

DEVELOP 10 TIMES FASTER



25

## SUMMARY

### Introduction

Preliminary points .....	9
Tutorial overview .....	9
Legend of the symbols in this tutorial .....	10
How to access the online help .....	11
If you are familiar with WINDEV Mobile 24 .....	13
What is WINDEV Mobile used for? .....	13

## PART 1 - OVERVIEW OF WINDEV MOBILE

### Lesson 1.1. Discover WINDEV Mobile

Overview .....	18
Starting WINDEV Mobile .....	18
Development environment .....	19
The editor .....	19
The menu bar (ribbon) in details .....	20
The environment colors .....	21

## PART 2 - ANDROID APPLICATION

### Lesson 2.1. My first Android project

Overview .....	26
Necessary configuration for Android .....	26
My first project .....	26
Creating the project .....	26
My first window .....	30
Overview .....	30
Creating the window .....	30
Displaying a message .....	31
First test .....	33
First deployment on the device .....	34
Principle .....	34
Implementation .....	35

### Lesson 2.2. Interface (UI)

Overview .....	40
Choosing the resolution according to the device .....	40
Window orientation .....	40
Practical example .....	41

Don't forget to visit our site ([www.windev.com](http://www.windev.com)) on a regular basis to find out whether upgraded versions are available.

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

This documentation is not contractually binding. PC SOFT reserves the right to modify or delete any topic dealt with in this document.

All product names or other trademarks mentioned in this publication are registered trademarks of their respective owners.  
© PC SOFT 2020: This publication may not be reproduced in part or in full without the express consent of PC SOFT.

Touchscreen management.....	43
Practical example.....	44
The different types of controls.....	45

### Lesson 2.3. Distributing the application

Overview .....	47
Deployment via Google Play .....	47
Deployment via a Web server .....	47
Deployment from the PC via ADB (advanced mode).....	48
Copying the application onto Mobile device.....	48

## PART 3 - IOS APPLICATION

### Lesson 3.1. My first iOS project

Overview .....	52
Required configuration for iOS .....	52
My first project .....	53
Creating the project.....	53
My first window .....	56
Overview .....	56
Creating the window.....	56
Displaying a message .....	57
First test.....	59
First deployment on the device .....	62
Principle .....	62
Implementation.....	62
Transferring the WINDEV Mobile project onto Mac .....	64
Compiling the project in Xcode .....	64

### Lesson 3.2. Interface (UI)

Overview .....	66
Choosing the resolution according to the device.....	66
Window orientation.....	66
Practical example.....	67
Touchscreen management.....	70
Practical example.....	70
The different types of controls.....	71

### Lesson 3.3. Distributing the application

Overview .....	73
Deployment.....	73

## PART 4 - PROGRAMMING CONCEPTS

### Lesson 4.1. Concepts and terminology

Overview .....	78
Main concepts.....	78
Terminology .....	79

### Lesson 4.2. The WLanguage basics

The variables .....	81
What is a variable.....	81
Declaring and assigning a variable .....	81
The types of variables .....	82
The scope of variables .....	83
Simple operations on the variables.....	84
Tips .....	85
The conditional IF and SWITCH statements.....	86
The IF statement .....	86
The SWITCH statement .....	87
The loops .....	88
The FOR statement .....	88
The LOOP statement .....	88
The WHILE statement .....	89
The procedures .....	90
Types of procedures .....	90
Creating and calling a procedure.....	90
Procedure parameters .....	91
Procedure result.....	93

## PART 5 - ANDROID AND IOS: DEVELOPING AN APPLICATION

### Lesson 5.1. Overview

Overview .....	98
The sample project used.....	99
Opening project .....	101
Choosing the platform.....	101
Enabling the iOS platform.....	101

### Lesson 5.2. Developing the application

Project analysis.....	105
Displaying the list of products .....	105
Creating the window.....	106
Creating the form window.....	110
Creating the window.....	110
Creating controls .....	111
Displaying the form from the list of products .....	113

<b>Managing the product creation and modification .....</b>	<b>114</b>
Modifying the product form.....	114
Creating a new product .....	118
<b>Using the camera and displaying an image .....</b>	<b>121</b>
Overview .....	121
Creating the button for taking photos .....	121
Taking photos .....	124
Selecting a photo in the photo album .....	124
<b>Managing the bar code of product .....</b>	<b>125</b>
Overview .....	125
Implementation.....	125
<b>Managing anchors in the Product form .....</b>	<b>127</b>
<b>Using the Map control .....</b>	<b>128</b>
Creating the window.....	128
Creating the Map control .....	128
Displaying stores on the map .....	129
<b>Using a Multiline Zone control.....</b>	<b>130</b>
Creating the window.....	130
Creating the Multiline Zone control .....	131
Modifying the Multiline Zone control.....	131
Programming the menu .....	134
<b>Application test.....</b>	<b>134</b>
 <b>Lesson 5.3. Managing the orientation of windows with the layouts</b>	
<b>Overview .....</b>	<b>136</b>
<b>Using anchors .....</b>	<b>136</b>
<b>Running the application test .....</b>	<b>142</b>
 <b>Lesson 5.4. Window with search</b>	
<b>Overview .....</b>	<b>144</b>
<b>Creating the query used to find the products .....</b>	<b>145</b>
Creating the query .....	145
Query test .....	146
Adding a selection condition.....	147
Test of query with parameters .....	148
<b>Creating the interface .....</b>	<b>149</b>
Creating the window.....	149
Managing the search .....	151
<b>Improving the window .....</b>	<b>153</b>
Managing "Pull to refresh" .....	153
Using a sliding menu .....	155

<b>Lesson 5.5. Window with scroll</b>	
<b>Overview .....</b>	<b>160</b>
<b>Creating the Form window with swipe .....</b>	<b>160</b>
Creating the Form window .....	160
Creating the internal window .....	161
Managing the swipe in the form window.....	161
<b>Window test .....</b>	<b>164</b>
<b>Conclusion.....</b>	<b>165</b>
 <b>Lesson 5.6. Deploying the application</b>	
<b>Overview .....</b>	<b>167</b>
<b>Generating the Android application.....</b>	<b>167</b>
<b>Generating the iOS application .....</b>	<b>171</b>
Transferring the WINDEV Mobile project onto Mac .....	173
Compiling the project in Xcode .....	173

## PART 6 - WINDOWS MOBILE APPLICATION

<b>Lesson 6.1. My first Windows Mobile project</b>	
<b>Overview .....</b>	<b>178</b>
<b>My first project .....</b>	<b>178</b>
Creating the project.....	178
<b>My first window .....</b>	<b>181</b>
Overview .....	181
Creating the window.....	181
Displaying a message .....	182
<b>First test.....</b>	<b>184</b>
<b>First deployment on the device .....</b>	<b>185</b>
Principle .....	185
Implementation.....	185
 <b>Lesson 6.2. Databases</b>	
<b>Format of databases .....</b>	<b>188</b>
HFSQL database .....	188
CEDB .....	189
AS/400 .....	189
<b>Sharing data between two applications .....</b>	<b>190</b>
Handling the same data files .....	191
Copying the data files onto the mobile device (Pocket PC for example) .....	191



## Lesson 6.3. Advanced programming

<b>Overview .....</b>	<b>193</b>
Opening project .....	193
Project description .....	193
<b>Displaying the list of products .....</b>	<b>194</b>
Creating the window .....	194
Creating the Table control .....	194
<b>Creating the form window .....</b>	<b>197</b>
Creating the window .....	197
Creating controls .....	197
Displaying the form from the list of products .....	199
<b>Managing the product creation and modification .....</b>	<b>200</b>
Modifying the product form .....	200
Creating a new product .....	201
<b>Creating a menu window .....</b>	<b>204</b>
Creating the window .....	204
Customizing the menu .....	205
Programming the menu .....	206
<b>Running the application test .....</b>	<b>206</b>
<b>Creating the executable .....</b>	<b>207</b>

## Lesson 6.4. Distributing the application

<b>Overview .....</b>	<b>210</b>
<b>Setup in MSI format .....</b>	<b>212</b>
Initial setup .....	212
Update in MSI format .....	212
<b>Setup in CAB format .....</b>	<b>212</b>
Initial setup .....	212
Update in CAB format .....	213

## Conclusion

# INTRODUCTION

## Preliminary points

**Caution:** This is a tutorial. We advise you to check the online help when you are using WINDEV Mobile.

The purpose of this tutorial is to help you discover WINDEV Mobile, get familiar with the editors and learn the WINDEV Mobile concepts.

This tutorial does not cover all the features in WINDEV Mobile.

**This tutorial is intended for developers who are already familiar with WINDEV.** It only presents the main concepts required to develop an application for mobile devices (Android, iPhone, iPad, Windows Mobile, etc).

If you are not familiar with WINDEV, we recommend that you read the WINDEV tutorial beforehand.

Remark: The WINDEV tutorial is available in WINDEV, WEBDEV and WINDEV Mobile online help.

Spend at least a few hours to follow this tutorial and learn WINDEV Mobile: it is a good investment!

As WINDEV Mobile is constantly evolving, the screenshots in the tutorial may differ from the windows displayed in your product.

## Tutorial overview

The tutorial has been designed to help you develop on the main mobile platforms:

- Android.
- iOS (iPhone/iPad).
- Windows Mobile/CE.

This tutorial is divided into the following sections:

- Part 1: Overview of WINDEV Mobile.
- Part 2: Creating and deploying a first Android application.
- Part 3: Creating and deploying a first iOS application.
- Part 4: Quick reminder of main concepts of WINDEV Mobile and WLanguage.
- Part 5: Creating an application for data management for Android and iOS. This part is common to Android and iOS and it allows you to create management windows for these two platforms.
- Part 6: Creating applications for Windows Mobile.

This tutorial has been designed to progressively teach you how to use WINDEV Mobile. By following this tutorial:

- you will discover the main concepts explained informally; these are the concepts you must learn and understand.
- you will also be asked to perform operations that illustrate the concepts just explained.

As you progress through the tutorial, if you want to take a closer look at a concept or if you want to get more details about a programming function, see the online help (accessible from the editors). The size of a lesson is not necessarily proportional to its relevance.

**And don't forget to take a look at the examples supplied with WINDEV Mobile: they are very instructive!**



Tip

The tutorial may have evolved since this document was printed. Feel free to consult the online version (<http://doc.windev.com>).

## Legend of the symbols in this tutorial



This symbol indicates the duration of the lesson. Please note that the actual time may vary according to your level of experience.



An example is available to complement the lesson. The examples are available in the WINDEV Mobile home page (Ctrl + <).



This symbol introduces a "Tip": reading the associated text is strongly recommended.



This symbol introduces a "Warning": reading the associated text is essential.




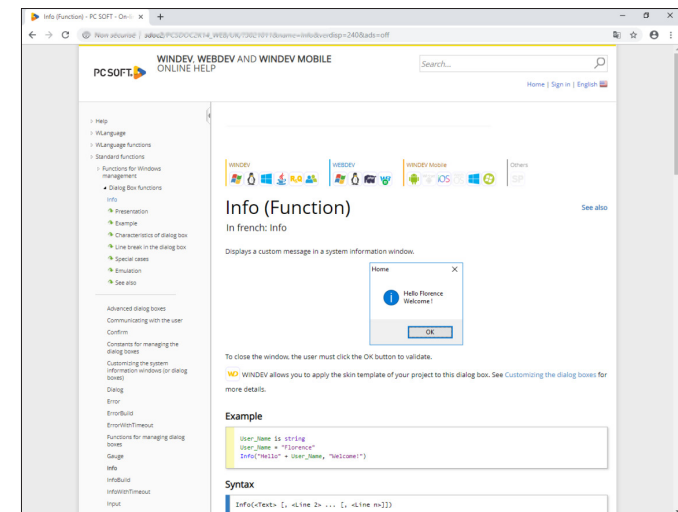
This symbol introduces a "Note": reading the associated text is recommended.

## How to access the online help

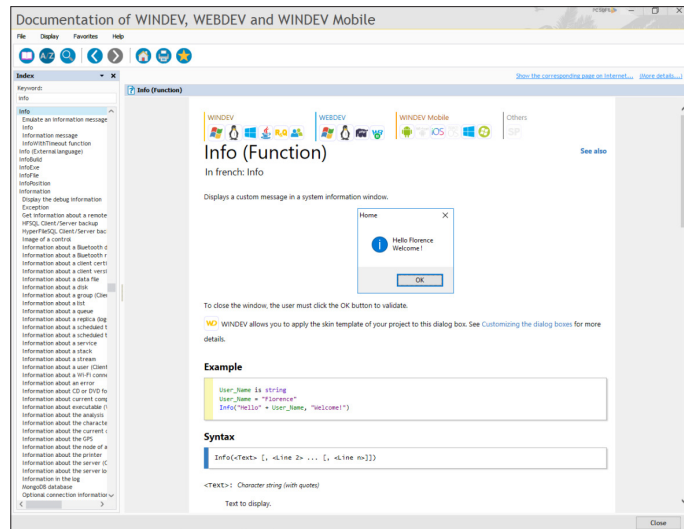
The online help of WINDEV Mobile allows you to get detailed information about the 3700 WLanguage functions. The online help also contains the help about the editors and the controls, tips, ...

