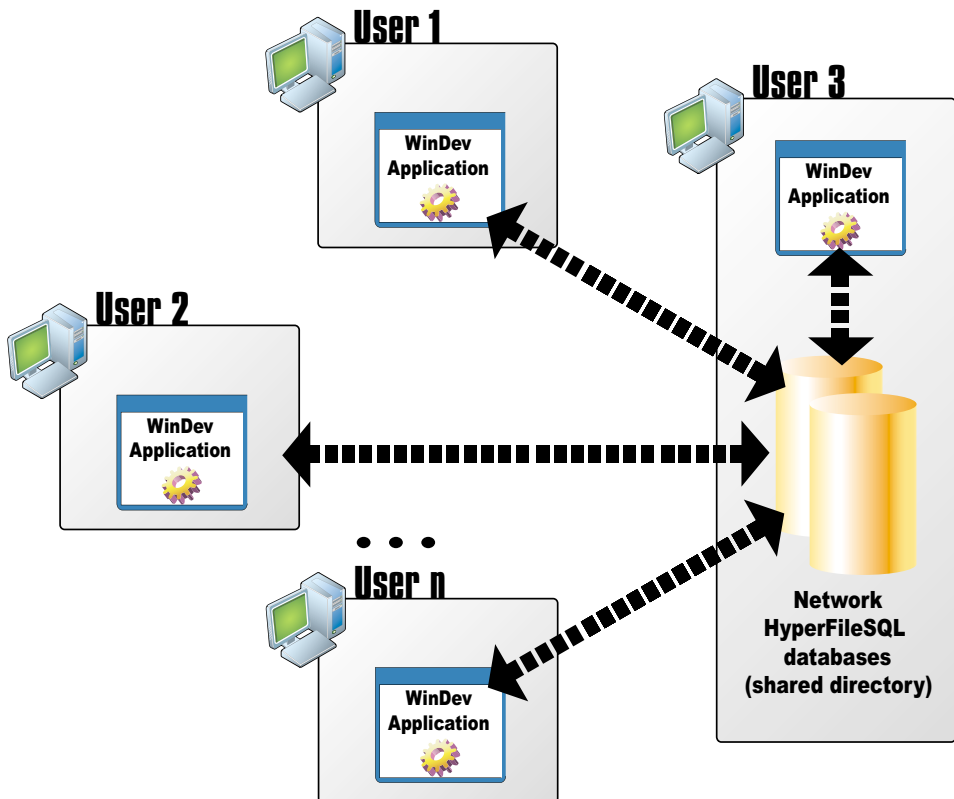


HyperFileSQL Classic

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 file, ...) are performed on each user computer.

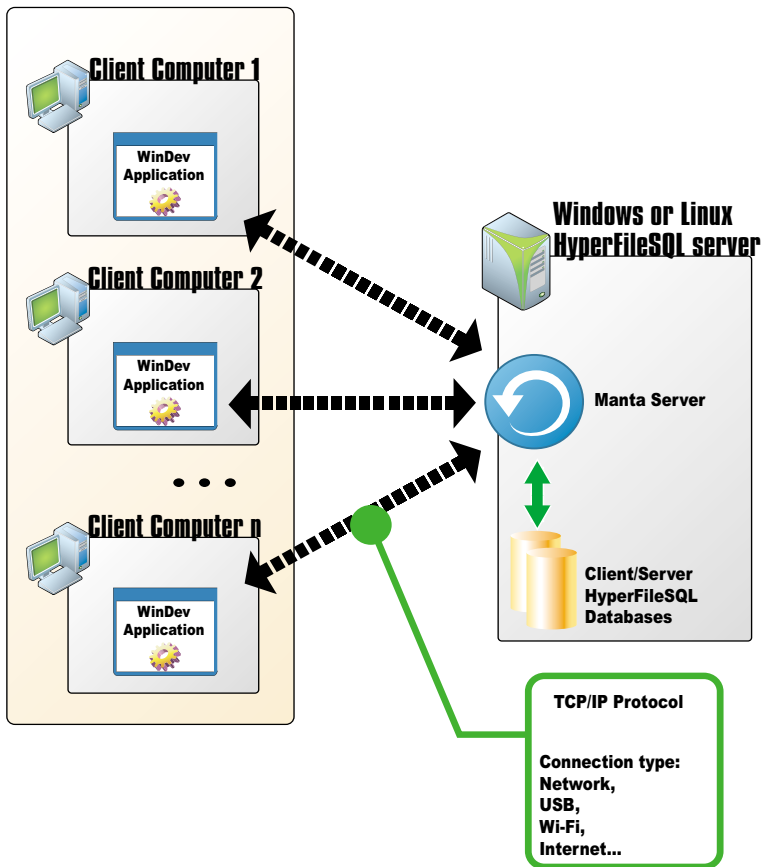


HyperFileSQL Client/Server

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:

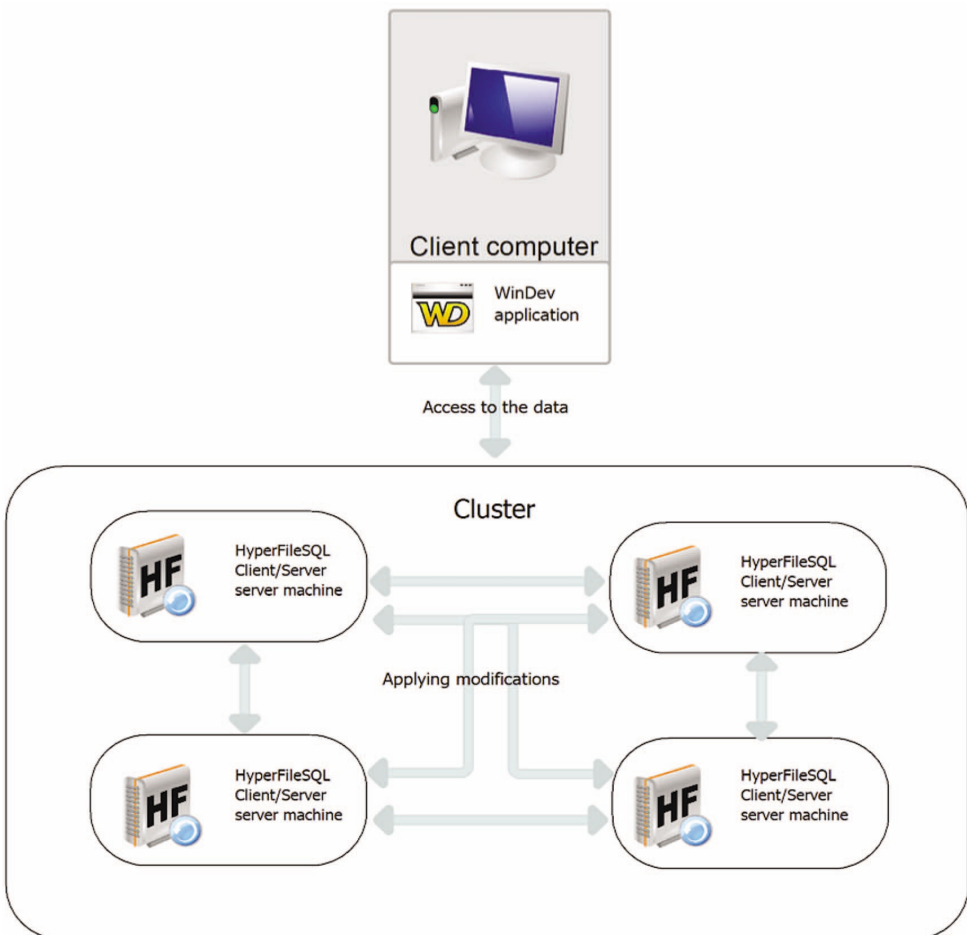
- A HyperFileSQL Client/Server application can be run on different user computers (called client computers) that communicate with the database server via the network.
- The data files are found on a server. Only the server physically accesses the data files.
- All the processes (query, read/add operation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



HyperFileSQL Cluster

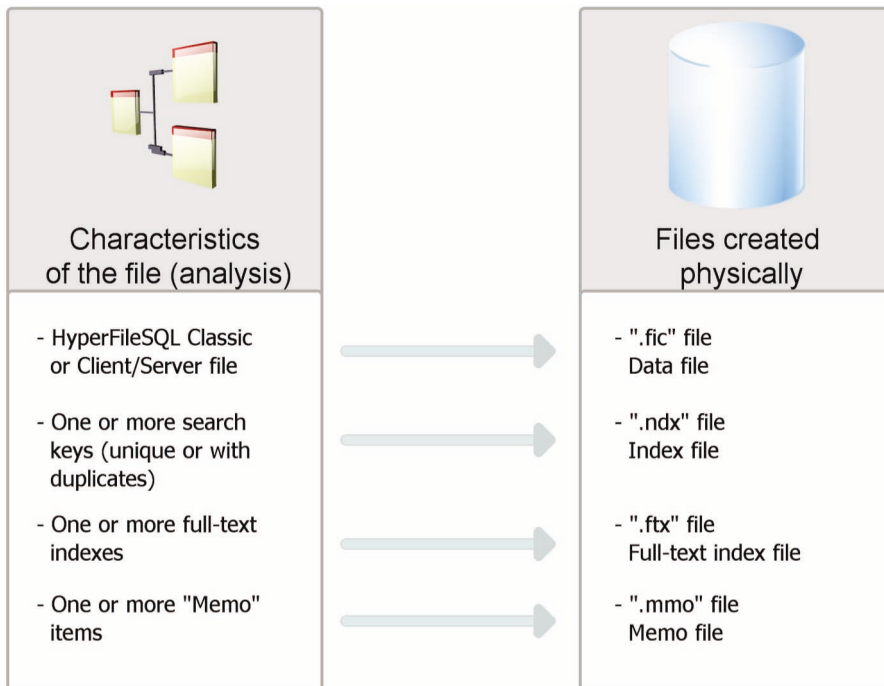
HyperFileSQL Cluster is an extension of the HyperFileSQL Client/Server database. In a database cluster, all the different HyperFileSQL servers contain a copy of the databases and are synchronized in real time.

- 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 physically created files

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



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

Associating the controls with the data

A window can display information coming from:

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

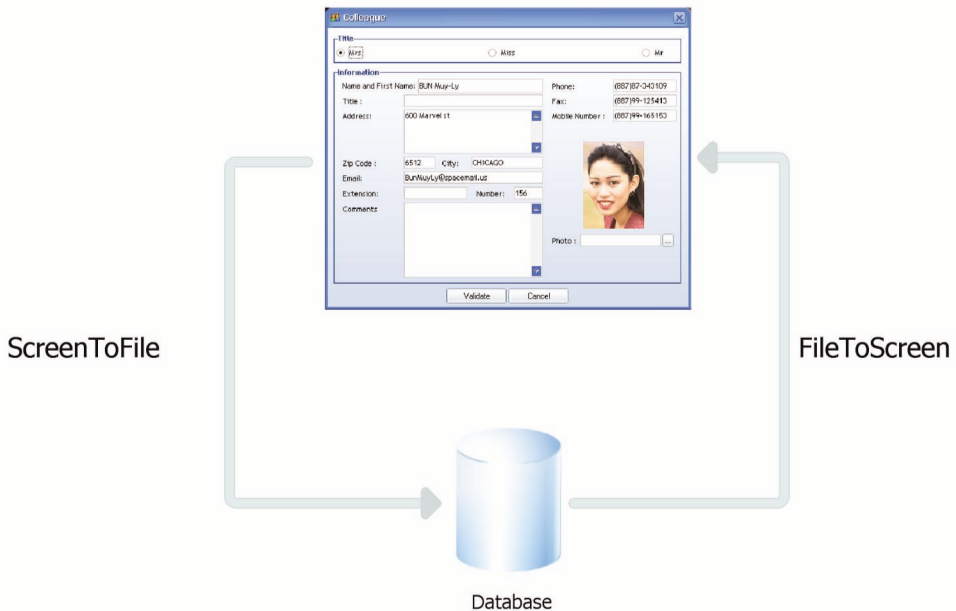
To display this information in a window, the controls of this window must be linked to:

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

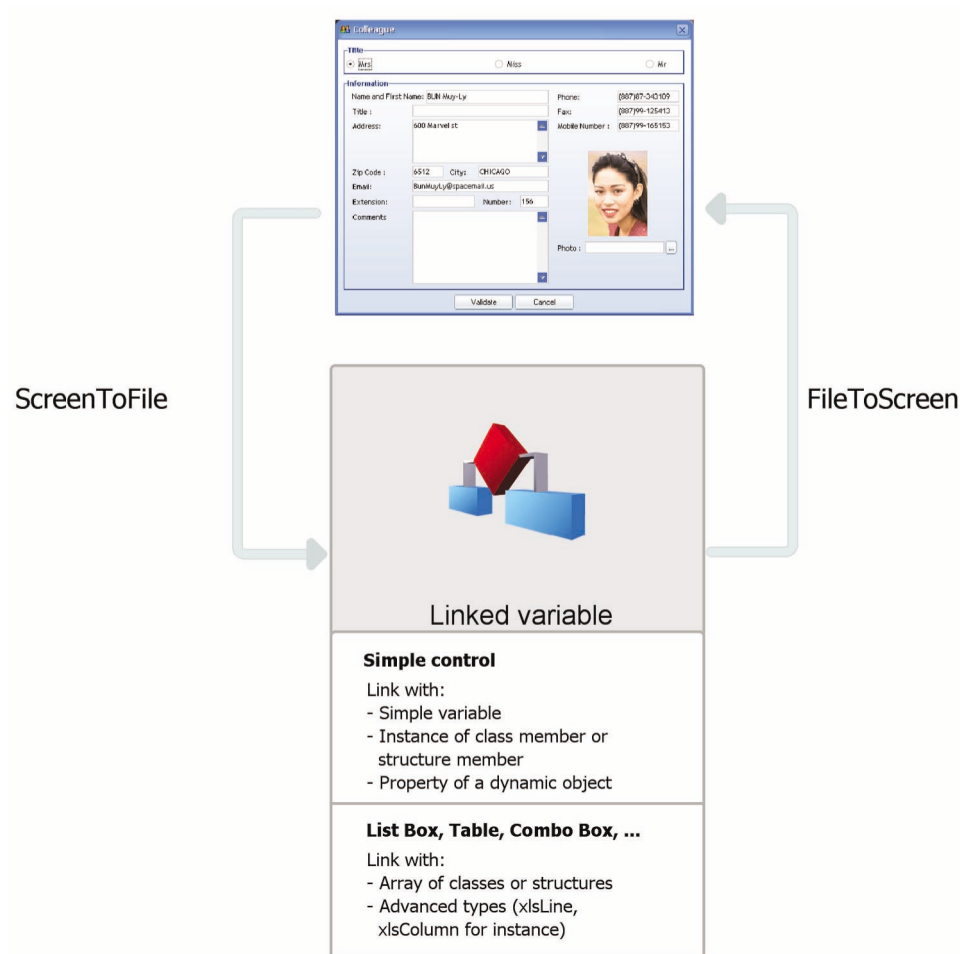
The method for displaying and retrieving information is straightforward:

- The link between a control and an item or 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 displayed on the screen.
- **FileToScreen** is used to update the data displayed on the screen with the information saved in the file or with the information saved in the variable.