The online help is available at any time in WINDEV Mobile:

- In the code editor, a specific help is available for each function via the F1 key.
  - Each dialog box displayed by WINDEV Mobile proposes a  button allowing you to access the corresponding help page.
  - The help menu of the editors ("Help" option available on the "Home" pane, in the "Online help" group of the WINDEV Mobile menu) allows you to start the online help.
- The help can be displayed:
- in an Internet browser, if you have access to Internet:



- in a specific "help browser":



## Remark

We advise you to check the online help on Internet rather than the local online help. Indeed, the Internet online help is updated on a regular basis.

The online help of WINDEV, WEBDEV and WINDEV Mobile on Internet is available from any computer equipped with an Internet access, without the product being necessarily installed.

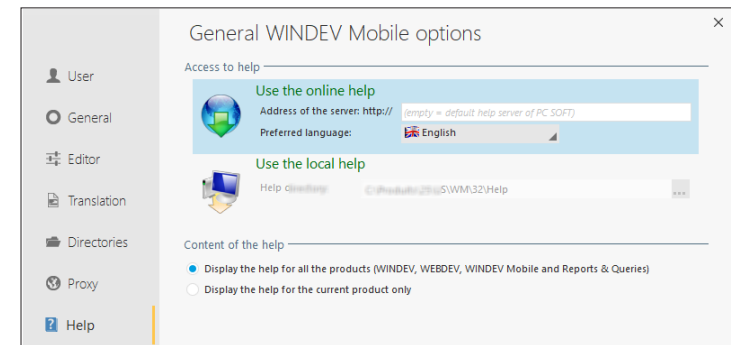
Each Web user can add comments about the documentation pages: personal notes, examples, links, ...

**Note:** If you have no access to Internet, you have the ability to start the local help from the product:

1. On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WINDEV Mobile".

2. In the "Help" tab, select:

- the access mode to the help database.



- the help content: help common to WINDEV, WEBDEV and WINDEV Mobile or help for the product currently used.

## If you are familiar with WINDEV Mobile 24

If you are familiar with WINDEV Mobile 24, following this tutorial will do no harm: it's a good opportunity to "review" the features of WINDEV Mobile!

## What is WINDEV Mobile used for?

WINDEV Mobile is an IDE (Integrated Development Environment). It allows you to develop applications in many fields:

- Management of stocks,
- Inventories, tracking of goods,
- Adjustment and monitoring of machines on an assembly line,
- Taking orders for fast processing in a temporary outlet (fairs, schools, booth, ...),
- Customer forms,
- Help with making snap decisions on a cell phone,
- Checking the identity of visitors at an event: trade fair, presentation of products, ...
- On-call doctors or vets,
- Taking information in a temporary outlet: trade fair, street poll, stadium, ...
- Returning leased heavy equipment (tools, vehicles, ...) to a parking lot,
- ...

WINDEV Mobile is a development environment that includes all the tools required to develop an application.

Unlike other programming languages, there is no need to find and add modules to be able to design, check and install an application.

The 5GL (5th Generation Language) of WINDEV Mobile, named WLanguage, will surprise you by its simplicity: a few hours are all you need to get the hang of it, a week is usually all it takes to fully master its potential!

No more programming hassle, WLanguage is available in English and in French!

DEVELOP 10 TIMES FASTER

## PART 1

### Overview of WINDEV Mobile

## LESSON 1.1. DISCOVER WINDEV MOBILE

**This lesson will teach you the following concepts**

---

- Starting WINDEV Mobile.



Estimated time: 10 mn

## Overview

WINDEV Mobile is an IDE (Integrated Development Environment) allowing you to develop applications for Android, iOS, Universal Windows (Windows 10), Windows Mobile, ... in several fields: business, industrial, medical, hospitality, ... The developed applications can give access to information stored in the databases.

This tutorial will explain how to create your applications (with or without database) and how to improve them by using the different features proposed by WINDEV Mobile.

## Starting WINDEV Mobile

- ▶ Start WINDEV Mobile 25 (if not already done).
- ▶ A welcome wizard starts if **WINDEV Mobile 25 was never started before**. This wizard is used to:
  - If you worked with an earlier WINDEV Mobile version, this wizard allows you to retrieve the existing configurations.
  - If you are a new user, this wizard allows you to configure your environment. This allows you to choose the screen configuration used and to configure the Control Centers.
- ▶ If you have already launched WINDEV Mobile 25, log in if necessary. The development environment starts. The home page is displayed. This home page is used to:
  - create a project,
  - open an existing project,
  - open an example,
  - open one of the projects found in the tutorial.
- ▶ Let's take a look at the development environment of WINDEV Mobile. To do so, from the home page:
  - Click "Tutorial".
  - Double-click on "iOS/Android application (Answer)".
  - The corresponding project is opened in the editor. The project dashboard is displayed. The project dashboard allows you to check the progress of a project via several elements (called widgets).

## Development environment

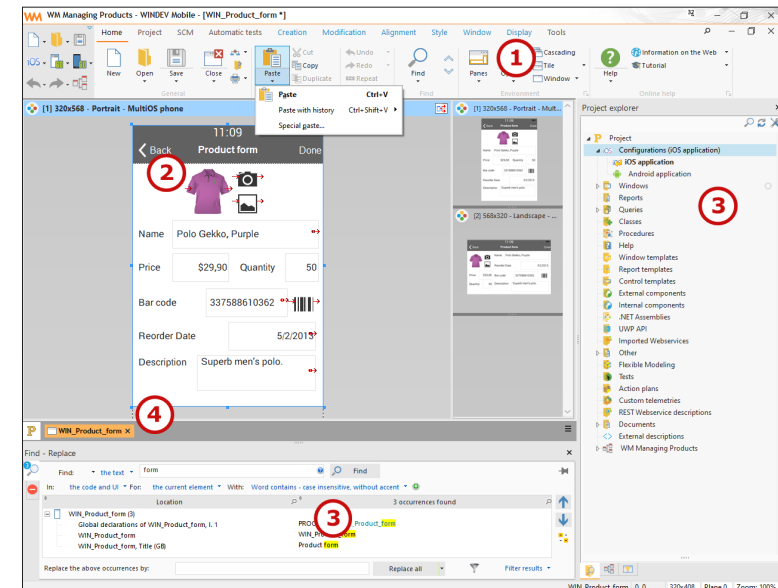
### The editor

The development environment of WINDEV Mobile includes a specific interface and several editors allowing you to create the different elements of your applications.

For example, the window editor is used to create windows, the report editor is used to create reports, ...

- ▶ To discover WINDEV Mobile, we are going to open the "Product form" window:
  1. Press Ctrl + E.
  2. In the window that is displayed, type the name of the window to open: WIN\_Product\_form.
  3. Validate. The window is displayed in the editor.

All editors are using the same environment:



1. **Menu of editors**, displayed in ribbon format (we'll see how to use it in the next paragraph).
2. **Current editor** (window editor in this case). This space allows you to see the element currently created or modified in WYSIWYG (What You See Is What You Get).
3. **Panes**. The interface of WINDEV Mobile includes several panes allowing you to quickly access different types of information. Some examples:
  - the "Project explorer" pane (displayed on the right) is used to list all project elements by category.



- the search pane (displayed at the bottom) is used to perform searches in the entire project and in its elements.

These panes can be hidden by pressing Ctrl + W if necessary.

**4. Bar of opened documents.** This bar is used to quickly see all opened elements. A simple click on the button corresponding to the element displays it in its own editor.

### The menu bar (ribbon) in details

The menu bar of WINDEV Mobile is presented in ribbon format. This ribbon includes panes in which the different options of editors are grouped.

We are going to take a closer look at the main ribbon elements, as well as how we will be using it in this tutorial.



### The different ribbon elements

The ribbon includes 3 areas:

- the button area, on the left (1).
- the pane area, at the top (2).
- the option area (3).

Let's take a closer look at these areas.

#### The button area (1)

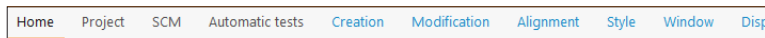


The button area groups the **quick access buttons**. These buttons are used to perform the most usual operations, common to all editors: save, open, create, ...

The product logo is used to display the "About" window, the custom menus and the drop-down menus found in the former interface of editors.

The arrow at the top right of the button area allows you to find the toolbars and drop-down menus of the old editor interface.

#### The pane area (2)

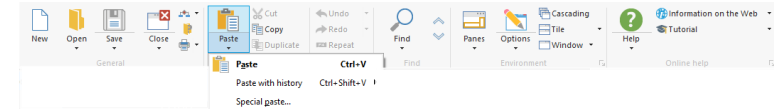


The different ribbon panes are used to access the options of different editors for the current project. Several types of panes are available:

- the current pane: The name of the pane appears in bold and is underlined by an orange line.

- the popup panes, specific to the current element: The pane name is displayed in blue.
- the available panes: The pane name is displayed in black.

### The option area (3)



The options displayed in the ribbon differ according to the selected pane. Several types of options are available:

- Options to check,
- Buttons to click,
- Button with arrow used to expand the options. Two types of buttons with arrow are available:
  - the buttons with arrow used to expand a menu.
  - the buttons with arrow used to expand a menu (click on the arrow) or to perform a default action (click on the button icon).

The options are organized by group. Each group of options has a name and it can also include a group button. This button is used to perform a specific action according to the current group: display the description of current element, display the help, ...

In this tutorial, to identify a menu option, we will be talking about panes and groups. For example: To display the help, on the "Home" pane, in the "Online help" group, click "Help".

### The environment colors

The environment is using a light theme by default.

Several other themes are also available:

- Light theme, grey ribbon. In this mode, the menu bar is not colored anymore: it is grayed.
- Gray theme. In this mode, the environment and the interface windows are displayed on a light gray background.
- Dark theme. In this mode, the environment and the interface windows are displayed on a black or dark gray background.

► To modify the theme used by the environment:

- On the "Home" pane, in the "Environment" group, expand "Options" and select "General options of WINDEV Mobile".
- In the "Editor" tab, in the "Themes" area, select the theme to use.
- Validate. The theme will be taken into account during the next start of WINDEV Mobile.

Remark: To improve the readability of this manual, the light theme will be used for the different images that illustrate the operations to perform.

## PART 2

Android  
application



## LESSON 2.1. MY FIRST ANDROID PROJECT

**This lesson will teach you the following concepts**

---

- Required configuration.
- Creating an Android project.
- My first window.
- My first test.
- First deployment.



Estimated time: 1 h

## Overview

To start developing with WINDEV Mobile for an Android platform, we are going to create a first project. This project will contain a window used to display a message.

This first example will present the main concepts of development for Android with WINDEV Mobile. Before creating our first project for Android, the development computer must be configured.

## Necessary configuration for Android

To develop an application for the Android platform, the following elements must be installed on the development computer:

- The JDK: The JDK (Java Development Kit) distributed by Oracle is used to compile the generated Java files.
- The Android SDK of Google: The Android SDK (Software Development Kit) is a set of files and applications distributed by Google in order to allow the compilation of applications for the Android operating system.

**Caution:** The Android SDK includes sections corresponding to the versions of device platforms (5, 6, 7, ...).

The download and the setup of Android SDK are proposed if necessary when generating the Android application from WINDEV Mobile.

- Gradle tool: This tool is required to compile and generate Android applications. Gradle can be downloaded and installed if necessary when generating the Android application from WINDEV Mobile.

See the online help for more details (download addresses, ...).

We advise you to restart the computer once JDK and SDK have been installed.

## My first project

### Creating the project

We are now going to create our first project for Android.



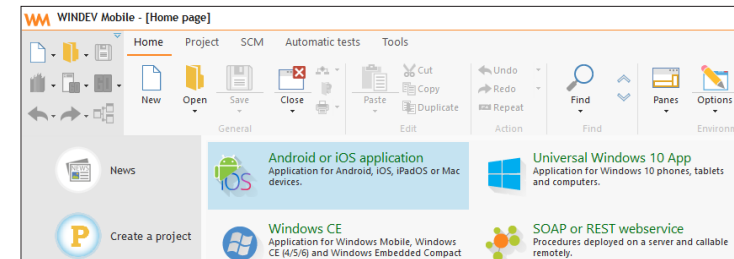
Answer

A corrected project is available. To open this project:

1. Display the WINDEV Mobile home page (Ctrl + <).
2. Click "Tutorial" and select "My Android project (Answer)".

► To create a project:

1. Start WINDEV Mobile 25 (if necessary).
2. Display the WINDEV Mobile home page if necessary (Ctrl + <).
3. On the home page, click "Create a project" then "Android or iOS application"..