Link between control and item



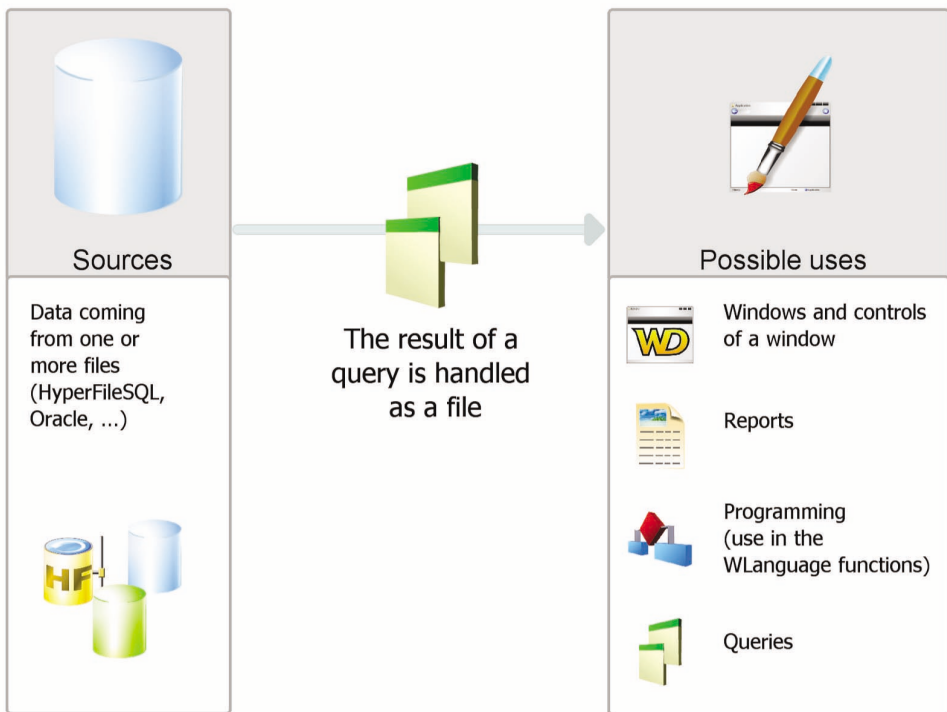
Link between control and variable



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 files.

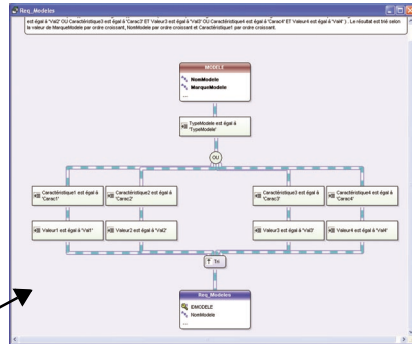
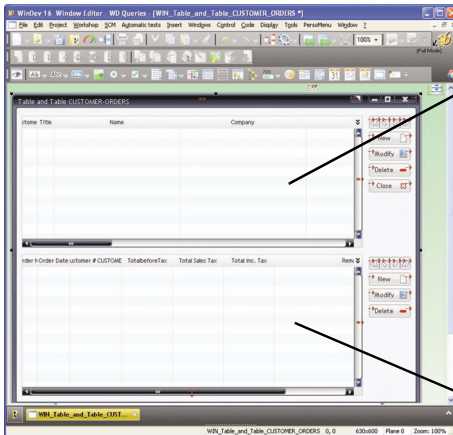
The query editor is used to easily create queries, without programming.



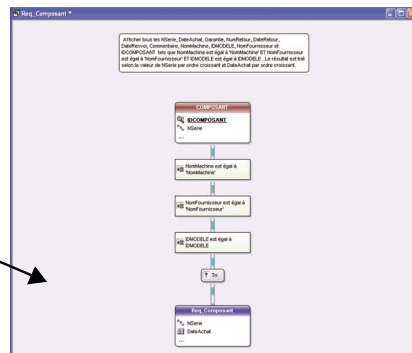
Note: in programming, a query can be handled like a file. Especially, it can be associated with a display control (a table for example) that will present the data returned by the query.

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 instance), the embedded queries used by this window will also be copied.

The Table/Looper controls

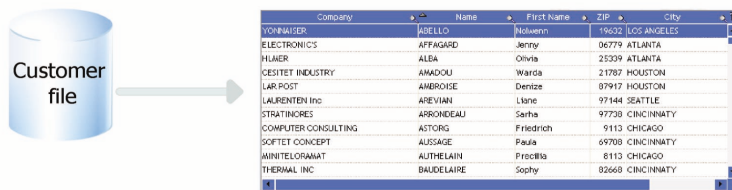
The Table/Looper controls can be used to display a set of information (the content of a data file for example). The content of these controls can come from three different sources:

- "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. Browsing the file is used to display the data in the table. The file is read for each row displayed: the record read is displayed in a table row.



The data displayed that is not linked to the file is not kept when refreshing the table display (when scrolling for instance).

The WLanguage functions starting with "Table" are used to handle the browsing tables with direct access. The addition or the deletion of a table row involves the addition or the deletion of the record in the linked 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 (with **TableAddLine** for example)



Since the data is found in memory, the table allows you to perform all the operations on the data (sort on any column, search performed in the columns, ...).

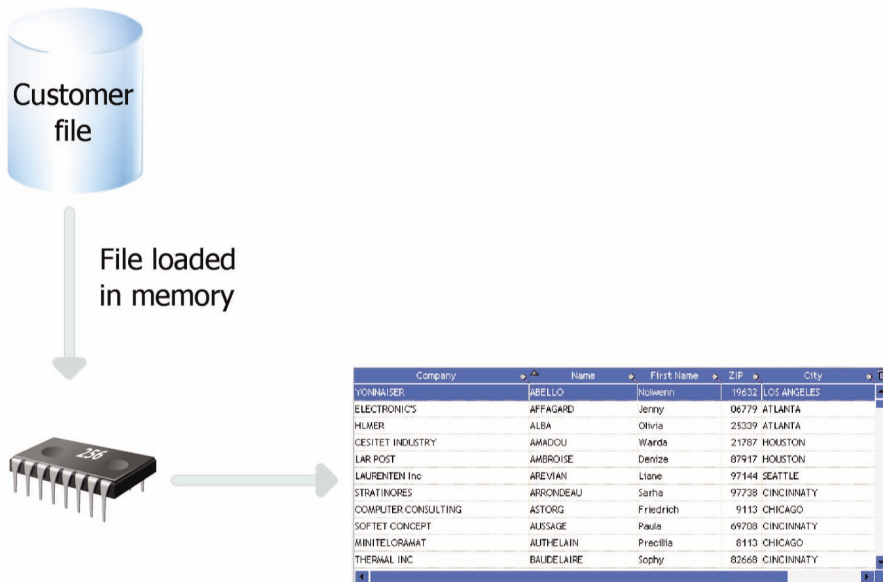
"Browsing table loaded in memory"

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 file is entirely loaded in memory. The sort and the search are available on all the columns.

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

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

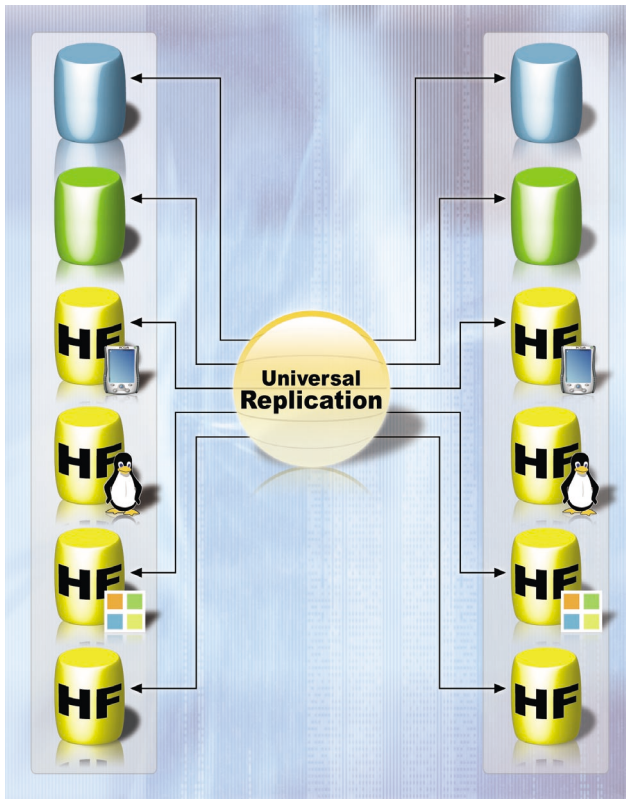


Note: these different fill modes are available also for list boxes and for combo boxes.

Universal Replication

The universal replication is used to update some databases of the same format or some databases of different formats (HyperFileSQL, Oracle, SQL Server, ...). You can for instance 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 instance, 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 changes must be done by programming via *HRpIFilterProcedure*.

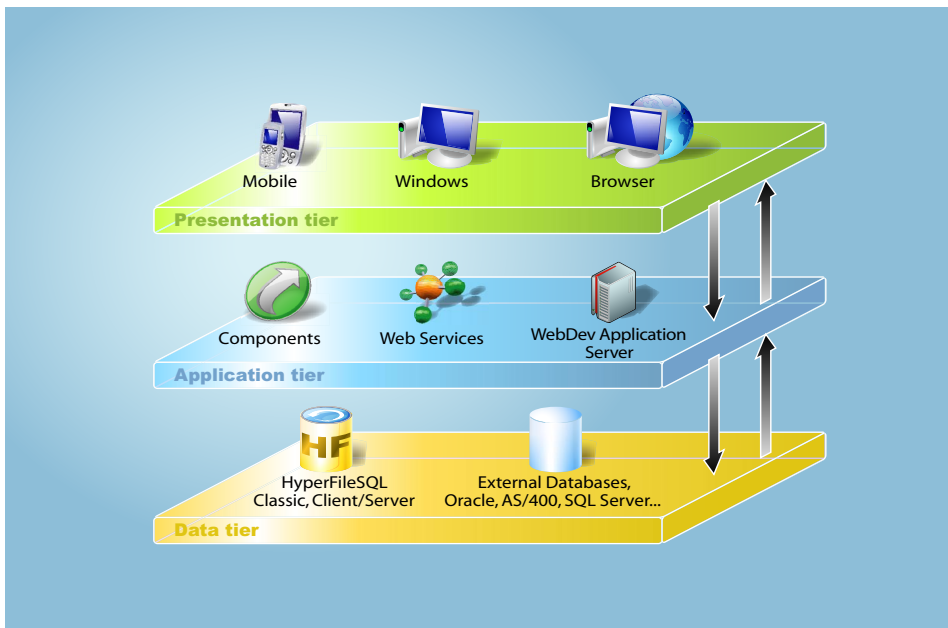
3-Tier Architecture

The 3-tier architecture is a model for application architecture.

Its basic principle consists in separating 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 them is to facilitate maintenance and future upgrades of the application. This ensures higher security because access to the database is allowed via the process tier only. It also optimizes teamwork and multi-target development.

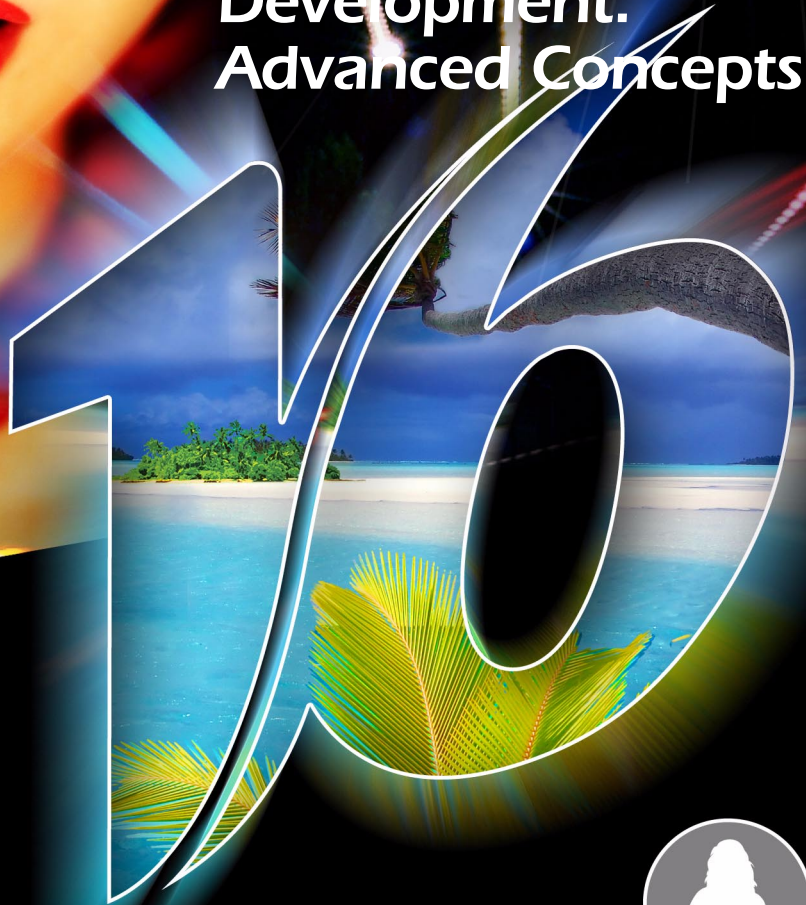


The 3-tier architecture is fully compatible with the development of applications with WinDev.



PART 4

Application Development: Advanced Concepts



PC SOFT

DEVELOP 10 TIMES FASTER

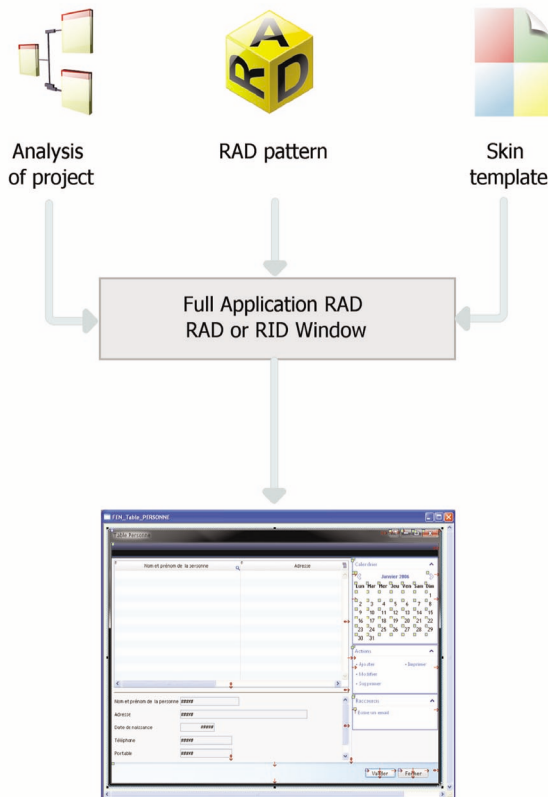
RAD RID

RAD (Rapid Application Development) and RID (Rapid Interface Development) let you create windows from:

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

In RAD generation, the generated windows contain all the code needed for them to work. The test of these windows can be immediately run with the data found on the development computer.

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



Example of window generated
by RAD or RID

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:** describes the overall structure of a system.
- **Use case diagram:** presents the system's features from the user's standpoint.

- **Object diagram:** presents 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:** presents the behavior of a method or a use case flow.
- **Sequence diagram:** presents the chronological order of the messages sent and received by a set of objects.
- **Collaboration diagram:** presents the structural organization of the objects that send and receive messages.
- **State-transition diagram:** presents a sequence of states that an object goes through.
- **Deployment diagram:** presents the distribution of the hardware (the nodes) used in a system and the association between executable program and this hardware.

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

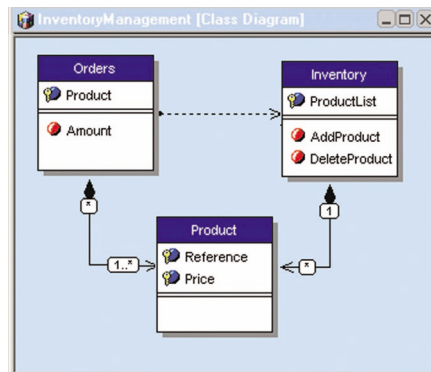
2 The various UML diagrams

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:

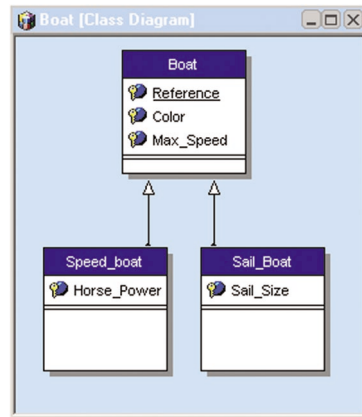
- **Class:** presents the structures of the application. Each class is divided into three compartments:
 - *the class name* indicates what the class is and not what it does.
 - *the class attributes* provide the characteristics of the class.
 - *the class operations* indicate the possible actions on a class.