4. The project creation wizard starts. The different steps of the wizard help you create your project. The information specified in this wizard can be modified later.

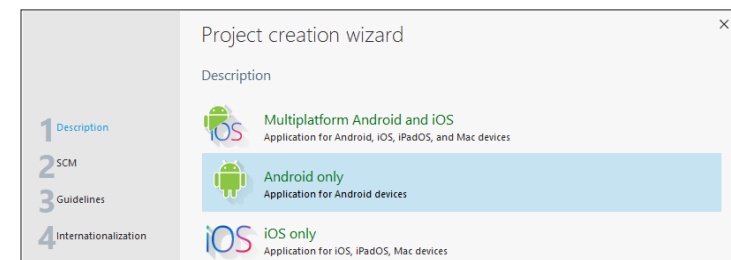


Remark

Tip: Other method for creating a project:

1. Click  among the quick access buttons of WINDEV Mobile menu.
2. The window for creating a new element is displayed: click "Project".

5. The first step of the wizard allows you to define the project execution platform:



In our case, select "Android only" and go to the next step of the wizard.

6. The wizard proposes to create a blank project or a project based on an example. Choose "Create a blank project" and go to the next step.
7. The wizard proposes to choose the type of Android devices affected by the project:
  - Generate an application for phones and tablets.
  - Generate an application for phones.
  - Generate an application for tablets.
  - Generate an application for a specific device.



Remark

If the application is intended to operate on several Android devices (phones with different sizes or resolutions for example), we advise you to use one of the following options:

- "Generate an application for phones and tablets",
- "Generate an application for phones".

In this case, WINDEV Mobile proposes the smallest resolution to create the application windows. Using anchors (see "Interface (UI)", page 39 and "Developing the application", page 104) will allow the application to operate on all devices.

8. In this example, we are going to generate an application for phones. Select "Generate an application for phones" and go to the next step.



Remark

You own an Android device and you want to run the application test on this device? Select "Generate an application for a specific device". The wizard next step is used to select the requested device.

9. The wizard proposes to type the name of project, its location and its description. In our case, this project will be named "My\_Android\_Project". WINDEV Mobile proposes to create this project in the "My Mobile projects\My\_Android\_Project" directory. You can keep this location or modify it via the [...] button.

10. Go to the next step.

11. The wizard proposes to add documents. Go to the next step.

12. This step allows you to use the Source Code Manager (SCM). We will not use it in this example. Select "No, do not use SCM". Go to the next step.

13. This step is used to define the code style. Don't modify the suggested options. Go to the next step.

14. This step is used to define the style book of application. We will keep "Material Design Blue 2".

15. The other steps of the wizard are not important for our first project. Click "Finish" directly. The project is automatically created.

16. The window for creating a new element is displayed. This window is used to create all elements that can be associated with a project.

## My first window

### Overview

The first window allows the user to display a welcome message via the "Display" button.

You may think this is too basic but we advise you to create this window. You may be surprised by how intuitive and how easy it is to use the editor of WINDEV Mobile. Furthermore, this window will allow you to discover concepts that are fundamental for the rest of this tutorial and to see the entire process for developing an Android application with WINDEV Mobile.

### Creating the window

► To create the window:

1. In the window for creating a new element, click "Window" then "Window".

Remark

As a new project was created, the window for creating a new element is automatically displayed.

To display the window for creating a new element, all you have to do is click  among the quick access buttons of WINDEV Mobile:



2. The window creation wizard starts.
3. In the list of proposed windows, select "Blank". The skin template used is displayed at the bottom right of wizard. "Material Design Blue 2", the skin template that was selected when creating the project is selected by default.

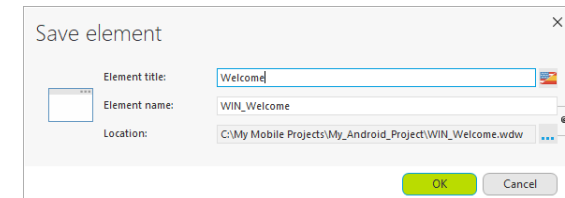
Remark

Skin templates allow you to quickly create outstanding interfaces. A skin template defines the window style as well as the style of all controls that will be used in this window. Thus, there is no risk of obtaining an ugly interface.

4. Validate. The window is automatically created in the editor. The window for saving an element is displayed. This window is used to specify:

- the element title. For a window, this title will be displayed in the Action Bar of window.
- the element name that corresponds to the window name. This name will be used in programming.
- the element location. This location corresponds to the directory in which the physical file corresponding to the element is saved. The window is a "WDW" file, saved in the project directory.

5. Specify the title of the element: "Welcome". The name of the element is automatically proposed: "WIN\_Welcome".



### Remark

Let's take a look at the window name proposed by WINDEV Mobile: this name starts with the letters "WIN\_". This prefix is automatically added because the project is using a code style.

The code style is used to define a prefix for each type of object, allowing you to quickly identify the element:

- a window starts with "WIN\_",
- a button starts with "BTN\_",
- etc.


If you do not want to use this code style, you can simply disable it: on the "Project" pane, in the "Other actions" group, expand "Code style" and uncheck "Use code style".

6. Save the window by clicking "OK".

### Displaying a message

You are now going to create a Button control used to display a message.

► To create the "Display" Button control:

1. On the "Creation" pane, in the "Usual controls" group, click . The control appears in creation mode under the mouse pointer.
2. Move the mouse toward the position where the control will be created in the window (at the top of window for example). To drop the control in the window, simply click on the window.
3. Right-click the control that was just created. The popup menu of control is displayed. Select "Description" from this popup menu. The description window of the Button control appears.

- Modify the control characteristics by entering the following information:



1. This control is named: "BTN\_Display".
2. The control caption is: "Display".



Remark

To modify the name and caption of the Button control, we have been using the control description window (also called "7-tab window").

The control name and caption can also be modified from the window currently being edited:

1. Click the control to select it.
2. Press the Enter or Space key: the caption becomes editable.
3. Type the new caption and validate.

- Validate the control description window ("OK" button). The new control caption appears in the window editor.
- We are going to display a message in a dialog box (a small window proposed by the system). To do so, we will be using our first WLanguage function: **Info**.



Remark

The programming language supplied with WINDEV Mobile is named WLanguage. It is a 5th-generation language (5GL) that includes highly sophisticated commands.

1. Select the control if necessary.

Remarks:

- When the control is selected, several handles appear around the control.
- To select the edit control, all you have to do is click it with the mouse.

2. Display the popup menu of control (right mouse click).
3. Select "Code". This option opens the code editor of WINDEV Mobile, in which all WLanguage statements can be typed.



Remark

The code editor proposes different events for each type of control. These are the events related to the control.

Therefore, two events are displayed for the "Button" control:

- "Initializing", run when the window is displayed.
- "Click", run when the user clicks on the control.

Remark: Additional events can be added if necessary.

4. Write the following WLanguage code in the "Click" event of the "BTN\_Display" control:

```
Info ("Hello")
```

**Note about the assisted input:** As soon as the first two characters are typed, WINDEV Mobile proposes all words of WLanguage vocabulary containing these characters. The assisted development is a very powerful feature. No more mistakes when entering the name of an element: the syntax errors are reduced. All you have to do is select the requested word and press Enter to validate. You can focus on the algorithm.




Remark

When typing this WLanguage code in the code editor, you probably noticed that the different elements use different colors. This is the syntactic coloring. The code editor allows you to easily identify the different elements handled by the code:

- the WLanguage functions are colored in blue,
- the character strings (between quotes) are colored in purple,
- the names of controls are colored in cyan.

These colors can be modified element by element in the options of code editor (on the "Home" pane, in the "Environment" group, expand "Options" and select "Options of the code editor").

**Info** displays the message passed in parameter.

- Save the modifications by clicking  among the quick access button or by pressing Ctrl + S.
- Close the code editor (cross at the top right of code editor). The window editor is redisplayed.

## First test


For an Android application, WINDEV Mobile allows you to run the application test on the development computer via the simulation mode. This test simulates an Android device on the development computer. This test is useful when no Android device can be used by the developer. However, this test does not allow you to use the hardware components of device (GPS, SMS, camera, ...).



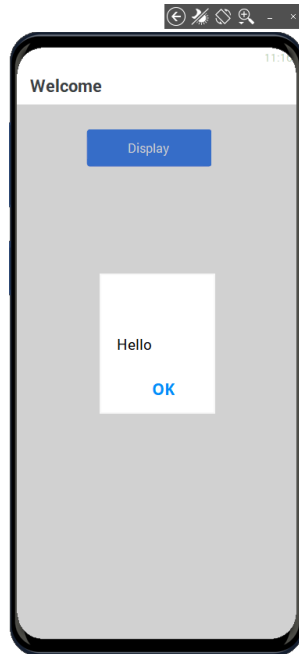
Remark

WINDEV Mobile also allows you to run a test of the application via the Android emulator (AVD) supplied with the SDK.

See the online help for more details.

- We will now run the window test in simulation mode.
1. Click  among the quick access buttons (or press F9).
  2. Validate (if necessary) the information message regarding the simulator mode.
  3. Choose (if necessary) the management mode of editor during the test (editor minimized or not).

4. The created window is started in execution.
5. Click the "Display" button.
6. Validate the system window that is displayed.



- Any developer knows that running a program test can be a long and tiresome job. In WINDEV Mobile, a SINGLE CLICK allows you to run the test of window, report or procedure while you are creating it. This is both simple and fast!
- Click "x" in the menu above the simulator box to close the window.
- The WINDEV Mobile editor appears again.

## First deployment on the device

### Principle

To run the application in stand-alone mode on the Android device, you must:

- Connect the device via a USB port.
- Generate the application. An "apk" file will be created. This file contains all elements required to run the application on an Android device.
- Select your device at the end of generation. Copying the application ("apk" file) can take several seconds.

Let's take a look at these different steps.



### Remark

**Caution:** Enabling the USB debugging is required to run tests on the phone. If this operation is not performed, the phone will not be detected by WINDEV Mobile.


To enable the USB debugging:

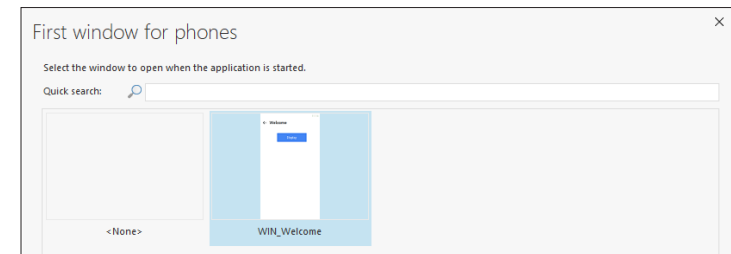
1. On the phone, select the "Parameters" menu.
2. Select "About the device".
3. Click "Build number" several times to enable the developer mode.
4. Move one level up.
5. The "Development option" choice appears. Select this option.
6. Check "USB debugging".

Remark: The operations to perform may change according to the phone version and to its make. For example, for a Samsung Galaxy Notes 3, you must "tap" several times the "Version number" control found in the "About the device" option in order to enable "Development option". In any case, a Google search with "usb debugging <device name>" allows you to get the operating mode adapted to the device used.

### Implementation

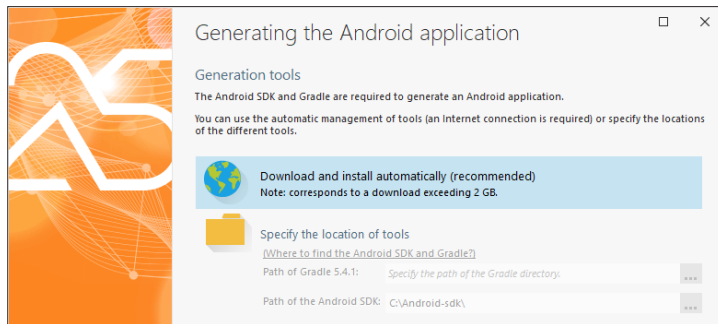
- To generate the Android application:

1. On the "Project" pane, in the "Generation" group, click "Generate" (you also have the ability to click  among the quick access buttons).
2. WINDEV Mobile proposes to select the first project window. In our example, select "WIN\_Welcome" and validate ("OK" button).



3. The Android application generation wizard starts.
4. The first step of wizard allows selecting the application generation mode. Indeed, the generation format may change depending on the mode selected. For this tutorial, select "Generate for free deployment (private store, corporate fleet, etc)". Go to the next step.

5. The next step of the wizard is to check the tools needed to generate the Android application.



The wizard proposes to:

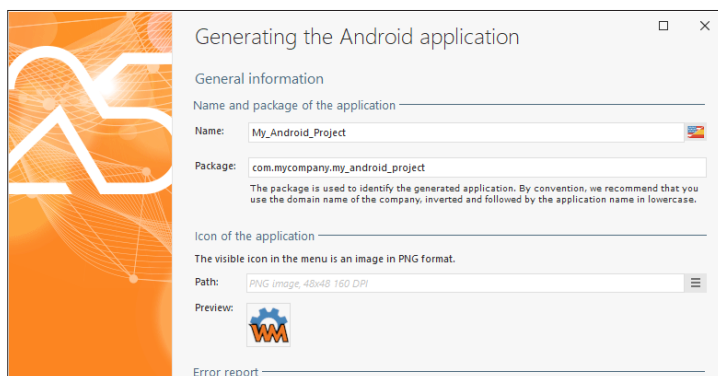
- **Download and automatically install the Android SDK and Gradle:** in this case, you will only have to validate the license. WINDEV Mobile takes care of everything. If an update is available, the generation wizard will automatically propose to perform the necessary updates. Only an Internet connection is required. This option is available only if Gradle was not downloaded and installed beforehand.
- **Use the tools automatically installed.** This option is available if the tools have been downloaded and installed during a previous generation.
- **Specify the location of tools already installed on your computer.** In this case, all you have to do is specify the setup paths of Gradle and Android SDK.

6. Select the option corresponding to your configuration and go to the next step.

Remark: If you have chosen to download and install the tools, the next step may be relatively long and require validating the license.

7. The next wizard step is used to:

- define the application name (displayed below the icon used to start the application) and the corresponding package.
- select the application icon in the image catalog of WINDEV Mobile.



8. Go to the next step. This step is used to define:

- the application splash screen,
- the information saved in the manifest,
- the options of the application:
  - the start mode of application (when starting the device or not).
  - the management of maximized windows (multi-window management or not).

9. Go to the next step. The wizard is used to define the version number of application.

10. Go to the next step. This step is used to sign the application. The wizard proposes:

- to sign the application.  
Remark: A specific signature is required to distribute the application. See the online help for more details.
- to use a generic signature that can be used for testing the application.

11. Select "Use a generic signature".

12. Go to the next step. As we have chosen to generate the application for a fleet of devices, the wizard proposes to choose the type of devices. Keep the default options.



Remark

If you have a device connected to the PC, use "Detect hardware automatically". Thus, next time an application is generated, this type of device will be automatically selected.

13. Go to the next step.

14. The wizard allows you to include specific files (data files, images, ...). This possibility will not be used in our example. Keep the default options.

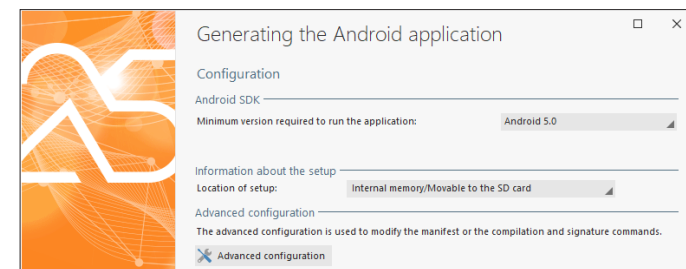
15. Go to the next step. The wizard allows you to include specific libraries. Keep the default options.

16. Go to the next step. The wizard is used to include remote dependencies used by the project. Keep the default options.


17. Go to the next step. The wizard is used to define the application permissions. By default, according to the WLanguage functions used in the application, WINDEV Mobile detects the necessary permissions.

18. Go to the next step. You have the ability to restrict the download of the application on Google Play Store to the devices equipped with the features used. Keep the default options.

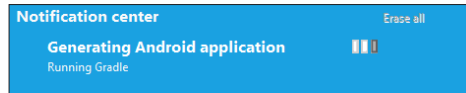
19. Go to the next step. The wizard is used to configure the options of Android SDK. Keep the default options.




20. Go to the next step.

21. End the wizard. The generation is automatically performed in background task. The  icon indicates that the generation is in progress.

22. When the generation is ended, a popup window is displayed in the editor:



Remark

This window can be displayed at any time by clicking the  icon.

23. To copy and run the application on the device linked to the computer or on an emulator, click "Deploy".

24. A new window is displayed, allowing you to select the runtime device. If you own an Android device connected to the development computer, select the device connected to the PC.



That's it, our first application is generated and run on the Android device.

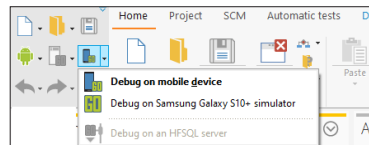


Remark

In this lesson, we ran a window test in the simulator. WINDEV Mobile allows you to run several types of tests of the entire application.

You can test your application:

- in the simulator proposed by WINDEV Mobile: select "  Debug on XXXX simulator" among the quick access options. This option allows you to quickly test the project on the PC. This is the option we use in this tutorial.
- directly on the mobile phone (default option): select "  Debug on mobile device" among the quick access options: the generation will be performed and the application will be copied and run on the cell phone connected to the current computer. You can test the application directly on your device.



## LESSON 2.2. INTERFACE (UI)

This lesson will teach you the following concepts

- Choosing the resolution according to the device.
- Window orientation.
- Touchscreen management.



Estimated time: 30 mn



## Overview

The Android system is available on the phones and on the tablets. WINDEV Mobile allows you to easily create interfaces that adapt to the device used.

## Choosing the resolution according to the device

When creating a project, you will have to choose the resolution that will be used for the project windows. Two cases may occur:

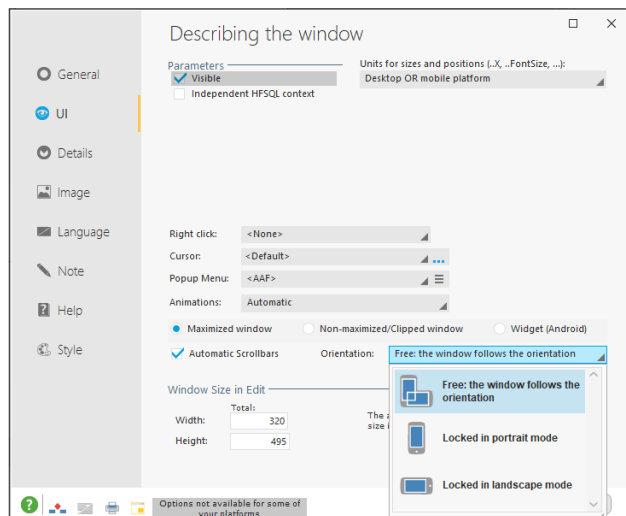
- **You are using a single target device:** in this case, all you have to do is select this device in the list proposed by the wizard.
- **You are using several target devices with different screen resolutions:** in this case, you must choose the smallest resolution common to all devices. Via the anchoring of controls in the window, the content will be adapted to the resolution.

## Window orientation

In Android, a window can have one of the following orientations:

- Free: the window follows the device orientation,
- Locked in portrait mode,
- Locked in landscape mode.

This orientation is defined in the "UI" tab of the description window of window ("Description" from the popup menu of window).



In the two last cases, no specific operation is required.

For a free window, the organization of controls and their size must adapt to the orientation. The anchoring mechanism must be used to get a proper result.

## Practical example

- Open (if necessary) the "My\_Android\_Project" project that was created in the previous lesson.




Answer

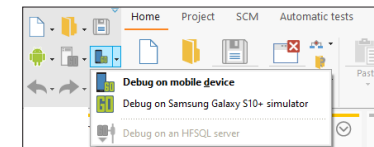
A corrected project is available. To open this project, in WINDEV Mobile's home page (Ctrl + <), click "Tutorial" and select "My Android project (Answer)".

In our example, the project was created for a phone and it was tested in portrait mode in the simulator.



We are now going to run its test in landscape mode in the simulator.

- Run the project test in simulator mode:

1. In the quick access button area, expand "GO" if necessary and select the option "  Debug on Generic Android Phone simulator".



Remark

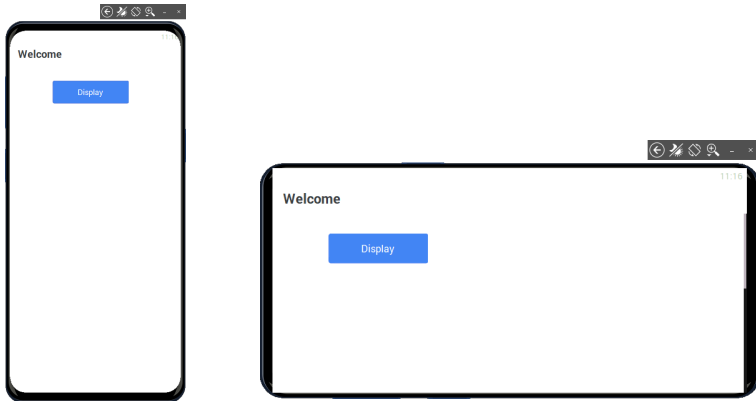
By default, the project test on a mobile device is proposed among the quick access buttons. After selecting "  Debug on simulator" for the first time, the corresponding icon (  ) will be automatically proposed among the quick access buttons.

2. Validate (if necessary) the window that opens up.
3. The window is displayed in portrait mode.
4. The simulator shows an icon-based menu:



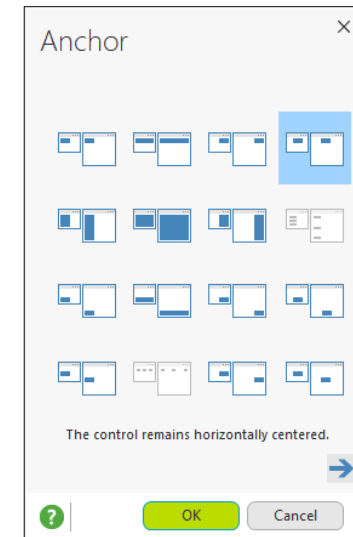
5. Modify the window orientation with the "Rotation" option .

6. The window orientation changes on the screen. In our example, the button location does not change: it does not adapt to the screen orientation.



- We are now going to modify our window in order for the "Display" Button control to be centered and to remain centered regardless of the device orientation.
- Stop the test and go back to the editor.
- To center the Button control in the window:
  1. Select the control (click on the Button control).
  2. On the "Alignment" pane, in the "Centering and distribution" group, click "Center in the parent (horz)".
- In order for the Button control to remain centered in the window, we are going to use the control anchoring:
  1. Select the control if necessary.
  2. Display the popup menu (right mouse click).

3. Select "Anchor": the window for defining anchors is displayed:




4. Select "Horizontally centered" and validate ("OK" button).



**Remark**

In the window displayed in the editor, you will notice the little red arrows in the control. These arrows indicate that the control is anchored.

- Run the project test on simulator (  among the quick access buttons):
  - The button is centered in portrait mode.
  - Change the orientation of simulator.
  - The button remains centered in landscape mode.

## Touchscreen management

One of the most important aspects of the interface for a mobile application is the touchscreen management.

A "multi-touch" feature is a technique allowing the user to interact with a device via several contact points.

Handling images is one of the most common multi-touch features. The display size on a phone being reduced, it is often necessary to zoom an image and/or to move inside an image.

This allows you to perform a zoom on an image via the contact of 2 fingers moving apart.

To manage the "multi-touch", WINDEV Mobile proposes:

- Specific options available in the Image control.
- Specific WLanguage functions.
- Specific optional processes.


See the online help for more details.

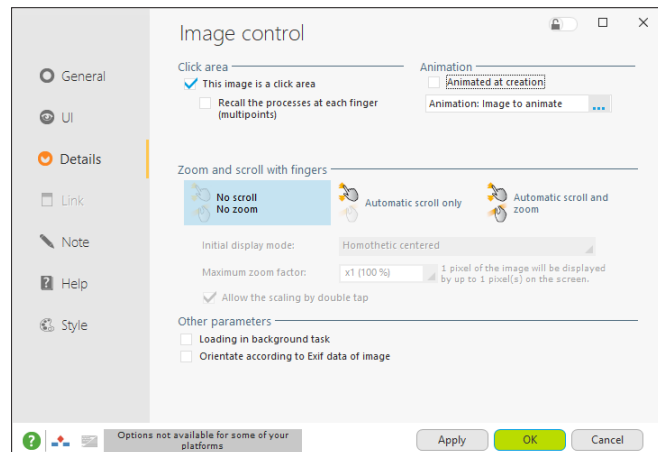
**Practical example**


- Open (if necessary) the "My\_Android\_Project" project that was created in the previous lesson.

**Answer**

A corrected project is available. To open this project, in the WINDEV Mobile home page (Ctrl + <), click "Tutorial" and select "My Android project (Answer)".

- In the "WIN\_Welcome" window, create an Image control:
1. On the "Creation" pane, in the "Usual controls" group, click "Image".
  2. The Image control appears in creation under the mouse.
  3. Move the mouse in the window toward the position where the control will be created. To drop the control in the window, simply click again.
  4. Double-click the Image control: the description window of control is displayed.
  5. In the "General" tab, select an image from your disk in the "Image" edit control (via , "Browse").
  6. If the image is found in a directory other than the project directory, WINDEV Mobile proposes to copy the image file into the project directory. Accept by clicking the "Copy the file into the suggested directory" button.
  7. Display the "Details" tab: the options for multi-touch management are displayed:




8. Select "Automatic scroll and zoom".
9. Validate the control description window.
10. Save the window (click  among the quick access buttons).