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

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

- **Public:** the element is visible by all the other classes.
- **Protected:** the element is visible to the class itself and to its subclasses.
- **Private:** the element is only visible to the class.
- **Relationship:** describes the behavior of the classes among themselves. Three types of relationships are available:

- **Association:** structural relationship among 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 *Stocks* class: before adding a product to an order, one needs to verify that the 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.



- **UML package:** divides and organizes the diagram's representation (the same way a directory organizes files). Each package can contain classes and relationships.

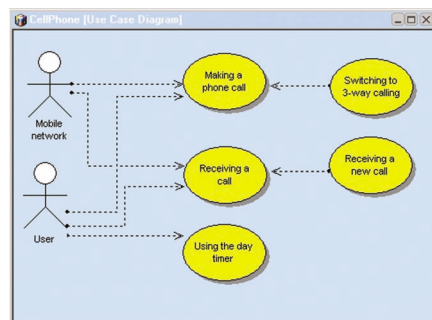
Via the generation of a class diagram, it is possible to create the structure of the WinDev classes used in your application.

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



A use case diagram contains the following elements:

- **Actor:** represents the role of the application's 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*.

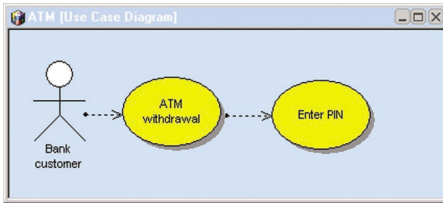
- **Use case:** describes a sequence of actions performed by the application. For example, *Place an order, Enter an order, Create a new customer form, ...*

A use case describes the actions performed by an application but it does not specify how the application performs these actions.

- **Relationship:** describes the behavior of 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*, customers must enter their *PIN number*. In this case, the *Get Cash* action depends on the *Password Input*.



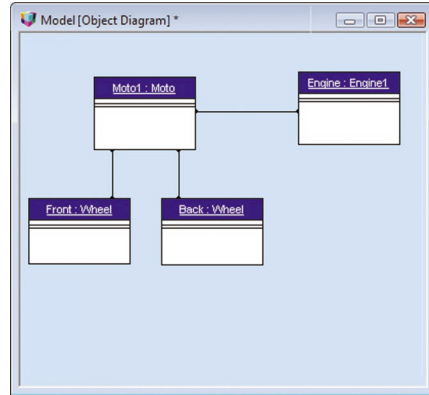
- **Generalization:** Relationship used to organize the elements according to a hierarchy. For example:
 - two types of *Customer* actor are available: *Individual Customer* or *Business Customer*.
 - the checking of a user identity can be performed in two ways: password input or fingerprint check.
- **UML package:** divides and organizes the diagram's representation (the same way directories organize files). Each package can contain actors and use cases.

2.3 Object diagram

An object diagram represents 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 instance).

For instance, the diagram below presents a part of the general structure of motorcycles:



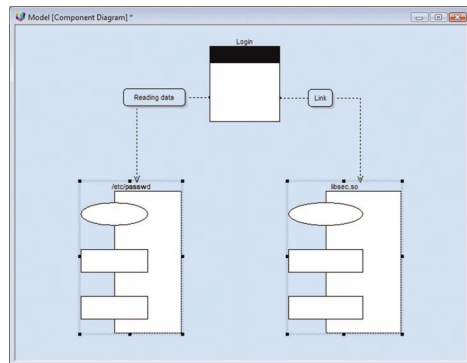
An object diagram contains the following elements:

- **object:** represents the instance of a class. Note: If a class diagram is open, you can 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 instance: a window that contains scrollbars, buttons, ...
- **link:** presents the relationships among 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.



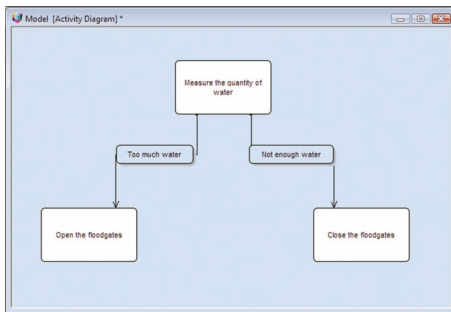
A component diagram contains the following elements:

- **module**: represents the various physical elements that make up a computer application. For instance: a file, a library, ...
A module can be represented:
 - either by a **specification** showing the module interface. This specification may be **generic** in the case of classes with parameters.
 - or by its **body**, which presents the module's implementation.
- **task**: represents a component that has its own control flow (thread).
- **main programs** of the computer application.
- **subprograms**: groups the procedures and functions that don't belong to classes.

2.5 Activity diagram

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

For example, the following diagram presents a dam's flow



An activity diagram is made of the following elements:

- **activity**: presents a specific step in a workflow. For example: "Create an estimate", "Open a window", "Check knowledge", ...
- **synchronization bar**: used to synchronize the different activities:
 - by indicating the activities to perform before a given activity. For example: "Press clutch" and "Change gear" before "Release clutch".
 - by indicating the activities to perform concurrently.
- **object**: attaches activities to the object that performs them. For example, the "Order" and "Pay"

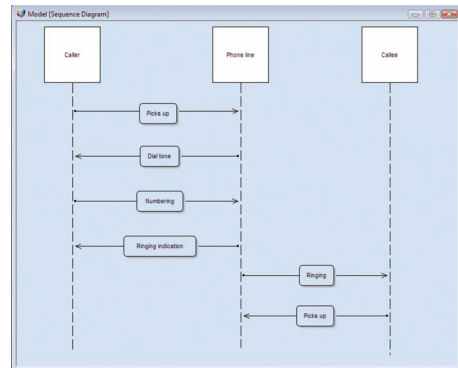
activities are attached to the "Customer" object; the "Teach" and "Check knowledge" activities are attached to the "Teacher" object.

- **send signal**: shows the sending of a signal to an object.
- **wait for signal**: shows the wait for a signal coming from an object.
- **transition**: shows the passage from a completed activity to another activity. For example: "Too much water", "Not enough money", ...

2.6 Sequence diagram

A sequence diagram presents 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:



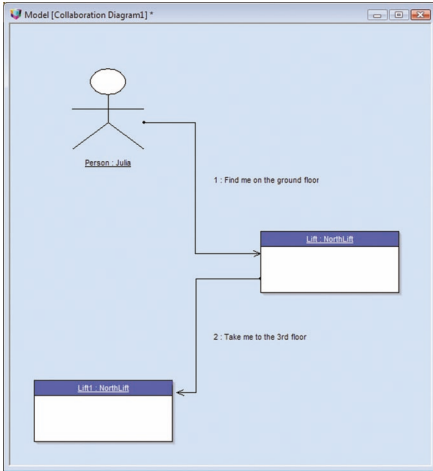
A sequence diagram contains the following elements:

- **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 instance: "Caller", "Called", ...
- **object activation period**: on the life line of an object you can insert activation period for the object. These periods represent the times when the object is active.
- **message**: shows, using horizontal arrows, the messages exchanged between the various 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 instance: "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 the role of an external element. A person for example.
- **message**: represents the messages exchanged between the different objects.

2.8 State-transition diagram

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

A state is defined by its duration and by its stability.

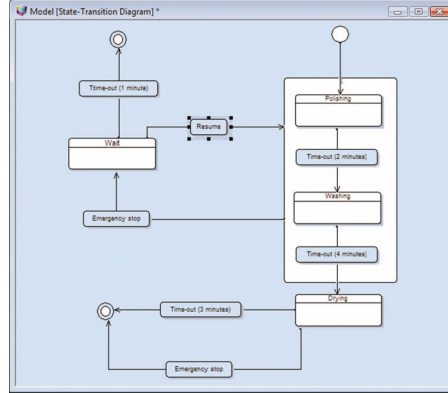
A transition represents the instantaneous change from one state to another one.

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 washing machine for cars:



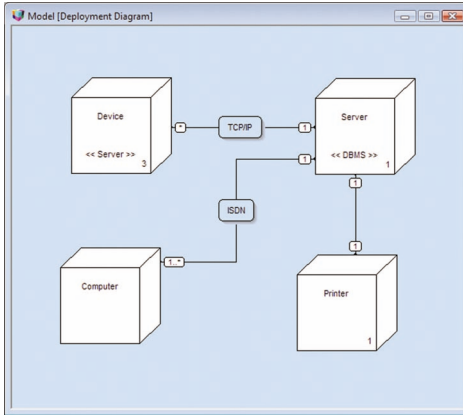
A state-transition diagram contains the following elements:

- **state**: represents the value of the object attributes at a given time.
- **initial state**: represents the state when the system starts.
- **final state**: represents the state in which the system is at the end of the operation.
- **super-state**: useful for structuring the diagram by specifying several levels between states.
- **history**: represents the last active state of a super-state.
- **stump**: used to symbolized the states found in a super-state. This enables you to link these states to other states that do not belong to the super-state.
- **transition**: represents the passage from one state to another.

2.9 Deployment diagram

A deployment diagram presents the physical layout of the hardware devices 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:

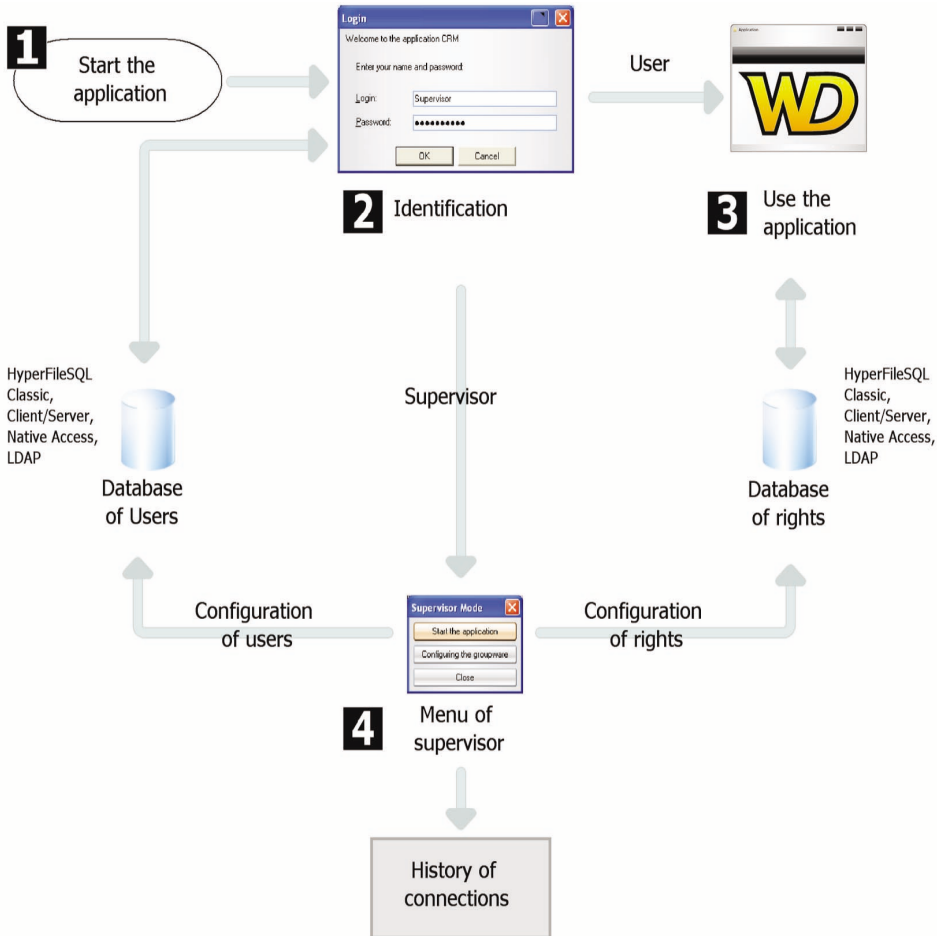
- **node class:** represents a class of hardware resource. For example: a server, a pc, a printer, ...
- **node instance:** represents a hardware resource. For example: server number 3, printer number 7, ...
- **connection:** describes the communication support between two nodes. For example: TCP/IP.

User Groupware

The User Groupware enables you 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 you to define the role of the different contributors. It is often necessary to define several access levels according to the users. Indeed, all the 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 a sales management application, the application proposes the following features:

- See the price list
- Modify the price list
- Enter the orders
- Enter the customers.

The accesses differ according to the user. Some examples:

- the secretaries can see the price list and create orders
- the salesmen can see the price list, place 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 work?

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: user groupware enables you to connect through a LDAP directory.

2.2 The supervisor level

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

The configuration window allows the supervisor to:

- start the application.
- configure the groupware.

Configuring the groupware consists in creating 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 128, for more details.

Note: all the information regarding the users and their rights are stored in HyperFileSQL format files. The files regarding the users can be common to several applications. The files containing the rights granted to the users on the various windows found in the application are specific to the application and they cannot be shared.

3 Implementing the user groupware

3.1 Adding user groupware to an application

To implement the user groupware in your WinDev application, one option is all you need: "Workshop .. Configure the user groupware". The window for configuring the user groupware lets you define:

- The management mode of the user groupware.
- The way to launch 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:

- **Included by default:** all the user groupware programming is automatically included in your application.

A new window is included in your project: the login window. It comes up when starting the application and allows the user to connect to the application. You can customize this window by applying the skin template of your application for instance.

- **Custom inclusion:** all the user groupware programming is automatically included in your application. **All the windows required to manage the user groupware (login, user management) are automatically added to your project.** You can customize all the windows used by the user groupware this way.

The various windows used by the user groupware are presented in details in "WinDev windows", page 130.

Starting the user groupware

Two modes are available for starting the user groupware:

- **Automatic launch:** the login window of the user groupware is displayed 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 launch:** the login window will only be opened if the `gpwOpen` function 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 instance.

Automatic login in test mode

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

LDAP directory

If the company of your customer uses an LDAP directory, you have the ability to branch the user groupware in 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 right management:** In this case no groupware 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 right management:** only 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 works with Active Directory. It does not operate with openLDAP.

User Groupware data files

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.
- **Data files used to manage the user rights** on the different windows of the application. These 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 instance after reindexing files or when opening files with the WDMAP tool.

See "The data files", page 130, for more details about these files and their configuration.

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, just 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, select "Manual start" in the User Groupware options. Use the `gpwOpen` WLanguage function to open the login window.
- The windows managing user groupware are run before the initialization code of the project.
- The window defined as the first window of your application in your project, will be run after the login screen (if the user is not the supervisor).
- If you do not want to launch the user groupware during the test of the application, simply clear "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 supports the user groupware ("Workshop .. Generate the executable"), a user groupware specific screen is displayed.

This enables you to define the location of the common files of the user groupware (files containing information about the users).

3.4 Installing an application that uses the user groupware

An application that uses the user groupware is installed the same way as a standard application via "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, the list of setup files must be customized.
- If the groupware files have not been configured, only the Supervisor user will be able to login when the application is started for the first time.

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

4 Configuring the application that 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 the supervisor:

- Name: SUPERVISOR
- Password: SUPERVISOR

2. Choose "Configure the groupware".

4.1 Managing the users

To create a new user, you must specify:

- the last 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 into 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's windows.

For each association between a user and a window, a specific state 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 out. No input can be performed.
- **Invisible:** the element is not displayed.

5 Tips for developing an application with 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* property)

When configuring the user groupware, the supervisor can define other 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 files when creating the application's setup program.
- 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 User groupware elements

6.1 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 data files of the user groupware", page 133.

Login window (GPWLOGIN.WDW)

This window is used by all the application users to connect to the application. In this window, the user must enter his name and password (optional) that were previously defined by the application supervisor. When starting the application, you can use the "SUPERVISOR" login and the "SUPERVISOR" password.

Supervisor menu window (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 as well as his personal details.

User and association configuration 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 (GPWConfiguration-Details.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 (GPWConfiguration-Choice.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 (GPWConfiguration-Form.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 history of connections (and to clear this history if necessary).