11. A UI error appears in the error pane: the automatic window scrollbars are in conflict with the scrolling features of Image control.

12. To avoid this UI error, disable the window scrollbars:

- Display the description window of window ("Description" from the popup menu).
- In the "UI" tab, un check "Automatic Scrollbars".
- Validate the description window.

13. Save the window (click  among the quick access buttons). The UI error disappears.

**Remark**

The multi-touch management cannot be checked in the simulator. To check this feature, the application must be tested (or deployed) on the mobile device.

14. Close the project.

**The different types of controls**

WINDEV Mobile proposes several controls. These controls are used to display or enter data. Some controls are specifically intended for a mobile interface.

To develop your applications, you can use the standard controls (edit controls, images, radio buttons and check boxes) but also more specific controls such as:

- the Multiline Zone control to create interfaces (UI) similar to the native Android windows,
- the Map control to view a position on a map or an itinerary,
- the Ad control to display an advertising banner,
- the menu in the shape of "Action bar".

Some of this controls will be presented in the lesson "Developing the application", page 104.

## LESSON 2.3. DISTRIBUTING THE APPLICATION

### This lesson will teach you the following concepts

---

- Generating the APK.
- Available distribution modes.



Estimated time: 20 mn

### Overview

---

WINDEV Mobile allows you to develop applications for the Android operating system.

Once the applications are created, developed and checked, all you have to do is deploy them.

Several deployment modes are available:

- Deployment via Google Play (or another market).
- Deployment on a Web server.
- Deployment from the PC via ADB.
- Deployment by copy.

### Deployment via Google Play

---

Google Play is an online service used to download applications (free of charge or not) on mobiles devices compatible with Android.

Once published, the application can be downloaded by the users all around the world via the Google Play application installed on their phone.

The publication of applications on Google Play must comply with specific rules:

- During the first publication, you must register beside the Google Play service via a Google account. Once registered, you have the ability to publish or update as many applications as you want as many times as necessary.
- The published application must be signed with a private cryptographic key. You have the ability to sign your own application: using a third-party organism is optional. The validity period of certificate must end after October 23, 2033.

The applications generated by WINDEV Mobile are automatically signed by using the information specified in the generation wizard ("Signature of application" step) with a sufficient validity period.

**Caution:** The published application must not be signed with a generic key (whose use must be limited to the tests in GO mode).

- An icon must be associated with the application. The generation wizard of WINDEV Mobile allows you to define the icon to use.

**Remark:** Google Play is the most common application but other applications are available.

### Deployment via a Web server

---

You have the ability to propose Android applications for download from a link on a Web page.

To do so, you must:

1. Copy the "apk" file of the application onto the Web server that hosts the page proposing the download of the application.

Reminder: the "apk" file is created by WINDEV Mobile when generating the Android application.

2. Add a link into the Web page for download. This link has the following format:

```
<a href='Path of apk file on the server'>Link</a>
```

3. On the server, add the following MIME type: application/vnd.android.package-archive.

The user will only have to display the page with the browser of the phone. The application will be downloaded when the link is clicked. Then, all you have to do is click the downloaded file (in the download manager) to install the application.

**Caution:** The "Unknown sources" option must be enabled on the phone to allow this setup mode. To enable this option, go to the "Parameters" menu of phone, in the "Applications" sub-menu.

### Deployment from the PC via ADB (advanced mode)

ADB (Android Debug Bridge) is a tool supplied with the Android SDK. It is used, among other things, to install or uninstall from the PC an Android application (APK file) on a mobile device compatible with Android.

This setup mode is an advanced mode. We recommend that you see in the online help the specific commands used to install an application. To do so:

- go to "<https://developer.android.com/studio/index.html>".
- in the menu, select "User guides" then "Command line tools" and finally "adb".

**Caution:** The "Unknown sources" option must be enabled on the phone to allow this setup mode. To enable this option, go to the "Parameters" menu of phone, in the "Applications" sub-menu.

### Copying the application onto Mobile device

The easiest way to install an Android application on a mobile device is to copy the apk file onto the device and to run it. The following operations must be performed:

1. Connect the device to the PC by USB.
2. Copy the apk file of application onto the device (external memory for example).  
Reminder: the "apk" file is created by WINDEV Mobile when generating the Android application.
3. On the device, use a file explorer to go to the directory where the apk file was copied and click the file to start its setup.

**Remark:** Some devices do not propose a file explorer but several ones are available for free.

**Caution:** The "Unknown sources" option must be enabled on the phone to allow this setup mode. To enable this option, go to the "Parameters" menu of phone, in the "Applications" sub-menu.

iOS  
application

## LESSON 3.1. MY FIRST iOS PROJECT

**This lesson will teach you the following concepts**

---

- Required configuration.
- Creating an iOS project (iPhone or iPad).
- My first window.
- My first test.
- First deployment.



Estimated time: 40 mn



## Overview

To start developing with WINDEV Mobile for an iOS platform, we are going to create a first project. This project will contain a window used to display a message.

This first example will present the main concepts of development for iOS with WINDEV Mobile. Before creating our first project for iOS, the development computer must be configured.

## Required configuration for iOS

To develop a WINDEV Mobile application for iPhone/iPad, you must own:

- 1 PC,
- 1 Mac,
- 1 iPhone and/or iPad (optional).

### Why a PC?

WINDEV Mobile 25 is a Windows application that can be used in Windows 7, 10, ...

The application will be created on PC before it is compiled on Mac.

This PC requires no setup of Mac/Apple tools.

### Why a Mac?

A Mac is required because the project generated on PC must be compiled in a specific compiler to generate iOS applications. The minimum version of the operating system must be iOS 11.

Xcode is a development environment that is used to develop iOS applications (iPhone and iPad). This tool will be used to compile the applications generated with WINDEV Mobile.

The minimum version of Xcode must be version 11.

See the online help for more details (download addresses, ...).

## My first project

### Creating the project

We are now going to create our first project for iOS.

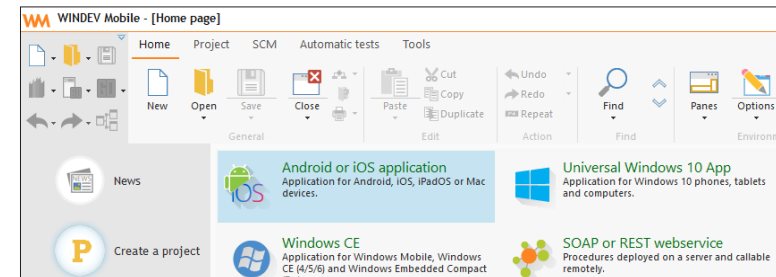


Answer

A corrected project is available. To open this project, in WINDEV Mobile's home page (Ctrl + <), click "Tutorial" and select "My iOS project (Answer)".

► To create a project:

1. Start WINDEV Mobile 25 (if not already done).
2. Display the WINDEV Mobile home page if necessary (Ctrl + <).
3. On the home page, click "Create a project" then "Android or iOS application".




4. The wizard for project creation starts. The different steps of the wizard help you create your project. The information specified in this wizard can be modified later.

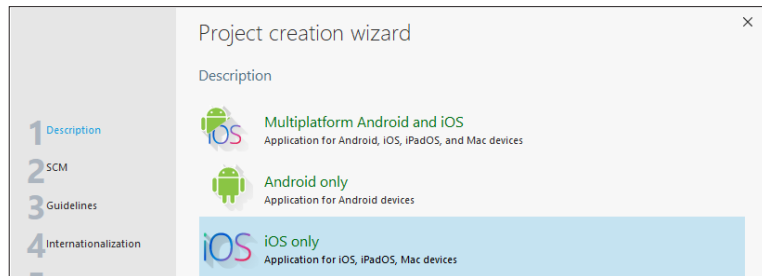


Remark

Tip: Other method for creating a project:

1. Click  among the quick access buttons of WINDEV Mobile menu.
2. The window for creating a new element is displayed: click "Project".

5. The first step of the wizard allows you to define the project execution platform:



In our case, select "iOS only" and go to the next step of the wizard.

6. The wizard proposes to create a blank project or a project based on an example. Select "Create a blank project" and go to the next step.

7. The wizard proposes to choose the type of devices affected by the project:

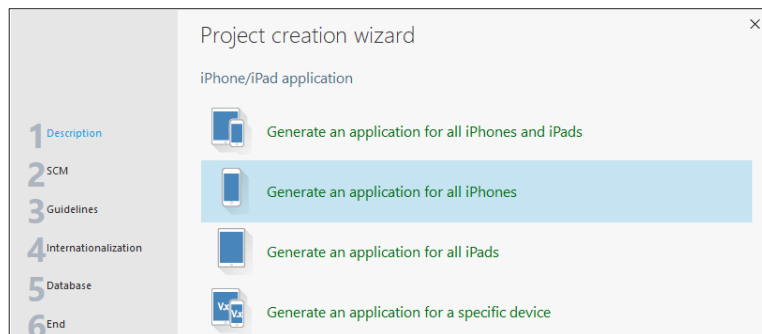
- Generate an application for all iPhones and iPads.
- Generate an application for all iPhones.
- Generate an application for all iPads.
- Generate an application for a specific device.

**Remark**

If the application is intended to operate on several iOS devices (phones with different sizes or with different resolutions for example), we advise you to use one of the following options;

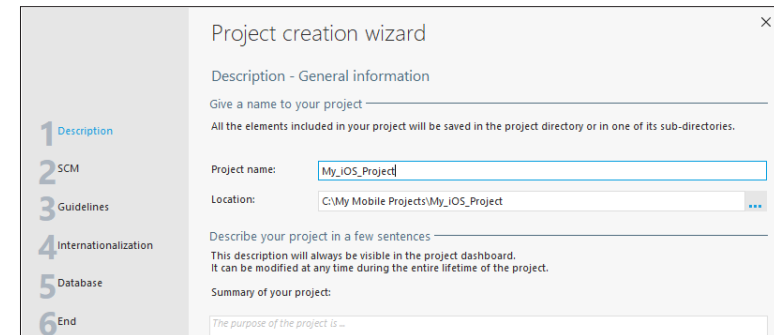
- "Generate an application for all iPhones and iPads",
- "Generate an application for all iPhones".

In this case, WINDEV Mobile proposes the smallest resolution to create the application windows. Using anchors (see "Interface (UI)", page 65 and "Developing the application", page 104) will allow the application to operate on all devices.



8. For this example, select "Generate an application for all iPhones" and go to the next step.

9. The wizard proposes to type the name of project, its location and its description.

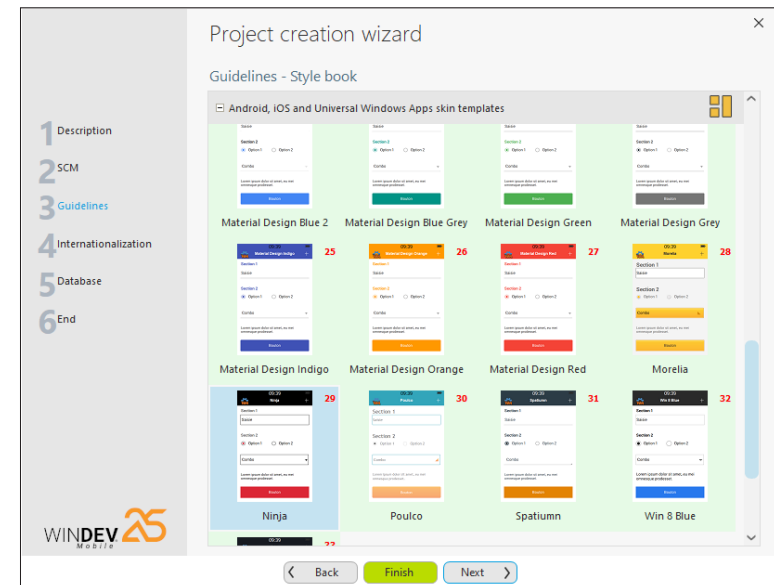


In our case, this project will be named "My\_iOS\_Project". WINDEV Mobile proposes to create this project in the "\My Mobile projects\My\_iOS\_Project" directory. You can keep this location or modify it via the [...] button.

10. Go to the next step ("Next" button).

11. In the left section of the wizard, click "Guidelines". This step is used to define the code style. Don't modify the suggested options. Go to the next step.

12. This step is used to define the style book of application. We will choose "Ninja".



13. The other steps of the wizard are not important for our first project, so click "Finish". The project is automatically created.

14. The window for creating a new element is displayed. This window is used to create all elements that can be associated with a project.



## My first window

### Overview

The first window that we will create allows the user to display a welcome message via the "Display" button.

You may think this is too basic but we advise you to create this window. You may be surprised by how intuitive and how easy it is to use the editor of WINDEV Mobile. Furthermore, this window will allow you to discover concepts that are fundamental for the rest of this tutorial and to see the entire process for developing an iOS application with WINDEV Mobile.

### Creating the window

► To create the window:

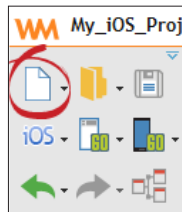
1. In the window for creating a new element, click "Window" then "Window".



Remark

As a new project was created, the window for creating a new element is automatically displayed.

To display the window for creating a new element, all you have to do is click among the quick access buttons of WINDEV Mobile:



2. The window creation wizard starts.

3. In the list of proposed windows, select "Blank". The skin template used is displayed at the bottom right of wizard. "Ninja", the skin template that was selected when creating the project, is selected by default. You can choose another skin template proposed in the list.



Remark

Skin templates allow you to quickly create outstanding interfaces. A skin template defines the window style as well as the style of all controls that will be used in this window. Thus, there is no risk of obtaining an ugly interface.

4. Validate. The window is automatically created in the editor. The window for saving an element is displayed. This save window shows:

- the element title. For a window, this title will be displayed in the Action Bar of window.
- the element name that corresponds to the window name. This name will be used in programming.
- the element location. This location corresponds to the directory in which the physical file corresponding to the element is saved. The window is a "WDW" file, saved in the project directory.

5. Specify the title of the element: "Welcome". The name of the element is automatically proposed: "WIN\_Welcome".



Remark

Let's take a look at the window name proposed by WINDEV Mobile: this name starts with the letters "WIN\_". This prefix is automatically added because the project is using a code style.

The code style is used to define a prefix for each type of object, allowing you to quickly identify the element:

- a window starts with "WIN\_",
- a button starts with "BTN\_",
- etc.

If you do not want to use this code style, you can simply disable it: on the "Project" pane, in the "Other actions" group, expand "Code style" and uncheck "Use code style".

6. Save the window by clicking "OK".

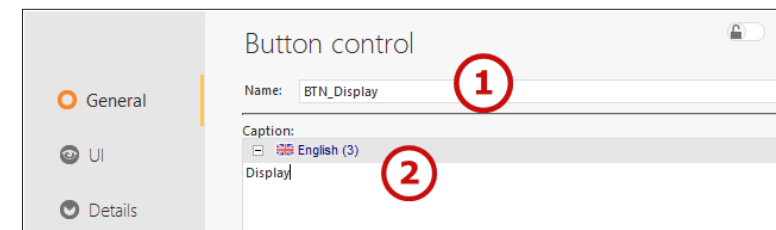
### Displaying a message

You are now going to create a Button control used to display a message.

► To create the Button control:

1. On the "Creation" pane, in the "Usual controls" group, click **Ok**. The control appears in creation mode under the mouse pointer.
2. Move the mouse toward the position where the control will be created in the window (at the top of window for example). To drop the control in the window, simply click again.
3. Right-click the control that was just created. The popup menu of control is displayed. Select "Description" from this popup menu. The description window of the Button control appears.

► Modify the control characteristics by entering the following information:



1. This control is named: "BTN\_Display".
2. The control caption is: "Display".



Remark

To modify the name and caption of the Button control, we have been using the control description window (also called "7-tab window").

The control name and caption can also be modified from the window currently being edited:

1. Click the control to select it.
2. Press the Enter or Space key: the caption becomes editable.
3. Type the new caption and validate.

- Validate the control description window ("OK" button). The new control caption appears in the window editor.
- We are going to display a message in a dialog box (a small window proposed by the system). To do so, we will be using our first WLanguage function: **Info**.



Remark

The programming language supplied with WINDEV Mobile is named WLanguage. It is a 5th-generation language (5GL) that includes highly sophisticated commands.

1. Select the control if necessary.

Remarks:

- When the control is selected, several handles appear around the control.
- To select the edit control, all you have to do is click it with the mouse.

2. Display the popup menu of control (right mouse click on the control).

3. Select "Code". This option opens the code editor of WINDEV Mobile, in which all WLanguage statements can be typed.



Remark

The code editor proposes different events for each type of control. These are the events related to the control.

Therefore, two events are displayed for the "Button" control:

- Initializing, run when displaying the window.
- Click, run when the user clicks on the control.

Remark: Additional events can be added if necessary.

4. Write the following code in the "Click" event of the "BTN\_Display" control:

```
Info("Hello")
```

**Note about the assisted input:** As soon as the first two characters are typed, WINDEV Mobile proposes all words of WLanguage vocabulary containing these characters. The assisted development is a very powerful feature. No more mistakes when entering the name of an element: the syntax errors are reduced. All you have to do is select the requested word and press Enter to validate. You can focus on the algorithm.




Remark

When typing this WLanguage code in the code editor, you probably noticed that the different elements use different colors. This is the syntactic coloring. The code editor allows you to easily identify the different elements handled by the code:

- the WLanguage functions are colored in blue,
- the character strings (between quotes) are colored in purple,
- the names of controls are colored in cyan.


These colors can be modified element by element in the options of code editor (on the "Home" pane, in the "Environment" group, expand "Options" and select "Options of the code editor").

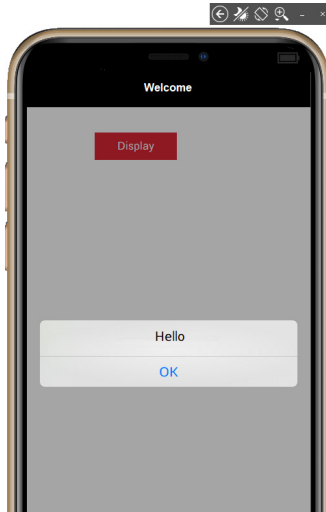
**Info** displays the message passed in parameter.

- Save the modifications by clicking  among the quick access button (on the left of ribbon) or by pressing Ctrl + S.
- Close the code editor (cross at the top right of code editor). The window re-appears.

## First test

For an iOS application, WINDEV Mobile offers several types of tests:

- **Project or window test on the development computer using the simulation mode.** This test simulates an iOS device on the development computer. This test is useful when the developer does not have a Mac or an iOS device to compile the application. However, this test does not allow you to use the hardware components of device (GPS, SMS, camera, etc).
  - **Project test directly on the device.** This test is useful when the developer does not have a Mac to compile the application. All the components of the device are accessible.
- We will now run the window test in simulation mode.
    1. Click  among the quick access buttons (or press F9).
    2. Validate (if necessary) the information message regarding the simulator mode.
    3. The created window is started in execution, in a shell corresponding to the selected device (iPad or iPhone).
    4. Click the "Display" button.
    5. Validate the system window that is displayed.



6. Click "x" in the menu above the simulator box to close the window.

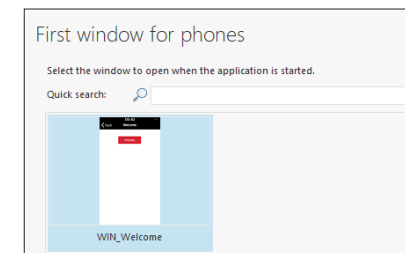
7. The WINDEV Mobile editor appears again.

- Do you have an iPhone? Now, let's test the project directly on the device.
  - First of all, make sure all the following conditions are met:
    - the iPhone/iPad and the development workstation must be on the same network (Wifi, for example).
    - WMDev must be downloaded to the iPhone/iPad (<https://itunes.apple.com/us/app/WMDev/id1437792304>).
    - the following network ports must be open:
      - 64000 (64000 à 64005 if several WINDEV Mobiles are running simultaneously).
      - 27280
- If a firewall is used, it must be configured to allow the use of these ports.

- Among the quick access buttons, select "Debug on a mobile device". The window for connecting to an iOS device appears. This window contains a specific QR code.



1. Start WMDev on the mobile device (iPhone or iPad).
  - Click on "+" and then on the button displaying a bar code.
  - Scan the barcode displayed under WINDEV Mobile in the connection window of an iOS device.
2. WINDEV Mobile proposes to select the first project window. In our example, select the "WIN\_Welcome" window and validate ("OK" button).



3. The mobile application starts on the iOS device.

## First deployment on the device

### Principle


To run the application in stand-alone mode on the device, you must:

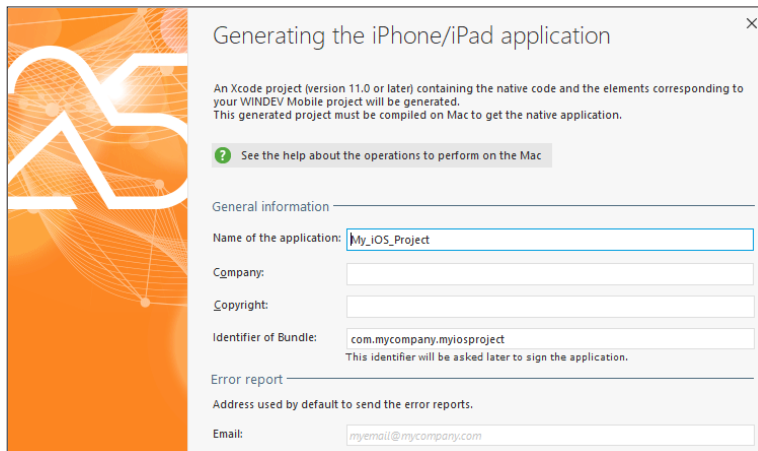
- Generate the iOS application (or Xcode project) in WINDEV Mobile.
- Transfer the generated Xcode project onto Mac in order to compile it.
- Compile the project in Xcode in order to generate the program.
- Then, the program can be installed on the connected device or started in the Xcode emulator. It will run in stand-alone mode.

Let's take a look at these different steps.

### Implementation

► To generate the iOS application:

1. On the "Project" pane, in the "Generation" group, click "Generate" (you can also click  among the quick access buttons).
2. The wizard for generating an iPhone/iPad application starts.



3. The first step of the wizard allows you to:
  - define the application name, the company and the copyright.
  - enter the bundle identifier.



Remark

This identifier is the unique identifier of your application beside Apple. It is defined and saved on the Apple developer account.

This identifier will be used to save your application in order to run its test and to deploy it. By default, WINDEV Mobile automatically proposes an identifier that respects the development standard of Apple. This identifier can be modified.

- type the email address that will be used by default to send the error reports generated by the application.
4. Go to the next step. You have the ability to specify the path of different application icons. Several icons can be supplied:
    - icons for iPad and iPad Retina (only if the application proposes windows for iPad),
    - icons for all iPhone models (for example, iPhone 4S and iPhone 6 Plus have different resolutions).
  5. Go to the next step.
  6. Specify the characteristics of the splash screen.
  7. Go to the next step. Specify the version number of the generated application.
  8. Go to the next step. This step is used to include specific files (data files, images, ...). This possibility will not be used in our example. Keep the default options.
  9. Go to the next step. This step is used to specify:
    - whether the application is allowed to be resized or not (Split View). The option allows the end user to display two windows of two different applications side by side on some devices.
    - whether files can be shared with iTunes. If this option is checked, you will have the ability to retrieve the application files on Mac during the synchronization. For example, if data files have been supplied with the application, the iTunes application will allow you to retrieve these files.
    - the minimum iOS version required to run the application.
  10. Go to the next step. This step is used to define the signature options. See the online help for more details.
  11. Validate the wizard ("Finish"). The generation is performed in the EXE folder of project directory. The directory containing the source code that will be used on Mac is named "Project\_name.xcode.gen" ("My\_iOS\_Project.xcode.gen" in our example). This is the directory that must be copied onto Mac. Click the "Open the generation directory" button.

The other operations must be performed on Mac. You must:

- Transfer the WINDEV Mobile project onto Mac.
- Compile the project in Xcode.

**Transferring the WINDEV Mobile project onto Mac**

- To transfer the WINDEV Mobile project onto MAC:
  1. Copy the entire folder generated in the EXE directory onto an external media (USB key, external hard disk, shared directory with MAC on network). This directory is named <Project Name>.xcode.gen.
  2. Paste this folder on the MAC that will compile the application.
  3. Open the folder on MAC and open the file named "Project\_name.xcodeproj".
  4. The project is opened in Xcode.

**Compiling the project in Xcode**

WINDEV Mobile automatically generates an Xcode project for your iOS compilations. To simplify the implementation of applications, WINDEV Mobile generates a "Scheme" for Xcode.

- To compile the project in Xcode:
  1. In the drop-down list found in the top left corner, select the compilation options. Select the scheme (corresponding to your application) then the compilation target (device currently connected or a simulator).
  2. To start the compilation, select "Product .. Clean" then "Product .. Build".
  3. A compilation status report appears at the top ("Succeeded", or the number of warnings and errors). You have the ability to click these symbols to see the list of errors/warnings.
  4. Once the program is compiled without error, you can start the simulation ("Product .. Run").
    - If the compilation target is the simulator, the application is started in the simulator window.
    - If the compilation target is the connected device, the application is started on the device.