6.2 The data files

Two categories of files are managed:

- Files used to identify users. These files can be common to several applications.
- Data files used to manage the user rights on the different elements of the application. These 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 user groupware data files 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. Enter the new password in the "File" tab. The characters typed are displayed as stars.

Note: this password is used for example when reindexing files or when opening files with the WDMAP tool.

User data files

These files are the following ones:

GPWUser	File of users. Contains all the characteristics of the users (login, 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 files are presented in details in the online help.

Where are these files created?

- **By default**, these data files (HyperFileSQLClassic mode) are created in a sub-directory of your project (EXE\GPW_<Project_Name> directory). 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 program, you can allow the users to configure the location of these files ("Additional modules" pane). Therefore, a common directory can be selected for several applications that manage the user groupware.
- If "Manual Run" was selected, *gpwOpen* is used to start the login window of user groupware as well as customize the directory of the data files of 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 set, the user data files (HyperFileSQL Classic mode) will be created in a subdirectory named \GPW_<Project name> of the executable installation directory.

Right data files

These files are the following ones:

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, enables 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 files are presented in details in the online help.

Where are these files created?

- By default, these data files (HyperFileSQL Classic mode) are created in a sub-directory of your project (EXE\GPW_<Project Name> directory).
- When creating the setup program, you can allow the users to configure the location of these files ("Additional modules" pane). You have the ability to select a specific directory for the rights files of the application.
- If "Manual Run" was selected, *gpwOpen* is used to start the login window of user groupware as well as customize the directory of the data files of 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 set, the right data files will be created in a subdirectory of the executable's installation directory named \GPW_<Project name>.

7 Modifying 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 should 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 setup windows are only seen by the application supervisor.

Caution: the GPWLOGIN and GPWSupervisorMenu windows should not be renamed.

7.2 User groupware and multilingual application

If your application is a multilingual application, you need to manage this feature also 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 need to:

1. Configure the user groupware as follows:

- all the user groupware windows must be included in the application.
- the "manual launch" option must be selected.

2. Translate the various 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).

Note: WDMMSG is an optional tool of WinDev that can extract and reinsert all the messages to translate.

3. Run the application in the requested language. To do so, you can, for instance:

- Ask (if necessary) for the language in which the application must be run (via a specific window for instance).
- 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)
// Launch the first window
// of the user groupware
gpwOpen()
```

See "Multilingual applications in practice", page 135, 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 instance.

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 the code with *gpwOpen*.

7.5 Defining menu rights

To define the rights on the popup menus of a WinDev application, all the windows of user groupware must be included. You can then modify the window

for managing 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 should 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.
- right data files.

The location of these data files can be configured:

- when including the user groupware for running the tests.
- when creating the executable
- when installing the application (by the person that performs the setup)
- by programming.

See "The data files", page 130, for more details.

7.7 Reinitializing the data files specific to user groupware

To reinitialize the user groupware data files, simply delete the data files specific to user groupware (user file and right file). These files are automatically re-created when the application is started.

Note:

- If the user files are deleted (*GPWUserConfiguration* and *GPWUser*), only the supervisor will be able to start the application.
- If the rights files are deleted (*GPWElementConfiguration*, *GPWConfiguration* and *GPWElement*), all the users will have all the rights on the entire application.

See "The data files", page 130, for more details about the various types of data files managed by the user groupware.

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 files contain confidential information and are therefore encrypted.

7.9 Managing the analyses

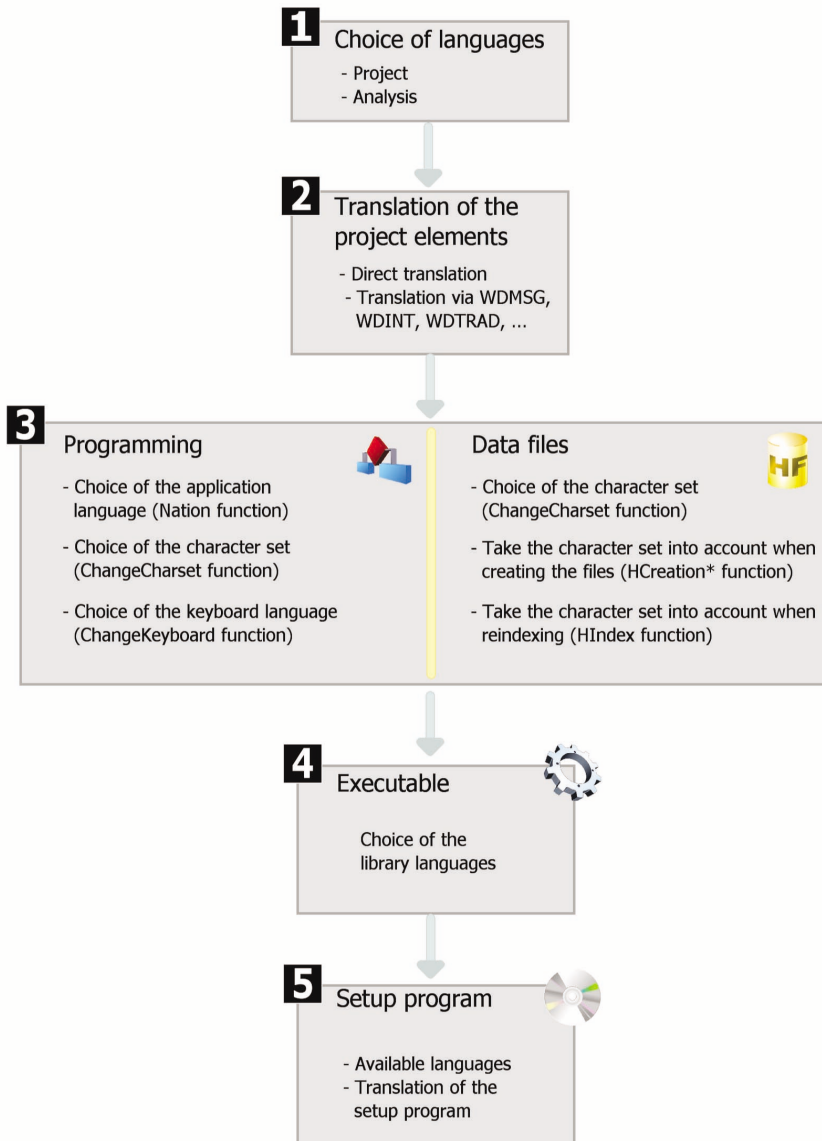
If a specific analysis is opened while configuring the user groupware, *gpwInitAnalysis* must be re-run (used to open the analysis of user groupware).

Note: at run time, you cannot open several analyses 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. Enter the different project elements (window, code, ...) in the different languages of the project.
3. Defining the project language by programming.
4. Managing the 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, and so on), the corresponding character set is automatically used when the translations are entered in these languages.
- If your application is a multilingual application, you need to manage this feature also in the windows of the user groupware and the windows of the automatic HyperFileSQL error management. See the online help for more details.
- Unicode management is available in the HyperFileSQL files and 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 offered for all the multilingual resources that can be translated (control captions, menu options, help message associated with a control, etc.).

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 for displaying numeric data (number, currency, date, time, duration, and so on), 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 language options also let you define the direction for writing and the character set used ("various" option).

Executable program

When creating the executable program, you can 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 enables you 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 in "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 will be automatically taken into account by all the languages common to the analysis and to the project.

- in the information printed in the program documentation (notes about the file or the item).
- in the information managed by "Reports and Queries". If the "Reports and Queries" program is provided with your WinDev application, you can translate the name of the 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 & Queries" tab of the file and item description.

2.3 Languages supported by the different project elements

By default, the different project elements (windows, reports, code, classes, set of procedures, and so on) support the same languages as the project where they were created.

An element can support more languages than the project (if the element is shared among several projects that support different languages for instance).

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 using 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 allow 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 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 language and the target language.

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, simply select the text to translate and click the "Translate" button.

Checking out and checking in the information to translate

The WDMMSG tool, available separately, lets you extract all the messages of a project (control captions, etc.) and check them back in after they've been translated.

Contact PC SOFT Sales Department for more details about the conditions for using this product.

3.2 Languages that use a specific character set

If your application supports languages that use special character sets (Greek, Russian, and so on), you need to enter the translation of the various messages 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 specific character set, the corresponding input language (character set used by the keyboard) is automatically displayed.

Therefore, if a caption is entered in the Russian section of the caption description, the keyboard will automatically switch to the Russian character set.

Reminder: to use special character sets, you need to install the files corresponding to the character sets you want in the regional options of Windows (control panel).

3.3 Translating specific windows

Management of 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 example in the `\Programs\Data\Preset Windows\HyperFileSQL - Automatic Help Windows`.
2. Customize the error management to use the HyperFileSQL windows for error management (`HOnError` function).
3. Translate the messages (see the previous paragraph).

Automatic error management

By default, the windows for automatic management of errors are supplied in English and in French. To translate them into another language, you must:

1. Include the default error windows in your application. These windows are supplied (along with their WLanguage code) as examples in `\Pro-`

`grams\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 including user groupware 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: if needed, clear the "automatic launch" option, this way you will be able to display a language selection window 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" * Uj)
```

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 convert all the messages in your code into multilingual messages, select "Code .. Multilingual messages .. Convert the 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
- have an option (menu option for instance or button) allowing the user to change the current language for the application

5 Managing specific character sets in the HyperFile SQL data files

When an application is used to display or save data in HyperFileSQL 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.

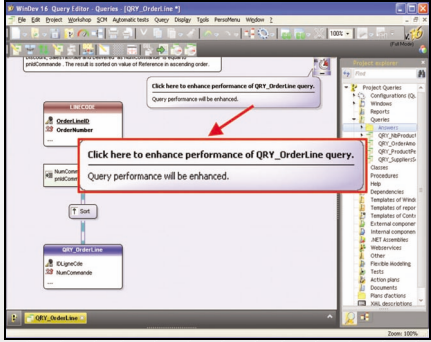
If the application is used in a country that uses a special character set, this character set must also be supported by the data files. Therefore, the searches, the sorts, ... will be performed according to the selected character set.

Optimizing an application

WinDev offers several methods for optimizing your applications:

- **The Aided Application Development (AAD)**, which suggests code or interface improvements when developing the application.

Developing the application

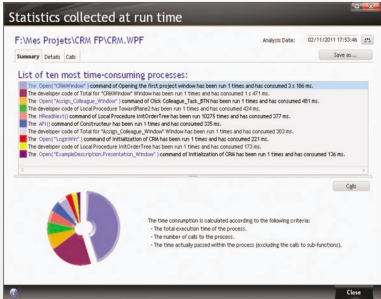


The screenshot shows the WinDev Query Editor interface. A query tree is visible on the left, with a callout box pointing to a node labeled 'QRY_OrderLine'. The callout contains the text: 'Click here to enhance performance of QRY_OrderLine query. Query performance will be enhanced.' Another callout points to a different part of the query tree.

Optimizing your queries, improving the anchors in your windows, ... These suggestions are made on a regular basis while you develop your applications.

- **The profiler** that is used to identify the time-consuming processes. Then, all you have to do is optimize the requested process.

Testing the application



The screenshot shows the 'Statistics collected at run time' window. It displays a list of the ten most time-consuming processes, including details like the command executed, the number of times it was run, and the total time consumed. A pie chart is also visible at the bottom left of the window.

Analyzing the performance of the application while a test is running, viewing the longest processes to optimize the source code.

To start the profiler, select "Project .. Performance profiler .. Analyze the performance".

- **The project Static Audit**, which notes the potential programming problems (empty groups, orphan elements, etc.) as well as the large files that may require long load time.

Static audit

Audit report indicating the problems found, analysis of the library (to detect large files) and analysis of the installation pack.

To start the static audit, use "Project .. Edition audit .. General report".

- **The Project Dynamic Audit**, which will detect the loss of resources, the unprocessed errors, the application slowness in the production environment.

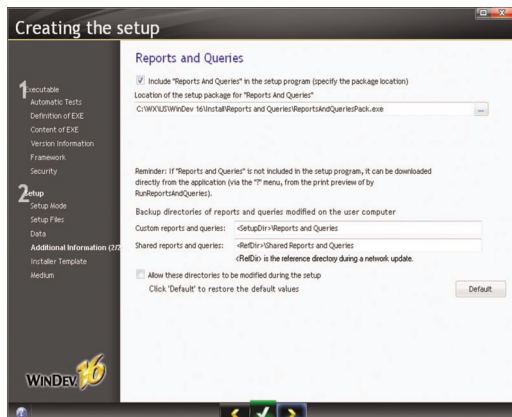
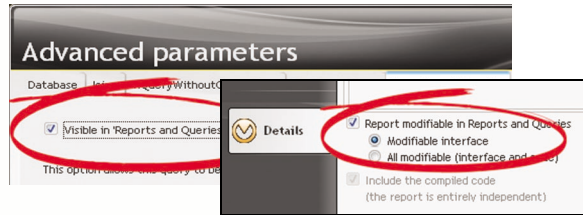
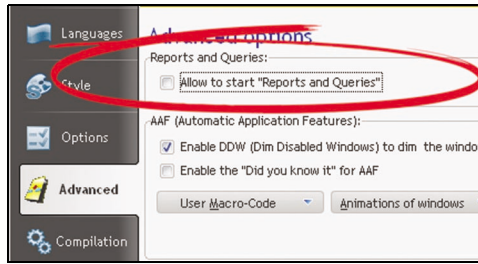
Dynamic audit

Analyzing the performance of the application while a test is running, viewing processes containing exceptions or errors.

To start the dynamic audit, use "Project .. Test mode .. Debug the project while the audit is enabled".

Distributing "Reports and Queries"

"Reports and Queries" (can be distributed along with your applications) allow the users to customize reports according to their own requirements. Of course, only the reports you have chosen will be customizable.

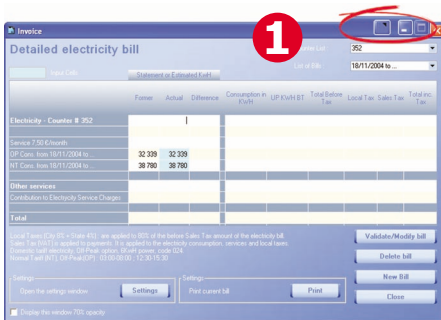


UMC: User Macro-Code

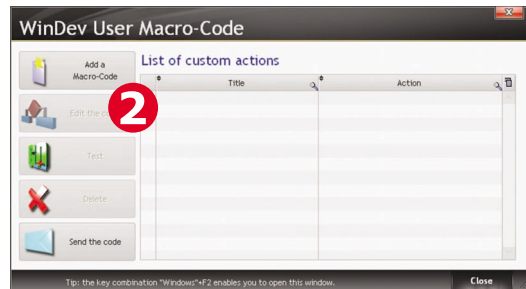
The User Macro Code (UMC) allows end users to enter their own WLanguage code directly and include it in an existing application.

Let's consider the following case:

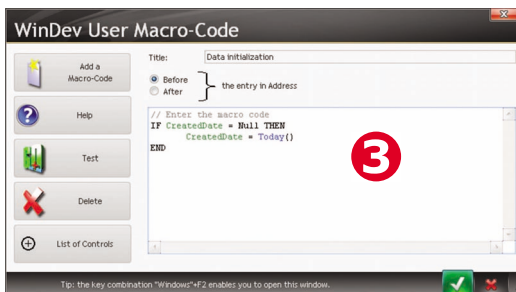
A window for entering a customer form requires the presence of an email address but it does not check its format. The user wants to check the presence of the "@" character in the address.



An icon found at the top of the window allows the user to customize the processes.



The window for configuring the UMC is displayed. A new action can be added by the user for this window. In this case, the email address is checked.



The user 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 UMC?

UMC is an option allowing advanced users to perform 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 they can be added to the application's source.

A large number of WLanguage features are accessible to the user and some restrictions can be set by the developer. For instance forbid the use of any file management functions. In order to insure data integrity, the record modification or addition commands are not available to the end user.