Then, you have the ability to run the test of your application on your iPhone or iPad.

## LESSON 3.2. INTERFACE (UI)

**This lesson will teach you the following concepts**

---

- Choosing the resolution according to the device.
- Window orientation.
- Touchscreen management.



Estimated time: 20 mn

## Overview

The iOS system is available on the phones (iPhone), on the tablets (iPad) and on the iPod. WINDEV Mobile allows you to easily create interfaces that adapt to the device used.

## Choosing the resolution according to the device

When creating a project, you must choose the device on which the application will be deployed:

- iPhone,
- iPad,
- iPhone and iPad.

Two cases may occur:

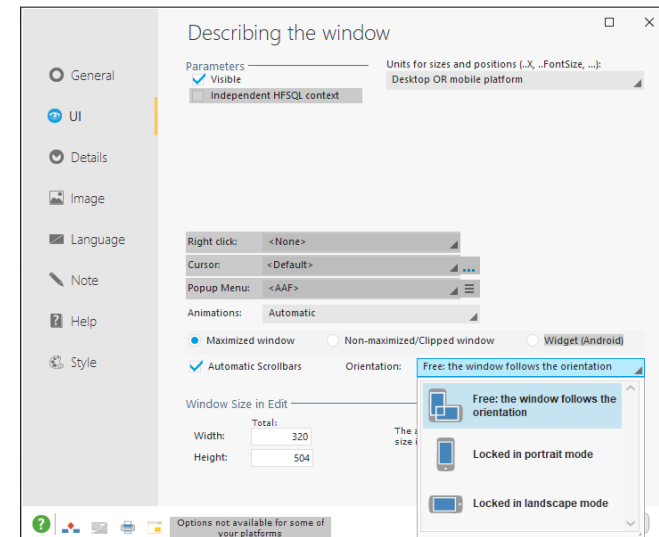
- **You know the target device:** in this case, all you have to do is select it. The wizard for window creation will allow you to create windows for this target device.
- **You don't know the target device:** in this case, in the wizard for project creation, select "iPhone and iPad". When creating the windows, you can:
  - develop your windows for iPhone. Via the anchoring of controls in the window, the content will be adapted to the iPad (recommended solution).
  - use the layouts to create different interfaces according to the runtime platform and to its resolution.
  - develop 2 sets of windows, one for iPhone, one for iPad.

## Window orientation

In iPhone or iPad, a window can have one of the following orientations:

- Free: the window follows the device orientation,
- Locked in portrait mode,
- Locked in landscape mode.

This orientation is defined in the "UI" tab of the description window of window ("Description" from the popup menu of window).



In the two last cases, no specific operation is required.

For a free window, the organization of controls and their size must adapt to the orientation. The anchoring mechanism must be used to get a proper result.

### Practical example

- Open (if necessary) the "My\_iOS\_Project" project that was created in the previous lesson.





Answer

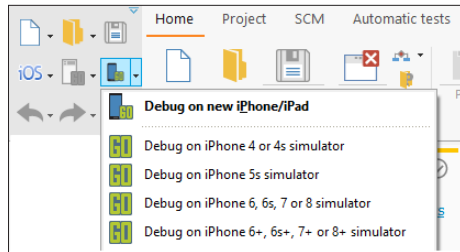
A corrected project is available. This project contains the different windows created in this lesson. To open the corrected project, in WINDEV Mobile's home page (Ctrl + <), click "Tutorial" and select "My iOS project (Answer)".

In our example, the project was created for a phone, and we have tested it in portrait mode in the simulator.



We are now going to run its test in landscape mode in the simulator.

- Run the project test in simulator mode:

1. In the quick access button area, expand (if necessary) "  " and select "  Debug on iPhone xxx simulator".




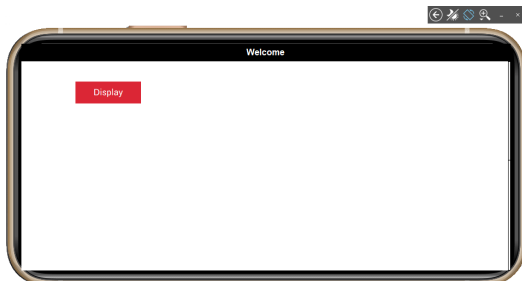
## Remark

By default, the project test on a mobile device is proposed among the quick access buttons. After selecting "  Debug on simulator" for the first time, the corresponding icon (  ) will be automatically proposed among the quick access buttons.

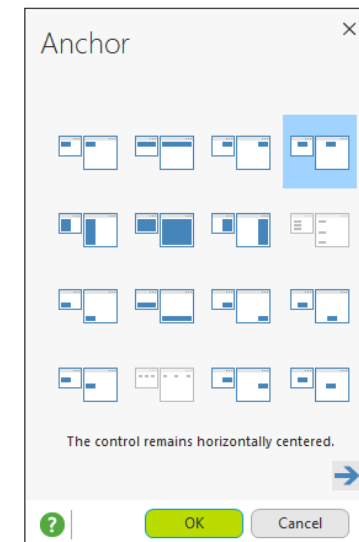
2. Validate (if necessary) the window that opens up.
3. The window is displayed in portrait mode.
4. The simulator shows an icon-based menu:



5. Modify the window orientation with the "Rotation" option (  ).
6. The window orientation changes on the screen. In our example, the button location does not change: it does not adapt to the screen orientation.



- We are now going to modify our window in order for the "Display" Button control to be centered and to remain centered regardless of the device orientation.
- Stop the test and go back to the editor.
- To center the Button control in the window:
  1. Select the Button control (click on the control).
  2. On the "Alignment" pane, in the "Centering and distribution" group, click "Center in the parent (horz)".
- In order for the Button control to remain centered in the window, we are going to use the control anchoring:
  1. Select the control if necessary.
  2. Display the popup menu (right mouse click).
  3. Select "Anchor": the window for defining anchors is displayed:




4. Select "Horizontally centered" and validate ("OK" button).



## Remark

In the window displayed in the editor, you will notice the little red arrows in the control. These arrows indicate that the control is anchored.

- Run the project test in simulator mode (  among the quick access buttons):
  - The button is centered in portrait mode.
  - Change the orientation of simulator.
  - The button remains centered in landscape mode.

## Touchscreen management

One of the most important aspects of the interface for a mobile application is the touchscreen management.

A "multi-touch" feature is a technique allowing the user to interact with a device via several contact points.

Handling images is one of the most common multi-touch features. The display size on a phone being reduced, it is often necessary to zoom an image and/or to move inside an image.

This allows you to perform a zoom on an image via the contact of 2 fingers moving apart.

To manage the "multi-touch", WINDEV Mobile proposes:

- Specific options available in the Image control.
- Specific WLanguage functions.
- Specific optional events.

See the online help for more details.


### Practical example

- Open (if necessary) the "My\_iOS\_Project" project that was created in the previous lesson.

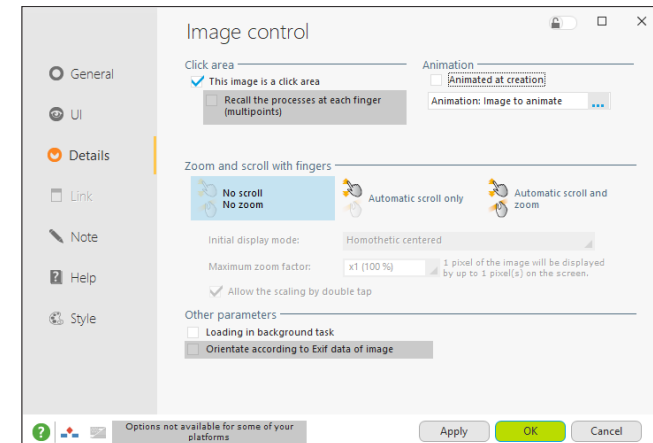




Answer

A corrected project is available. To open this project, in the WINDEV Mobile home page (Ctrl + <), click "Tutorial" and select "My iOS project (Answer)".

- In the "WIN\_Welcome" window, create an Image control:
1. On the "Creation" pane, in the "Usual controls" group, click "Image".
  2. The Image control appears in creation under the mouse.
  3. Move the mouse in the window toward the position where the control will be created. To drop the control in the window, simply click again.
  4. Double-click the Image control: the description window of control is displayed.
  5. In the "General" tab, select an image from your disk in the "Image" edit control (via , "Browse").
  6. If the image is found in a directory other than the project directory, WINDEV Mobile proposes to copy the image file into the project directory. Accept by clicking the "Copy the file into the suggested directory" button.

7. Display the "Details" tab: the options for multi-touch management are displayed:



8. Select "Automatic scroll and zoom".
9. Validate the control description window.
10. Save the window (click  among the quick access buttons).
11. A UI error appears in the error pane: the automatic window scrollbars are in conflict with the scrolling features of Image controls.
12. To avoid this UI error, disable the window scrollbars:
  - Display the description window of window ("Description" from the popup menu).
  - In the "UI" tab, uncheck "Automatic scrollbars".
  - Validate the description window.
13. Save the window (click  among the quick access buttons). The UI error disappears.



Remark

The multi-touch management cannot be checked in the simulator. To check this feature, the application must be deployed on the mobile device. We will see how to proceed in the "Android and iOS: Deploying an application" section.

14. Close the project.

## The different types of controls

WINDEV Mobile proposes several controls. These controls are used to display or enter data. Some controls are specifically intended for a mobile interface.

To develop your applications, you can use the standard controls (edit controls, images, radio buttons and check boxes) but also more specific controls such as:

- the Multiline Zone control to create interfaces (UI) similar to the native iOS windows.
- the Map control to view a position on a map or an itinerary.

Some of these controls will be presented in the "Android and iOS: Developing an application" lesson, page 95.



## LESSON 3.3. DISTRIBUTING THE APPLICATION

### This lesson will teach you the following concepts

---

- Available distribution modes.



Estimated time: 5 mn

### Overview

---

WINDEV Mobile allows you to develop applications for the iOS operating system. Once the applications are created, developed and checked, all you have to do is deploy them.

### Deployment

---

Three methods can be used to deploy the application on a device (iPhone or iPad).

- Via App Store: This type of distribution allows you to distribute your application via the App Store without any restrictions. Your application will include the signature linked to your certificate but it will not be linked to a single device.
- Via an In-House network: This type of distribution allows you to distribute your application via a Web server to a group of users working for the same company. To use this type of distribution, you must register to the iOS Developer Enterprise program. Your application will include the signature linked to your certificate but it will not be linked to a single device.
- Via an Ad Hoc network: This type of distribution allows you to install the application on an Ad Hoc network containing up to 100 devices (iPhone and iPad). The application must be recompiled for the target device by including the certificate for the signature as well as the unique identifier of the device.

**Caution:** To run the application test and/or to deploy the application on a device (iPhone or iPad), you must register beside the iOS Developer Program. This registration is not free of charge. For more information, please visit: <https://developer.apple.com/programs/whats-included/>

Three types of registration are available:

- iOS Developer Program - Individuals.
- iOS Developer Program - Organizations.
- iOS Developer Enterprise Program.

This registration is used to get a developer certificate allowing you to sign your applications in order to compile them and to distribute them. This certificate is not free of charge. This developer certificate is required even for a simple setup for test (debug) on a device.

**PART 4**

Programming  
concepts



## LESSON 4.1. CONCEPTS AND TERMINOLOGY

**This lesson will teach you the following concepts**

---

- Main concepts of WINDEV Mobile.
- Terminology used by WINDEV Mobile.



Estimated time: 10 mn

## Overview

In the previous parts, we have created our first Android application and/or our first iOS application. After these exercises, let's talk about the main concepts of WINDEV Mobile and about the terminology specific to WINDEV Mobile.

## Main concepts

WINDEV Mobile allows you to easily create an application. But what is an **Application**? An **application** is a tool used to automatically perform tasks, actions.

To create an application, WINDEV Mobile proposes to create a **project**. A project links and organizes the different program elements. The program corresponding to the application will be created from the project.

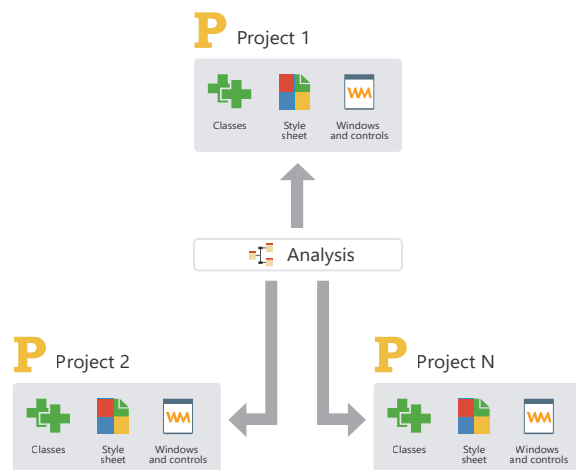
If your application is using data, WINDEV Mobile allows you to define the database structure via **the analysis**. The WINDEV Mobile analysis contains the description of the data files (also known as "Tables" in several databases). These data files will contain the application data.