2 How to enable/disable UMC?

You can of course disable this feature for a window or for the entire application if you want.

To disable UMC for a window, select the "Forbid the use of U.M.C" option in the "Details" tab of a window's description.

When creating the executable you can authorize 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 as an attachment. This way you can easily retrieve suggestions from your customers and include them in your project automatically! WinDev offers a "*.UMC" file import option ("File .. Import .. User Macro Code ...").

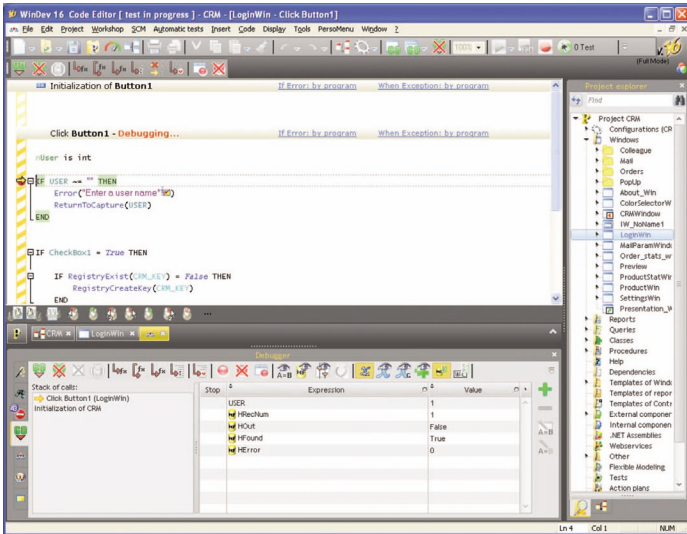
The various tests available

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 	<p>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.</p> <p>As soon as a problem occurs in the execution of the project, the debugger can be started to identify and fix the problem.</p>
Testing the current element 	<p>Used to run the test of this element as soon as its development is completed.</p> <p>The debugger can be run as soon as a problem occurs in the reports or windows.</p>
Testing a project step by step (via the debugger)	<p>Used to run the debugger when the application is started. This solution enables you to monitor how the application runs via the debugger.</p> <p>You also have the ability to debug a specific executable, an executable already run or an executable run on another computer.</p>
Unit test	<p>Used to run the test of windows, procedures and classes found in an application at different levels of development.</p> <p>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 directly modified in the code editor.</p>
Regression test	<p>Used to run the regression test of an element between the different versions of an application.</p> <p>To do so, select (or create) the test scenario that will be run.</p>

The debugger

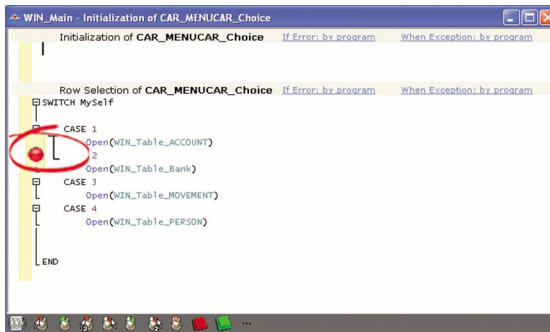
To help you optimize your source code, a debugger is supplied with WinDev. The debugger allows you to run your applications step by step, by viewing the running WLanguage code, the content of the variables, ...



Debugging the project

1. Using a breakpoint, defined in the code editor.

The debugger will be automatically started when the line preceded by a breakpoint is run. To insert a breakpoint, click in front of the code line: a red bullet is displayed.



2. Using the STOP keyword in the WLanguage code

The debugger will be automatically started when this code line is run.

3. Selecting "Project .. Test Mode .. Trace the project"

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 running on the development computer.
- be started from WinDev, on a remote computer.
- be already running 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. This enables you to run the test of the component 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

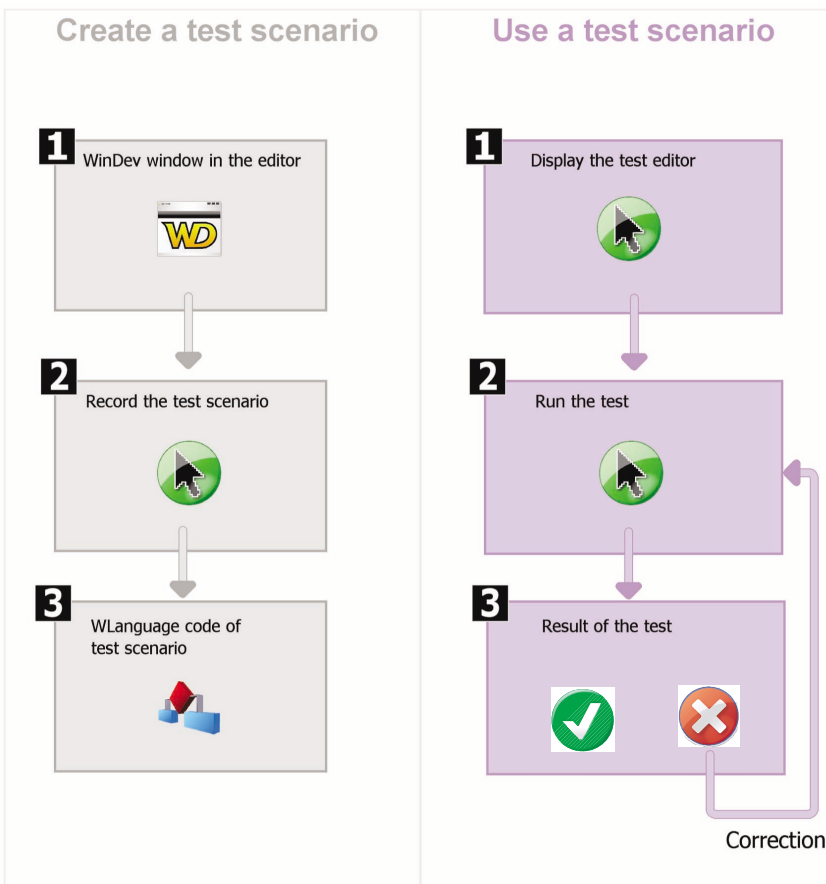
The unit tests (also called automatic tests) are used to run the test of windows, procedures and classes found in an application during the entire development. To do so, select (or create) the test scenario that will be run.

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 level of the application.
- indicates the modified elements whose test was not run.

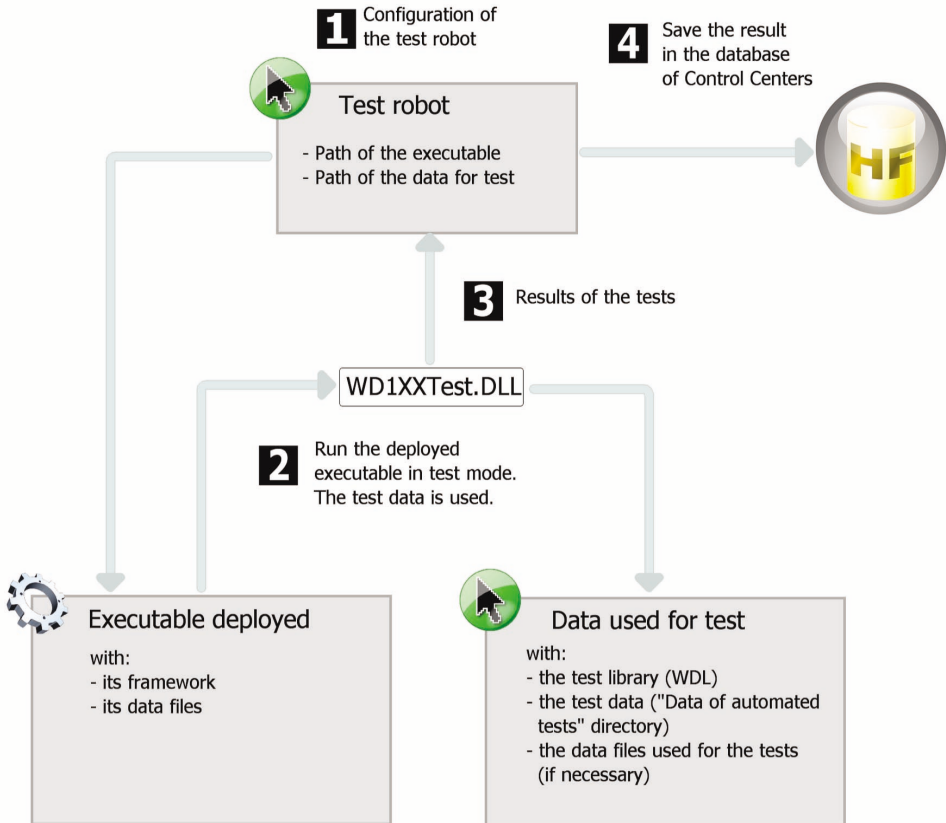


To create a unit test on the current window, click .

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.



Running the 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" book for more details),
- test of a single report (see the "Reports and Queries" book for more details),
- running the project step by step,
- testing your application's performance.
- regression and automated tests

testing the entire project allows you to simulate the running 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 can run the debugger to find out and fix the problem.

Testing a window lets you 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. As for

the test of the project, the debugger can be started as soon as a problem occurs.

Testing a query only lets you run only the current query. This enables you to check the operating mode of a query as soon as it is developed.

Testing a report only lets you run only the current report. This enables you to check the operating mode of a report as soon as it is developed. As for the test of the project, the debugger can be started as soon as a problem occurs.

Executing the project step by step enables you to start the debugger when starting the application. This solution enables you to monitor the progress of the application.

The performance test of your application lets you check and optimize the execution time of your application.

The regression test (or automated test) is based on running scripts. It lets you check during your application's execution, ... the existing features are always 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 project test can be run regardless of the current element in the editor.

2.2 Running the test of the current project and debugging it

To test a project from the editor, select "Project .. test mode .. debug the project" ([Ctrl] + [9] 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 when debugging.

This test mode is used to test 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).

The principle is straightforward:

- The developer starts the executable on his computer and goes directly to the location he wants in the application.
- In WinDev, the developer opens the project corresponding to the executable and requests to be attached to the running executable.
- The WinDev debugger lets you follow the program's 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 when debugging.

See the online help for more details.

3 Running the test of a WinDev project and debugging it on the user computer

WinDev offers several methods for testing a project and 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.

Two features are available:

- Running and debugging the executable on a

remote computer

- Debugging an application currently running on a remote computer.

For these two features, a specific configuration of the remote computer is required.


Note: to remotely debug a WinDev application, the WinDev application must include the WD160CPL.DLL library in its framework.

See the online help for more details.

4 Testing a window

4.1 Testing the window from the editor

To test a window from the editor:

1. Open the window to test.
2. Select "Code .. Test the window" ( 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 instance.

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 running. WinDev displays the editor that was used when the test was started.
- **2nd method:**
 - Go back to the editor with the taskbar or with [Alt]+[Tab].
 - Confirm the stopping of the test. WinDev displays the editor that was used when the test was started.

5 Trace a project

5.1 Debugging principles

Debugging an application consists in:

- checking the operating mode of a process,
- understanding the operating mode of an existing program,
- checking the value of the variables,
- checking the operating mode of the particular cases in an application.

The debugger enables you to perform all these operations.

Note: WinDev also offers you several trace tools (trace window, information box, ..). See the “Debugging without the debugger”, page 151, for more details.

5.2 Overview of the debugger

The debugger lets you trace your WLanguage programs in order to help you fine tune them.

The source code run is viewed on the screen. The executed processes are sorted in the "Debugger" pane.

The values of the variables can be viewed:

- individually in the rollover tooltip of each variable.
- in the "Debugger" pane.

5.3 Debugger features

The debugger enables you to:

- get the call stack
- view the content of the variables or expressions
- run code step by step with the ability to skip some blocks.
- use conditional break points
- 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 change things.

Indeed, the debugger introduces pauses in the execution of the program during which various tasks are performed by WinDev.

This is why the debugger cannot be used in a procedure called by a timer or in the code of a scrollbar.

Note: to find out all the debugger's limits, see the online help.

To debug these applications, you may want to follow the value of a variable, how 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 should only be displayed during the test of the application.

Displaying some information

Two tools enable you to display information:

- **the information boxes:** *Info* function of WLanguage.
Caution: the display of an information box is a locking operation.
- **the trace window:** *Trace* function of WLanguage.
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 debugging 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 OnTestMode() THEN
    Trace (StringToTrace)
  END
```

In this code, depending on the result from the *InTestMode* function, the trace window only appears when testing the application.

Such a 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).
- or in a text file (default mode).

```
PROCEDURE MyTrace(StringToTrace)
File is int
DebugMode is boolean = False
IF Position(CommandLine(),...
"/DEBUG") > 0 THEN
    DebugMode = True
END
If DebugMode THEN
    Trace(StringToTrace)
ELSE
File = fopen(...
"C:\Trace.txt"+,...
```

```
foCreateIfNotExist+...
foWrite+ foAdd)
IF File <> -1 THEN
fWriteLine(File,...
DateToString(DateSys()+...
"->"+TimeToString(...
TimeSys())
fWriteLine(File,...
StringToTrace)
fWriteLine(" ")
fclose(File)
END
END
```

Notes:

- The trace file: "C:\Trace.txt" is created by default. 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 then in order to analyze the reasons for its processing time and to optimize it.

6.2 Starting the performance profiler

To start the profiler, select "Project .. Performance profiler".

The project is automatically executed in test mode then. 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.

The performance profiler displays the result of the analysis.

Note: we recommend you use the performance profiler to optimize your application (before distributing it for instance).

6.3 Reading the result of the performance profiler

The performance profiler presents the result of the analysis in two tabs:

- the "Summary" tab presents the 20 longest processes.
- the "Details" tab presents all the processes started while testing 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 executed outside calls to a WLanguage function or calls to a custom function or procedure.
Parent	Process that called the function.

Note:

- The "Full execution" caption represents the total amount of time for running the test of the application with the performance profiler.

- The "Total Window XXX" 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 ensure the quality of your application:

- The test mode (Go from a project or from a window) is used to immediately test a change in your application. These tests can be saved as automated tests.
- The WDTTest tool, which allows you, for instance, to create validation and regression test in WinDev.

To automate these tests and to increase the quality of your applications, you have the ability to run **automated unit tests**. These tests allow you to check all the features offered by your applications.

7.2 Automatic tests

Each test contains a scenario that can be directly edited in the product interface. This scenario is written in WLanguage and it can be modified at any time.

These tests can be run, for instance, before creating the executable in order to check the operating mode of an application.

The following elements can be tested:

- WinDev windows.
- the sets of procedures.
- 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 the associated code are not included in the executable and they are not distributed along with your application. The number of tests for an application has no incidence on the size of the application supplied to your customers.

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

7.3 WDTTest

WDTTest is a tool that allows you to record and automatically run application test scripts (for WinDev or not).

A test script (also called "macro") contains mainly the different operations performed on an application (mouse movement, mouse click and keyboard entry).

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 operate properly between two updates.

WDTTest automates application test procedures.

Important: To successfully perform a test, you need to use the exact same configuration:

- on the computer where the test is created
- on the computer where the test is run

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



PART 5

Setup



DEVELOP 10 TIMES FASTER

PC SOFT

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 30MB) 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 have the ability to replace one or more DLLs (if a specific version is required for example).

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.

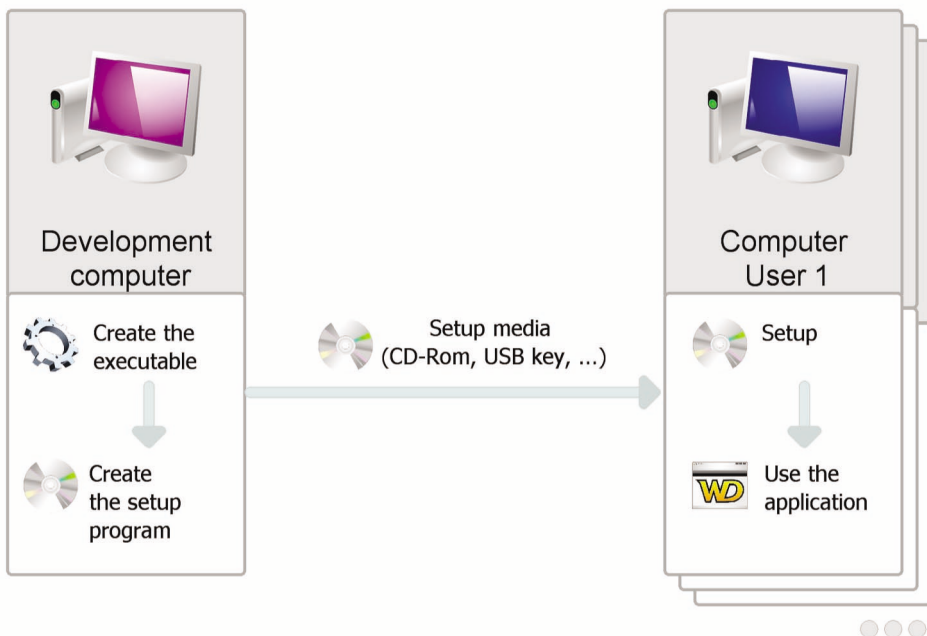
Benefits of the custom framework:

- 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.

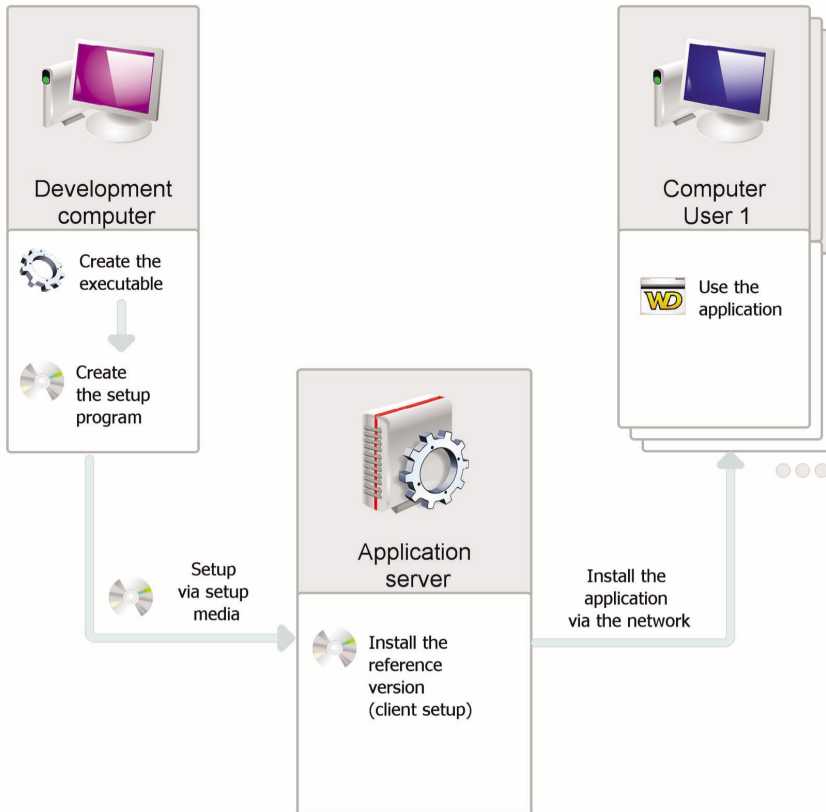


This setup mode is compatible with:

- 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.)

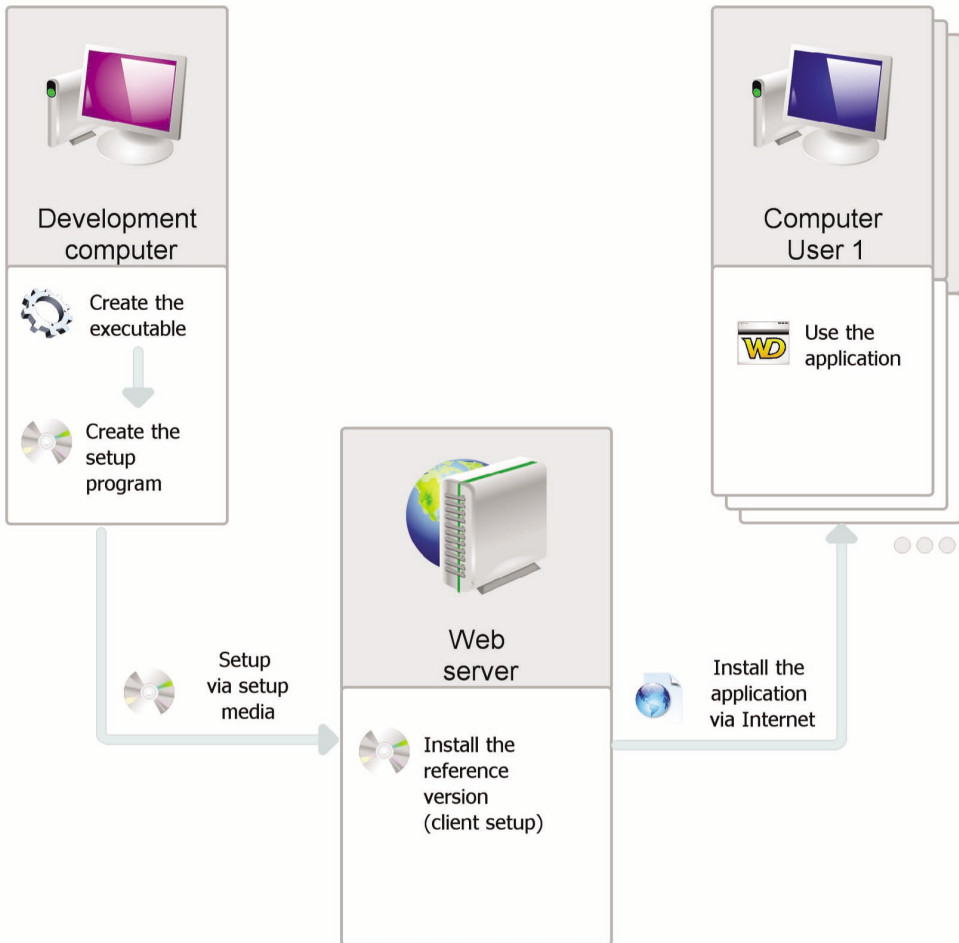
This setup mode is compatible with:

- 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

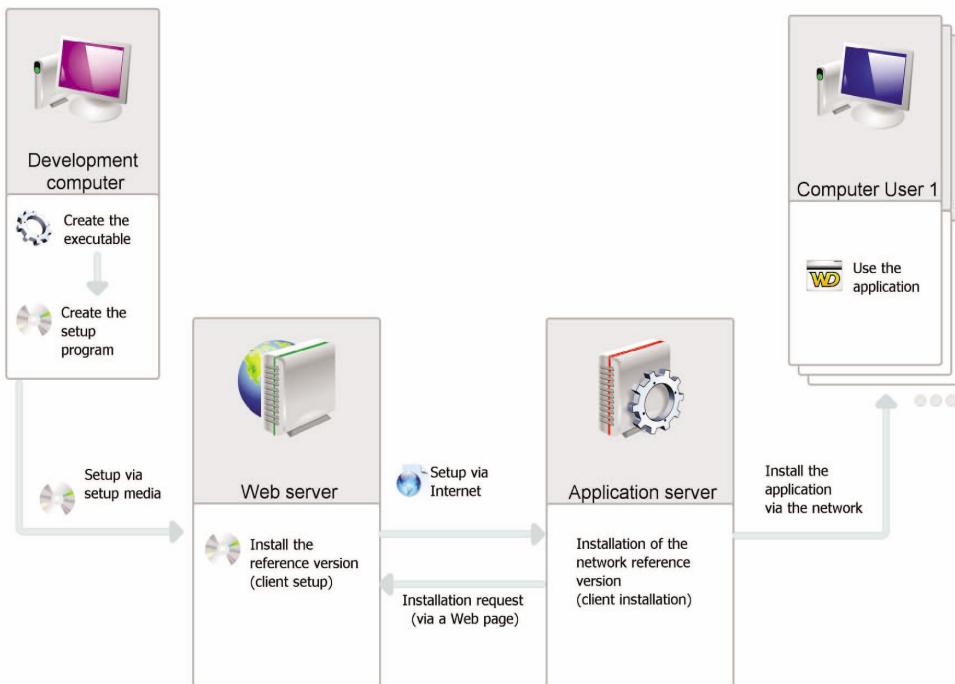
The multi-site setup allows you to install an application on several different sites (a site can represent a subsidiary for example).

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.

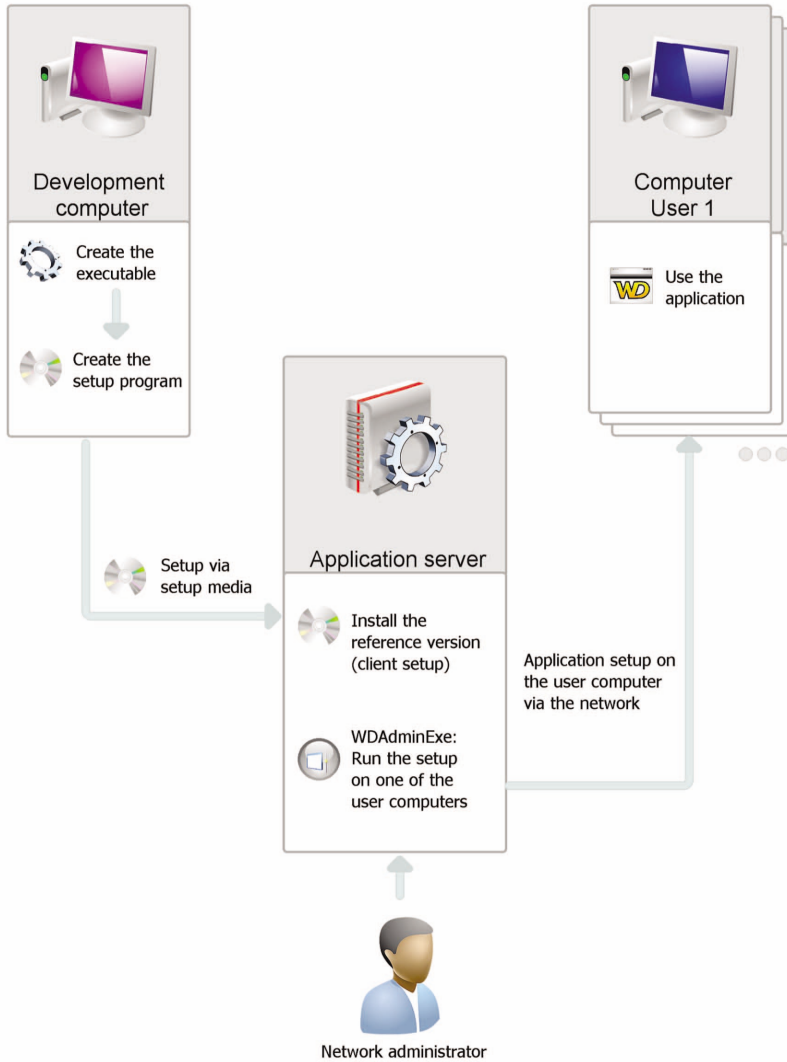
This setup combines the benefits of the network setup and HTTP setup by allowing a large-scale deployment as well as a granular control of the local setups.



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 WDAAdminEx.



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			
		Executable Self-Extractable	Directories to the size of diskettes	Directory unique	CD DVD
Supports for setup	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 WDCONFIG
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 instance)
Startup mode	"Workshop .. Create the setup procedure"	"Tools .. WDINST"	WDCONFIG project available in the "Examples" sub-directory of the setup directory of WinDev.

1.3 The different types of setup

Two types of setup are available in WinDev:

- **Stand-alone setup:** recommended for the single-computer applications.
The setup program installs the application on the user computer.
- **"Network setup with management of users":** recommended for the multi-computer applications.

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 an 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 instance:
 - 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 from WinDev ("Workshop .. Create the setup procedure"). This wizard enables you to create the different types of setup programs.
 - use WDINST ("Tools .. WDINST - Setup editor"). This editor enables you to customize the setup program.
- **supply some other files**, you must use the wizard for creating the setup program started from WDINST ("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 automatically starts.
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.

Monitoring your applications

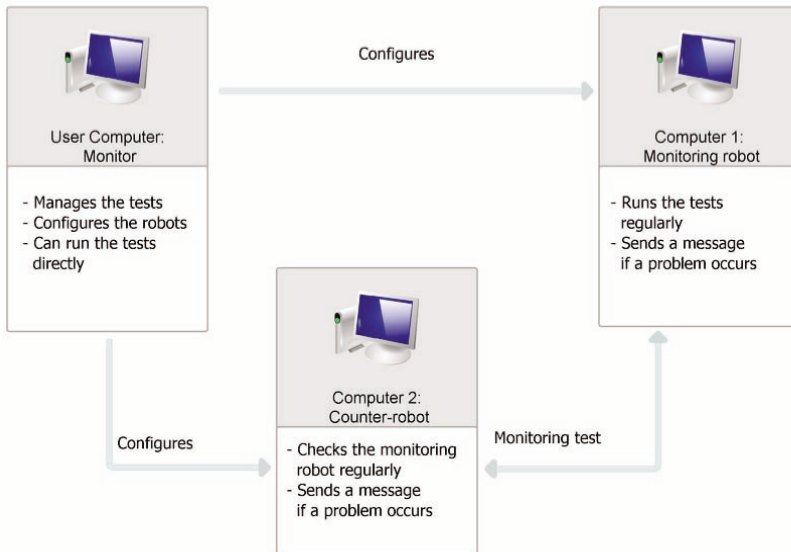
An application depends on several external parameters: Web site, HyperFileSQL server, network, ... To optimize the management of the incidents, WinDev proposes to use a monitoring robot.

Made of three executables started on different computers, the monitoring robot is used to run different tests: check of 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 using PC SOFT messaging (WDBAL).
- Email
- Running another program.

Furthermore, a sound alert can be implemented on the monitor.



Available types of tests

- | | | |
|---------------------------------------|---|-------------------------------|
| Internet test
(test of a Web site) | Test of HyperFileSQL
Client/Server servers | Network test
(SNMP) |
| Test of FTP server | Test of news server | Test of WinDev
application |
| Custom test
(in WLanguage) | ... | |



PART 6

Communication



DEVELOP 10 TIMES FASTER



PC SOFT

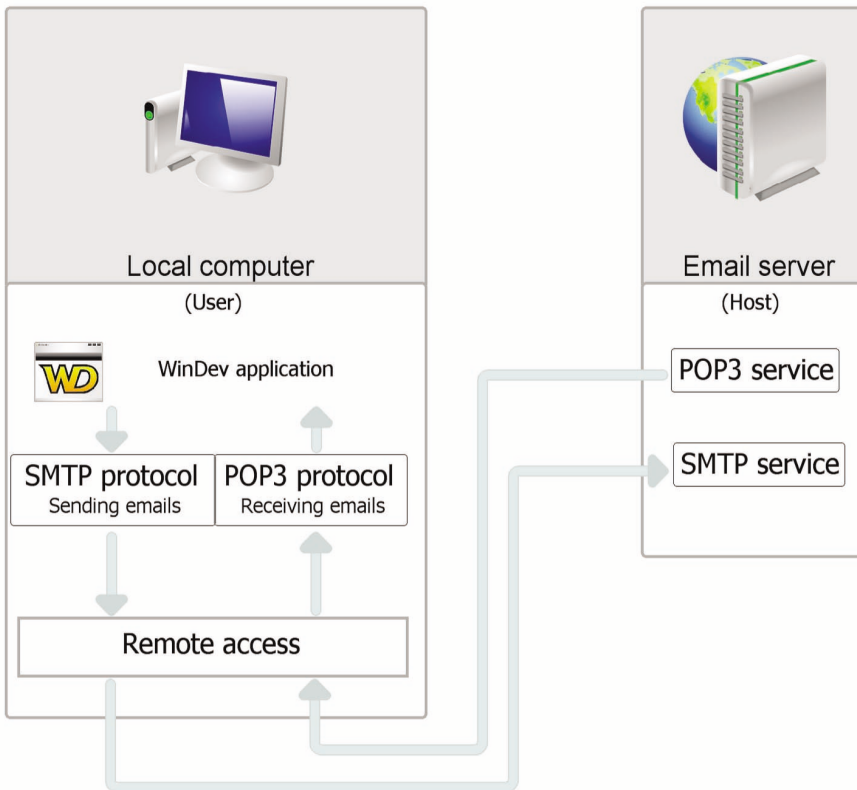
Managing Emails

WinDev proposes several methods for managing emails:

- the POP3 or IMAP protocols and the SMTP protocol (most common method)
- the "Simple Mail API (also called SMAPI or Simple MAPI)"

POP3 or IMAP and SMTP protocols

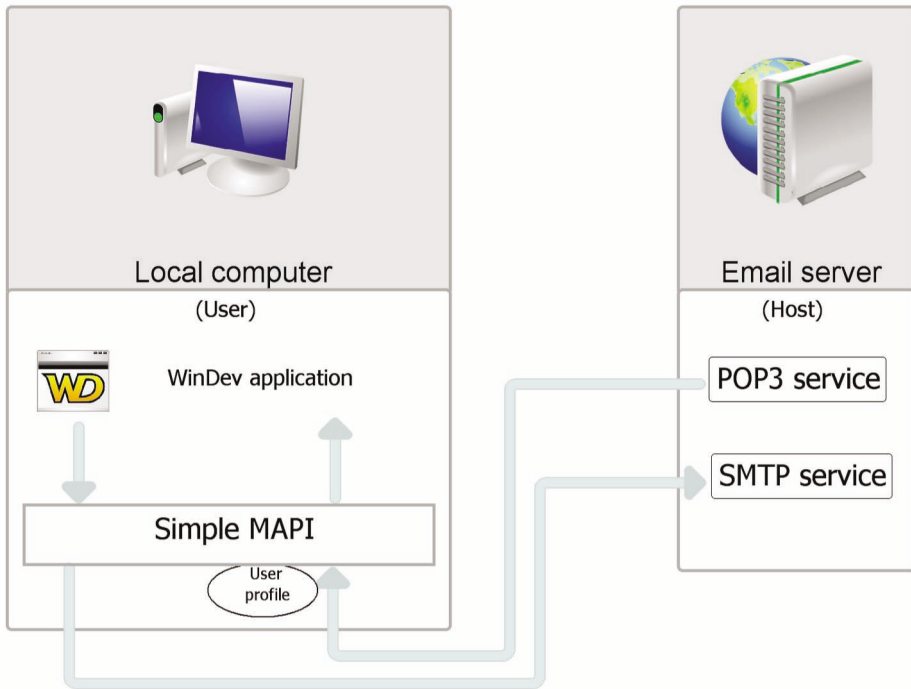
These protocols are the standard Internet protocols for email management. They are standardized by the RFC 1939, 3501 and 5321 documents. These protocols allow you to directly communicate with the server, available at your ISP.



- POP3 (Post Office Protocol) is a protocol for reading emails.
- IMAP (Internet Message Access Protocol) is a protocol for reading emails remotely.
- SMTP (Simple Mail Transfer Protocol) is a protocol for sending emails.

"Simple Mail API (also called SMAPI or Simple MAPI)"

This email management mode is used by most of the Microsoft applications and mainly by MS Exchange.



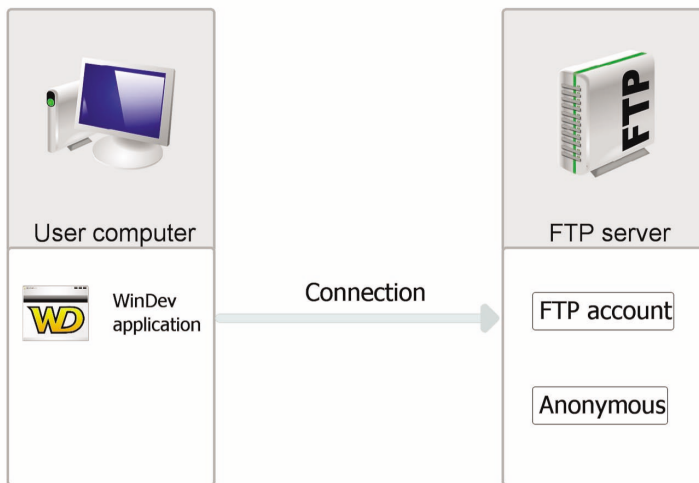
Transferring files by FTP

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

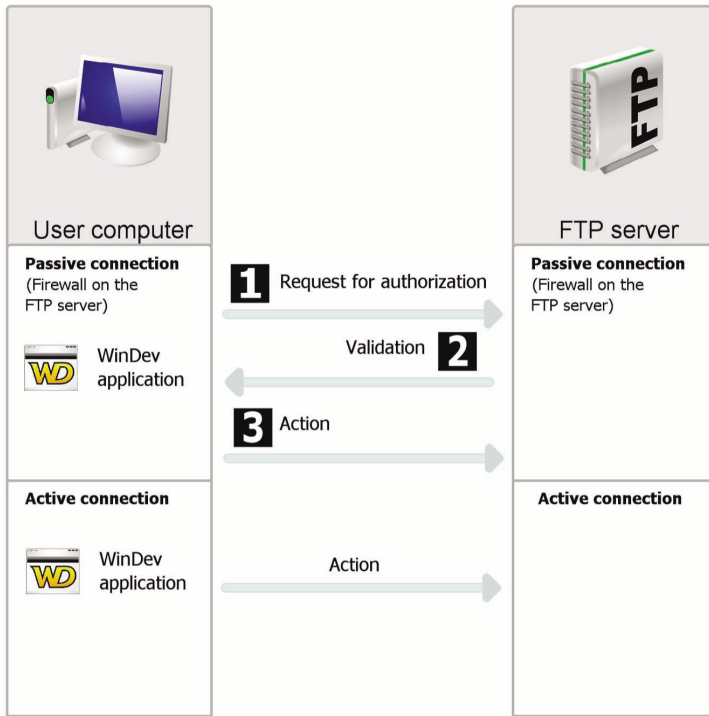
To handle the files found on an FTP server from a WinDev application, you must:

1. Connect to an FTP server (**FTPConnect**).

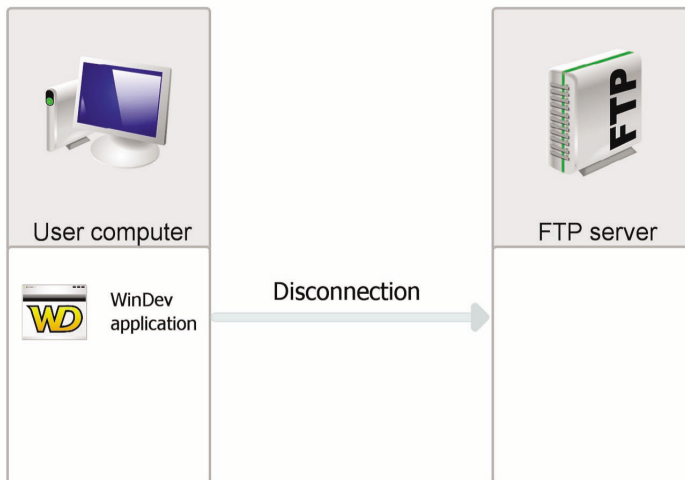


2. Transmit and retrieve some files (according to the connection mode). For a passive connection, the WinDev application must request the authorization from the FTP server before any operation is performed on the server files. You can also:

- 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. Disconnect from the server (*FTPDisconnect*).



Managing sockets

WinDev includes 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 instance, to communicate between computers connected via Internet.

Different possibilities

A WinDev application can manage sockets according to different modes:

- **Client WinDev application:** the application connects to any 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 socket of the server, 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

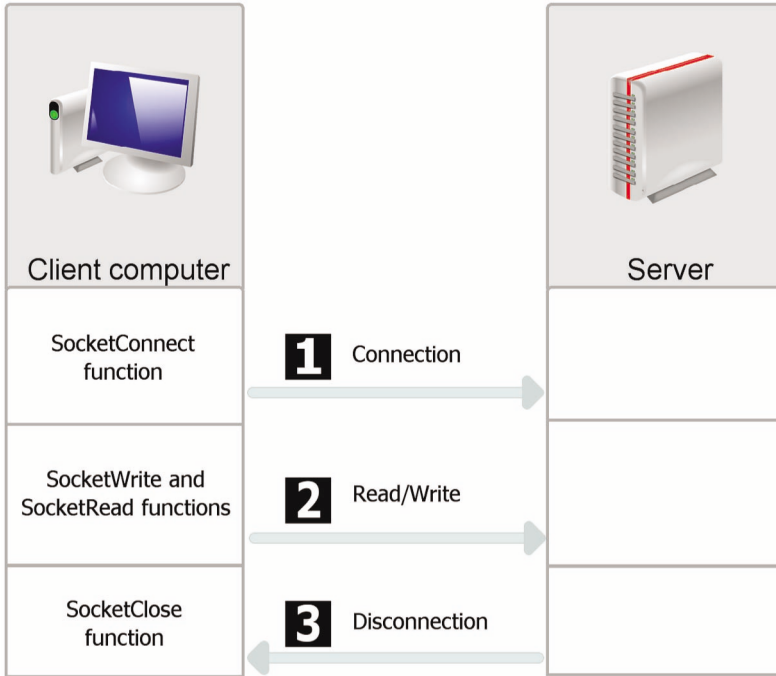
Once two computers have connected their socket, a communication stream is established between these two computers. These two computers can read and write some character strings on the socket.

To read and write on the server socket, the WinDev client application must use **SocketRead** and **SocketWrite**.

Step 3: Ending the communication

To end the communication, all you have to do is close the socket from the client computer with **SocketClose**.

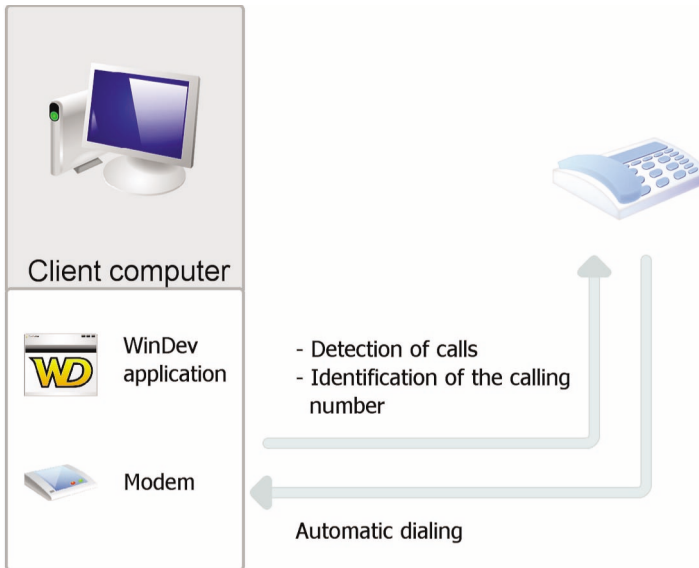
The different steps can be represented as follows:



Note: the SocketXXX functions can also be used to manage secure sockets according to the SSL protocol. To do so, use **SocketCreateSSL** and **SocketConnectSSL**.

WinDev and telephony

WinDev allows you to easily manage the incoming and outgoing phone calls via the telephony functions. These functions are used to manage a voice box, an answering machine, ... in a WinDev application directly.



Required configuration:

To be able to use the telephony features, you must have:

- a modem.

To save and play some recordings, this modem must have voice support.

To identify the caller, the modem must support "Caller ID" (caller ID).

- the TAPI 2.0 protocol.

Note: This technology can be used on all the systems; however, you must:

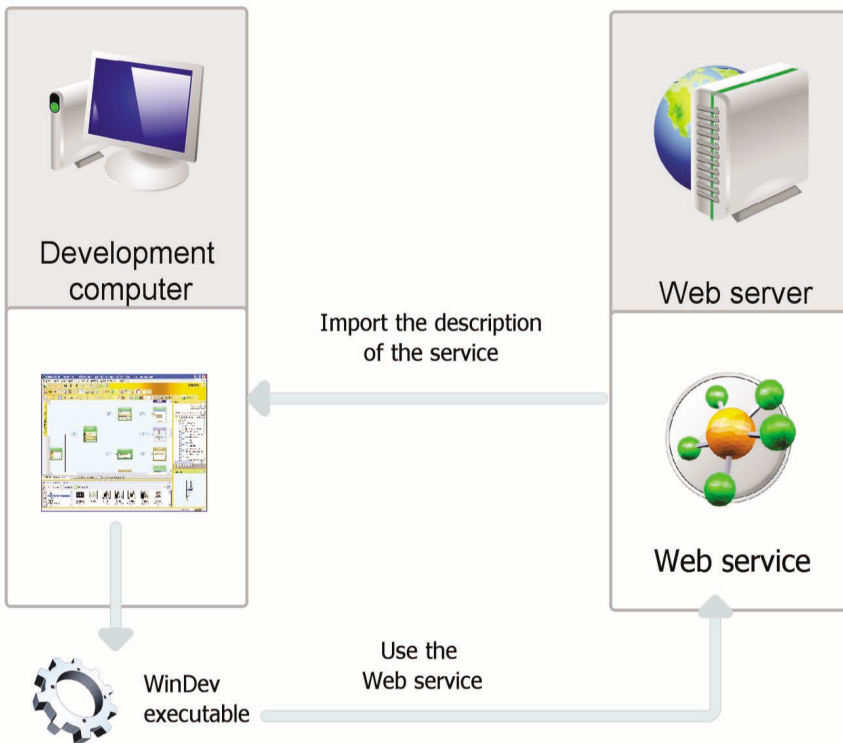
- install the service pack 4 (or later) if the system used is Windows NT4,
- upgrade the system if the system used is Windows 95 (can be downloaded from <ftp://ftp.microsoft.com/developr/tapi/tapi2195.zip>).

(note: address valid at the time this documentation is being written)

Webservices

WinDev allows you to easily use 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 also have the ability to expose one or more sets of procedures found in a WinDev application as 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



DEVELOP 10 TIMES FASTER

PC SOFT

Components supplied with WinDev

WinDev is supplied with several components. These components come with their source code and with an example.

Here is a list of the main components supplied with WinDev:

Name of the component	Description
Agent	Enables you to manage an agent.
Assisted directory input	Enables the assisted input for a directory path or for a file path in an edit control.
Assisted email input	Provides some assisted input when the user saves an email.
Balloon tip	Enables you to manage the display of a BalloonTip (advanced tooltip, displayed above the taskbar).
BankHolidays	Used to: <ul style="list-style-type: none"> - display a calendar and select a date - manage the public holidays in a calendar
Calculating HyperFileSQL statistics	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.
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.
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 from the different sources (preset tips in the code or from a file).
DisplayImage	Enables you to write a note in an image (screen shot for instance). You have the ability to insert some text, lines, etc.
Editable treeview	Enables you to directly rename the different elements found in a TreeView control.
EventLog	Used to: - List and read the events saved in the event log of Windows. - Add an event into the event log of Windows.
Explorer bar	Used to manage an address bar similar to the address bar of the explorer found in Windows Vista.
Extended Table	The extended tables (also called "TableView") are used to display some 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.
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.
Fax	Enables you to manage the faxes (operates in Windows 2000 and Windows XP only).
Floating help	Enables you to display temporary information to the user without interrupting his work.
Gantt chart	Enables you to create a Gantt chart from a task list.
Glossary	Used to organize and store the written expressions currently used by a customer. This enables you to propose an access to a "Glossary" that can be easily customized by the customer himself. All the glossary entries can be organized in tree-like categories.
GMT SOAP	Enables you to retrieve the system time and date of a remote computer via the SOAP protocol.
GraphicList	Enables you 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.

Limitations	<p>Enables you to limit an application by different methods:</p> <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	<p>Used to define, for an application, a configuration for the runtime logs:</p> <ul style="list-style-type: none"> - number of log files that must be kept, - maximum size for each log file, - information that must be logged. <p>This example simulates different processes to observe the management of log files (.wlog files).</p>
Memory zone	Enables you to view the entire content of a memory zone in a window. Also enables you to modify and add some values into the memory zone.
Photocopier	Enables you to photocopy some 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.
Printer spooler	Defines the list of less used printers in order to send them one or more print jobs.
Printout model	Enables you to manage the print profiles.
RTF search	Enables you to perform all types of searches in an RTF control.
Scanner	Enables you to digitize a document via a user-friendly interface.
Search engine	<p>Enables you to index documents (text, videos, sounds, ...) according to an identifier managed by the user (may 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>

Secured payment	Enables you to perform online bank transactions such as: - making a request for a debit by credit card - making a payment by credit card - canceling a bank transaction - making a request for direct debit - ...
Sending email	Used to: - create some preset messages for the next outgoing emails - create some contacts and lists of recipients - create some messages and send some emails
Sending SMSs	This component is used to send some SMSs from a simple Internet connection.
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.
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, ...
Simplexe	Used to perform some calculations from the algorithm of the Simplexe.
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.
Tactile Screen	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.
WIFI	Used to: - list the network cards available on the current computer - list the available WIFI networks - find out the quality of the WIFI signal - get some information about an existing WIFI connection

Two sub-directories are specific to each one of these examples:

- the "<ComponentName>-Example" sub-directory contains the example of a project that uses the component,
- the "<ComponentName>-Source" sub-directory contains the component project.

Some additional components are supplied with the Technical Support Newsletter (LST) or they can be downloaded from our site (www.windev.com).

Available examples 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.
- Teaching 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 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 .NET WMI	The WMI (Windows Management Instrumentation) is used to exhaustively manage the applications and the devices. WMI can be compared to a database that groups various information about a computer. 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 "standard" application).
WD Active .Net Directory	Enables you to view the content of an Active Directory.
WD Answering Machine	Uses the telephony features to manage an answering machine. This answering machine can be interrogated and configured remotely via a menu.
WD Burner	Enables you to burn audio CDs and data CDs, DVDs or Blue-Ray disks.
WD Burner Lite	Enables you 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 Email	This application is a full email client developed in WinDev. It is based on the Email objects. This email client is used to send and retrieve some emails by using the POP, IMAP and SMTP protocols.
WD Euro Calculator	Presents the multi-currency conversion via a calculator.
WD FTP Threads	Enables you 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 some 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	Full management of holidays and flex time for the company employees. Each user can display his own dashboard presenting the status of his holidays and flex time.
WD HTML Album	Enables you 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 Instant messaging	Presents the different functions for managing sockets with WinDev.
WD LDAP Explorer	Used to view the content of any LDAP directory and to modify the LDAP data.
WD Loan	Enables you to simulate loans and to: <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 For each case, you have the ability to print the depreciation schedule corresponding to your parameters.
WD Magnifier	Enables you to zoom a section of the screen via a magnifier. The zoom value can be multiplied by 8.

<p>WD Manager of .Net tasks</p>	<p>Enables you to list the applications and the processes via .NET functions. Enables you to perform different processes on the applications.</p>
<p>WD Managing a computer fleet</p>	<p>Enables you to manage a computer fleet. For each computer, you have the ability to define its characteristics, its components and its users. These features are implemented via the use of queries, reports and combo boxes.</p>
<p>WD Managing Contacts</p>	<p>Enables you to manage contacts and send them emails. 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. 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>
<p>WD Multimedia</p>	<p>Enables you to read and play:</p> <ul style="list-style-type: none"> - some videos (AVI, MPEG, ...), - some sound files (MP3, WAV), - some audio CDs, - some Flash animations.
<p>WD Outlook</p>	<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>
<p>WD Password</p>	<p>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, and so on) .</p>

WD POP3Proxy	This example presents a POP3 proxy. A proxy is a software that connects to a server on behalf of another program. In this example, the proxy is used to automatically archive the emails in a Hyper File database.
WD Puzzle	Explains how to use the Drag&Drop functions as well as the functions for handling the images in WLanguage.
WD RSS Reader	Presents the use of classes for reading the RSS streams
WD RTF editor	Enables you to edit text with RTF formatting.
WD Scanner - Twain	Used to: <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	Enables you to create a graphic schedule containing some click areas.
WD Search on Internet	Enables you 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 Shipping costs	Calculates the shipping costs for a package.
WD Synchronizing files	Enables you to synchronize two file directories. 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.
WD Telephony	Uses the telephony functions of WinDev to: <ul style="list-style-type: none"> - Dial a phone number - Detect and identify the incoming calls
WD Transferring files by FTP	Enables you to transfer some files: <ul style="list-style-type: none"> - from an FTP server to the local computer, - from the local computer to an FTP server.
WD Video conference	Enables you to establish a conference with a remote user. Enables you to send a file to a recipient and to establish a dialog with him or her.

WD Video surveillance	Enables you to broadcast images recorded by a Web camera via a network (Internet, Intranet, etc.).
WD ViewTIF	Small image editor developed in WinDev for images in TIF format only. WD ViewTif is used to open an image and to display the different pages of the image - in MDI mode (in several child windows), - in slide show mode.
WD WebCam	Explains how a Web camera can be included in a WinDev window. You can: - View the image broadcasted by the camera - Perform an immediate capture - Capture the scene broadcasted as a Windows animation ("AVI" file)
WD Zip	Enables you to create and read compressed archives in "WDZ" format (WinDev ZIP).
WDOrganizer	Enables you to synchronize the appointments in your Outlook, Lotus Notes and Google calendars.

Educational examples

WD Advanced input mask	This example presents the advanced input masks: - 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	Presents the different types of reports. Used to illustrate: - the internal reports, - the chart control, - printing in a RTF, HTML or PDF file.
WD Advanced Rotation	Used to perform the rotation of an image from any point that will be used as rotation center.
WD Age Pyramid	Explains how to create an "Age pyramid" chart from two histograms.
WD Agent	Includes an "agent" in a WinDev application via an ActiveX.

WD Alarm	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.
WD Animated	Presents the main animations that can be performed in an Image control. This example proposes: - some animations without code, - some programmed animations.
WD BarCode	Enables you to create some products and to associate them with an existing bar code via a bar code reader. These products can be printed as labels.
WD BGInfo	This example explains how to write some information on the background of a computer screen (IP address, name of the computer, today's date, etc.).
WD Bluetooth	This example presents the use of the Bluetooth functions: - to list the accessible Bluetooth devices - to list the services proposed by a device - to connect to a Bluetooth device - to send a file to a device via the "OBEX" functions of WLAN-language.
WD Calculations on date and times	Performs some calculations on the dates and times via procedures: - Calculate the duration between two moments (dates and times) - Calculate a sum of durations - Calculate an average of durations
WD Carousel	This example presents the use of the carousel control of WinDev. It can be handled like a ListView control.
WD ChainedList	Supplies the tools for building chained lists with simple chaining. A list with simple chaining is used to store and browse in organized way a set of ordinary values.
WD Characters	This program, powered by WinDev, is used to insert some special characters into an RTF control and to view the ASCII and ANSI codes
WD Chart	This educational example presents the use of the chart control. The following charts are presented: Semi-circular, Donut, Pie, Line, Scatter, Histogram, Radar, Area.

WD Click area	Manages the click areas on non-rectangular shapes. When the map is clicked, the selected region is displayed at the bottom of the screen.
WD Clickable map	Presents a cartography application based on a MICHELIN map (authorization #9905251). 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.
WD Clickable reports	The example is used to list the customers and to print this list of customers. During the print preview, a click performed on the customer name will open the form of the selected customer.
WD Closing Windows	Used to: - Close a Windows session - Stop and restart the computer
WD Coloring Search	This example explains how to highlight the sought words in the results of a full-text search.
WD Controlling Excel	Enables you to control Excel via OLE Automation. All the Excel functions can be used.
WD Controlling IE	Enables you to control IE via OLE Automation. All the IE functions can be used.
WD Controlling Open Office Database	This example proposes a class used to easily handle an Open Office database without having to handle any OLE objects.
WD Controlling OpenOffice Calc	This 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.
WD Controlling OpenOffice Writer	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.
WD Controlling Outlook	Enables you to control Outlook via OLE Automation. All the Outlook functions can be used.
WD Controlling Spreadsheets	This example explains how to control the OpenOffice Calc spreadsheet and the Excel spreadsheet with a generic code.
WD Controlling Word	Enables you to control Word via OLE Automation. All the Word functions can be used.

WD Controlling word processor	The purpose of this example is to propose a universal interface used to control these applications without having to worry about the application to control.
WD CSGroupware	Presents the user groupware with a HyperFileSQL Client/Server database.
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 to/from the automatic opening system for the removable devices by programming.
WD Direct printout	Enables you 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 Draw	Highlights some sections of a scanned document or annotates the images via the drawing functions of WLanguage.
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 the use of an internal component allowing you to draw some elements on a chart control: - Min/Max areas - Various areas - Flags - Plots and trackers,...

WD DuplexPrintout	This example presents a solution for double-sided printing that can be disabled, which means that the back sides may not be printed.
WD Duplication	This educational example explains how a database can be duplicated on a server.
WD Dynamic compilation	Presents: <ul style="list-style-type: none"> - 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 External file	Explains how to handle 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 Flowchart	This example is used to create some flowcharts.
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 some 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 some 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.
WD Handling shapes	Used to: <ul style="list-style-type: none"> - Draw some 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
WD Handling the Organizer control	Example for using the Organizer control. The following features are used: Creating an appointment, Saving an organizer, Restoring an organizer
WD Highlight	Enables you to highlight the control currently edited by changing its style. 3 types of highlighting methods are available: <ul style="list-style-type: none"> - Displaying a border around the control, - Changing the background color, - Changing the font.
WD Hotkey	Enables you to define your own keyboard shortcuts.
WD HSL	Enables you to define a color via its HSL components (Hue, Saturation, Lightness).
WD HTML Export	Manages the external files. Also explains how to handle the files in HTML format.
WD Importing HTML pages	Saves an HTML page found on a Web site. This page is analyzed in order to import all its dependencies (images, applets, etc.).
WD Input checks	Checks the data entered by the user (name, company, city, zip code, email address etc.) in a form window.
WD JAVA Calls	This example presents the call to the methods of the classes of a JAR archive via JavaExecuteFunction . This example also explains how to fill a memory table (same operating mode in Java and in Windows)
WD Java Chart Designer	This example explains how to create some charts via the drawing functions. Three types of charts are presented: <ul style="list-style-type: none"> - histogram - scatter - line.

WD Java Client	<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. 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 allows you to generate a .jar.</p>
WD Java Threads	<p>This Java educational example explains how to use threads. Critical section, semaphore, all the main points for multi-thread development are presented here.</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. This applet is used to read and send emails.</p>
WD Jump Lists	<p>JumpList in Windows 7 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 to identify the use of a keyboard key.</p>
WD Lasso	<p>Uses a selection lasso in an image control to perform a zoom in a preview control.</p>

WD License	Enables you 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 instance. The text must be entirely read, the choice for validating the window is activated only when the display control has entirely scrolled.
WD Line smoothing	Enables you to smooth a "line" chart by using a "cubic splines" algorithm.
WD Lock	Explains how to handle the functions used to automatically lock an application.
WD Mailshot	Enables you to send a "mailshot" by Internet.
WD Managing the Java errors	This educational example is used to understand the operating mode and the implementation of the mechanism for managing the WinDev errors.
WD MIME Extraction	Enables you to extract attachments from an email.
WD Multi-windowing	Enables you to understand the principle of multi-windowing in WinDev.
WD Network frames	The analysis of the frames that circulate on the network allows the administrators to audit the network and to detect inconsistencies. To listen to the network, the application is based on the Winsock 2 API and on the RAW Socket.
WD Number in words	Enables you to convert a numeric value into letters. The result of this conversion is displayed in a "Check Book" report.
WD OLE DB	Enables you to access the external databases via OLE DB.
WD OPC	Explains how the OPC protocol is implemented in WinDev.
WD OpenGL	Enables you to interface OpenGL and WinDev. OpenGL is a software interface used to draw three-dimensional objects (3D).
WD OSD	OSD means: On-Screen Display. It is the name given to the interfaces that appear on the computer or television screens for example. They are often used to perform settings. These types of menus are displayed above all the other elements found on the screen. This example explains how to create this type of interface in WinDev.
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 <code>InitParameter</code> , <code>LoadParameter</code> , <code>SaveParameter</code> . These functions are used to configure the backup of controls, variables and parameters.
WD Phonetics	Enables you 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 Pool of threads	Implements a "pool" of threads. 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 PowerPoint Generation	This example explains how to generate a PowerPoint presentation from: - the data found in a WinDev application such as the data found in tables, charts .. - and from a base PowerPoint document (.pptx).
WD Printing TreeViews	Enables you to print a Treeview. This example uses a "TreePrnt" class that can be easily reused in any Treeview control
WD Printing TreeViews	Explains how to print a TreeView control.
WD Printout Parameters	Illustrates the use of <i>Parameter</i> .
WD Process	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).
WD Progress bars	Enables you to create different types of progress bars.
WD Queries	Enables you to perform some calculations, to pass parameters or to print the result of a SQL query on a HyperFileSQL data file.
WD Reflexive link	Uses a "reflexive" link to manage the family trees: each person has several parents and several children. All these records are stored in the same file.
WD Regular expressions	Presents two methods for using the regular expressions: - checking the input format - checking out different elements while respecting the input format.
WD RemoteControl	Used to implement the remote control of a WinDev application.

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 RMI Server	<p>Presents the use of a WinDev RMI server to access data in HyperFileSQL format from a MAC for instance.</p>
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>Enables you to access your SAP data from your WinDev applications via the SAP functions.</p>
WD Screen saver	<p>Used to manage a screen saver via the management of timers and Windows events.</p>
WD Search by composite key	<p>Presents the use of filters and the use of composite keys in order to optimize the search times on a HyperFileSQL data file.</p>
WD Search Control	<p>This example explains how to manage an edit control used to perform some FullText "Google" searches in your database.</p>
WD Sending keys	<p>Illustrates how some keyboard sequences can be sent to an application with the WLanguage functions.</p>
WD Service	<p>This example presents the creation of a "service" application with WinDev.</p>
WD SharingMemory	<p>This example explains how to use the functions for sharing memory.</p>
WD SNMP	<p>This example displays some information about the specified network computer: IP address of the computer, its MAC address, its system ...</p> <p>This example interrogates the computer to get this information via the SNMP functions of WLanguage.</p>
WD SOAP Server	<p>Presents the use of the SOAP functions available in WLanguage.</p> <p>Presents two main topics:</p> <ul style="list-style-type: none"> - 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	Used 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. The following stock charts are managed by WinDev without any programming: <ul style="list-style-type: none"> - BarCharts - Candlestick - Line - Histogram - Minimum/Maximum
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 Table Export	Exports the content of a table to: <ul style="list-style-type: none"> - Word - Excel - The Windows clipboard - A text file
WD Tables and Combos	Enables you to select a product in a combo box and to display its details 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 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	Enables you 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	Presents the use of the TreeView controls.
WD TreeView Duplicates	This example is used to manage the duplicates in a treeview via 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 Trigger	Enables you 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 Triggers are used to control 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 .NET	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 .Net classes	Enables you to use the .NET classes in WLanguage. Used to: - analyze a character string, - configure the dialog boxes displayed.
WD Using aliases	Enables you to handle the same physical data file via several aliases.
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 particular action.
WD Who's locking	Signals to the users of a HyperFileSQL database on a network who is locking a record.
WD Windows Drag and Drop	Proposes a catalog of images. 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 advanced types (xlsDocument, xlsColumn, xlsRow and xlsCell).
WD XML	Enables you to import an XML file in HyperFileSQL format and to export files in XML format (eXtensible Markup Language).

Some additional examples will be supplied with the Technical Support Newsletter (LST) or will be available for download from our site (www.windev.com).