Remark

Describing the data files in the analysis does not mean that they are created. The data files are physically created when running the application.

One or more WINDEV Mobile projects can be linked to the same analysis. In this case, we talk of shared analysis. For example, an application for business management can be divided into several modules. Each module is using the same analysis (at run time, each application can also use the same data files).



## Terminology

As already seen, a WINDEV Mobile project (linked to an analysis if necessary) is used to create an application. Before we actually start working with WINDEV Mobile, let's go back to the vocabulary used in WINDEV Mobile. Indeed, several terms are specific to WINDEV Mobile and they may differ from the ones used in other tools.

The following terms are used in **the analysis**:

- **Data file:** The analysis is used to describe the structure of database files. A "Data file" corresponds to a "table" in some databases.  
In WINDEV Mobile, "**Table**" represents a graphic object used to view the content of a data file in a table and/or to enter rows. A table can be used to type the order details for example.
- **Record:** A record is sometimes called row. A data file record corresponds to all items defined for the data file.
- **Item:** In the analysis, an item represents a section of a data file. All items found in a data file are used to define the structure of a record.
- **Key/Index:** With WEBDEV Mobile and its HFSQL database, the concept of index is linked to the concept of key. The concept of key is part of the item characteristics. The keys are used to improve the speed for accessing data and to simplify the browse operations performed on the data files. In WINDEV Mobile, if a HFSQL data file includes several key items, a single index file will be created at run time.

The following terms are used in **the windows and reports**:

- **Window:** Windows are used to display or type information on the screen. Windows are also called "Screens" or "Dialog boxes". The user can directly act on the windows via controls, buttons, ...
- **Report:** Reports are used to get a custom view of information. This information can come from the database, from text files, from controls found in the windows, ... Reports can be generated in PDF in mobile applications.
- **Control:** "Control" is the term used to identify the different graphic objects displayed in a window or report.
- **Skin template:** The skin template is used to define the application style: visual appearance of windows, buttons, controls, ...
- **Style:** The style groups the graphic characteristics of an element: background image, border, font, ... The styles of different elements found in the interface of a WINDEV Mobile application are grouped in a style sheet.

## LESSON 4.2. THE WLANGUAGE BASICS

### This lesson will teach you the following concepts

- The different types of variables.
- Main WLanguage statements.
- The main WLanguage operators.
- Procedures and functions.
- Processing strings, numeric values and currencies.



Estimated time: 1 hour

### The variables

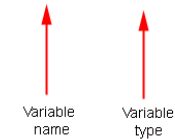
#### What is a variable

In a programming language, a variable is used to store data. These memory sections contain strings, numbers, etc.

The variables are used to perform calculations, to perform comparisons or to store information that will be used later.

#### Declaring a variable

`Price is currency`



#### Initializing a variable

`Price = 500.32`



A variable is represented by:

- a name: Name given to the variable so that it can be used by the language.
- a type: Nature of data stored in the variable (see The types of variables).
- a value: Information stored in the variable.
- a scope: Limit for using the variable in the program (see The scope of variables). The scope is mainly defined by the location where the variable is declared.

#### Declaring and assigning a variable

The variable must be declared (which means created) before it can be used.

- Example of simple declaration:

```
Price is currency
```

- **Price** represents the variable name.
- **is** is used to declare the variable. The everyday language is used in WLanguage.
- **currency** corresponds to the variable type.

- Example of multiple declaration:

```
LastName, FirstName are strings
```

- **LastName, FirstName** represent the names of variables.
- **are** is used to declare a set of variables.
- **strings** represents the type of variables.

When the variable is declared, you have the ability to assign it (or to give it a value). For example:

```
// Assign a currency variable
Price = 1256.67
// Assign a string variable
LastName = "Doe"
```

The **=** operator is used to perform this assignment.



Remark

In WLanguage, the " character (double quote) is the character used to delimit a character string. In the above example, the doubles quotes are used to assign the Doe value to the LastName variable.

The variable content can be read and handled: all you have to do is use the name given to the variable to access it.

The following example is used to read and display the content of Price variable on the screen:

```
Info(Price)
```

### The types of variables

The variable type is used to specify the kind of information that will be stored in the variable. The most common types are:

- boolean (True or False),
- string ("Doe"),
- integer (1234),
- currency (12,32),
- real (7,766666),
- etc.



Important

Use the type corresponding to the information that must be stored. Therefore, you will optimize memory and avoid calculation or process errors when using variables in WLanguage functions.

Most of these types of variables will be used in this tutorial.

See the online help regarding the relevant type for more details.



Remark

Other types are available, such as arrays, structures, dates, times, ...

Advanced variables are also available. These advanced types gather all the characteristics of the element being used in a single variable.

Advanced types can be used to handle XML documents, emails, XLS files, ...

See the online help for more details.

### The scope of variables

The variables can be declared anywhere in the code. However, according to the position of its declaration, the variable cannot be used to perform processes or calculations. We talk of variable scope.

Two types of scope are available:

- Global.
- Local.

#### Global scope

Global means that the variable has an extended visibility in the code. The variable is visible outside the location where it was declared. Several levels are available:

- Project and Set of procedures,
- Window, Mobile Window, Page, Report.

A variable declared **at project level** has the greatest visibility in the program. The variable is visible anywhere, in all program processes. However, you should not declare too many variables with this scope: indeed, the memory occupied by the variable is always reserved even if the variable is not used. Using a large number of global variables is not recommended in the program architecture. To pass variables to a process, we recommend that you use parameters (see "Parameters of a procedure" for more details).

A variable declared **at Set of Procedures level** and a variable declared at project level have the same visibility. The advantage of declaring a variable at the level of a Set is to group (or classify) the variables by theme (set) in order to make the project initialization process more readable.

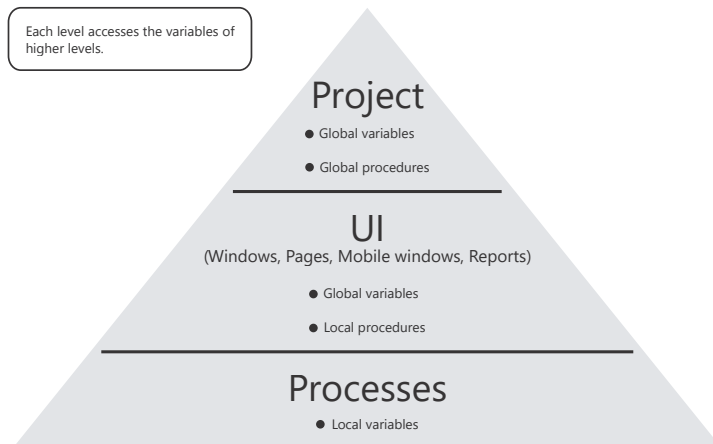
A variable declared **at Window, Mobile Window, Page or Report level** limits the scope of the variable to all the processes of the element (Window, Mobile Window, Page or Report) and its controls. This makes it possible to encapsulate and limit the uses.

#### Local scope

Local means that the variable has a limited visibility in the code. The variable is visible in the process where it was declared. This makes it possible to restrict the use of variable to the process.



## Summary scope diagram



A variable is global when it is declared:

- in the initialization code of project (or in the declaration code of set of procedures). The variable is global to the project.
- In the declaration code of global variables of window, page or report. The variable is global to the element (window, page or report) where it was declared.

In all other cases, a variable is local to the process where it is declared.

## Simple operations on the variables

Several mathematical operators can be used to perform calculations on variables:

- + to perform an addition.
- - to perform a subtraction.
- \* to perform a multiplication.
- / to perform a division.

Other operators can be used to perform calculations:

- ++ to increment from 1 (add 1 to the variable).
- -- to decrement from 1 (subtract 1 from the variable).
- += to assign by adding a value.
- -= to assign by subtracting a value.

## Examples:

```
// Declaration of variables
Cnt is int
V1 is int
Res is numeric

// Assignment
Cnt = 10
V1 = 3

// Use of operators
Cnt = Cnt + 3 // Cnt is equal to 13
Cnt ++ // Cnt is equal to 14
Cnt -= 8 // Cnt is equal to 6
Cnt = Cnt * V1 // Cnt is equal to 18
Res = Cnt / 5 // Res is equal to 3.6
```

Comparison operators are also available:

- < less than.
- > greater than.
- <= less than or equal to.
- >= greater than or equal to.
- <> different from.
- = equal to.

Other operators are available. See the online help for more details (keyword: "Operators").

## Tips

- It is very convenient to name the variables with long names (and to avoid short names such as i, j, k, ...). When reading the program again, you will be able to easily remember the variable purpose.
- To define the name of variables, all Unicode characters (including the accented characters) are accepted. Meaning improved readability! Caution: some characters are not allowed: space, =, dot, comma, ...
- It is very important to give the proper type to the variable according to its use. For example, to store several digits, you may have to:
  - use a numeric variable if this variable must be used for calculations.
  - use a string variable if this variable must be used to store digits without performing calculations (to store the social security number for example).

## The conditional IF and SWITCH statements

### The IF statement

This statement is used to run an action or another one according to the result of an expression. If the expression is checked, a process is run ; if the expression is not checked, another process can be started.

The IF statement can be used as follows:

```
IF <Expression to check> THEN
    Process to run if the expression is checked
ELSE
    Process to run otherwise
END
```

Code example: The following code selects a number at random and displays a message according to the value.

```
Tot is Currency
// Selects a number at random between 100 and 4000
Tot = Random(100, 4000)
IF Tot>2000 THEN
    Info("The amount is greater than 2000")
ELSE
    Info("The amount is less than or equal to 2000")
END
```

In this case, the expression to check corresponds to "Tot>2000".

Remark: Several code lines can be run during the process corresponding to a condition. In this case, the following syntax must be used:

```
IF <Expression to check> THEN
    Code line 1
    Code line N
ELSE
    Code line 1
    Code line N
END
```

### The SWITCH statement

This statement is used to evaluate an expression and to run a process for each possible expression value.

The SWITCH statement is used according to the syntax below:

```
SWITCH <Expression>
CASE Value 1:
    Process 1...
CASE Value 2:
    Process 2...
...
CASE Value N:
    Process N...

OTHER CASE
    Process ...
END
```

Example: The following code retrieves today's date and displays a different message according to its value. A specific message is displayed for the 1st and for the 15th of the month. In the other cases, today's date is displayed.

```
D is Date
D = Today()
SWITCH D..Day // Checks the day of the date
CASE 1: Info("We are the first day of the month")
CASE 15: Info("We are the 15th of the month")
OTHER CASE: Info("We are the: " + DateToString(D))
END
```

#### Remarks:

- If the code line "CASE 1:..." is run, the other code lines corresponding to the possible values are not run.
- Several values can be grouped in the same case. The different values are separated by a comma. For example:

```
Sub is int = 2
SWITCH Sub
    CASE 1,2: Info("Case 1 or 2")
    CASE 3: Info("Case 3")
    OTHER CASE: Info("Other case")
END
```

- Several code lines can be run during the process corresponding to a condition. In this case, the following syntax must be used:

```
SWITCH <Expression>
CASE Value 1:
    Process 1 - Code line 1...
    Process 1 - Code line 2...
CASE Value N:
    Process N - Code line 1...
    Process N - Code line 2...
END
```

## The loops

The loop statements are used to run a process in a recurring way. A specific loop statement is used according to the number of occurrences. Several statements can be used to perform loops:

- FOR...
- LOOP...
- WHILE...

### The FOR statement

The FOR statement is used when the number of occurrences to process is known. This statement is used to manage the number of occurrences via a variable in which the passages performed in the loop will be counted.

The syntax of FOR statement is as follows:

```
FOR Subscript = Start Value TO End Value
    Process to run
END
```

For example, the following code runs the process 2000 times:

```
FOR Cnt = 1 TO 2000
    // Process to run
END
```

Remark: An increment step of subscript can be defined via the STEP keyword. For example, the following code runs the process 2000 times and the Cnt variable decreases by 10:

```
FOR Cnt = 2000 TO 1 STEP -10
    // Process to run
END
```

### The LOOP statement

The LOOP statement is used to perform loops when the number of occurrences to process is unknown. In this case, a test must be used to exit from the loop.

The syntax of LOOP statement is as follows:

```
LOOP
    Process to run
    IF <Expression> THEN BREAK
END
```

For example:

```
Counter is int
Counter = 10
LOOP
    // Process to run
    Counter = Counter - 1
    IF Counter = 0 THEN BREAK
END
```



Tip

The LOOP statement and the FOR statement can have the same behavior: all you have to do is use the syntax with exit according to the number of iterations:

```
LOOP (<Number of iterations>)
...
END
```

Example:

```
LOOP(10)
    // Process to run
END
```

### The WHILE statement

The WHILE statement and the LOOP statement operate according to the same principle. The difference is that the test of exit condition is performed BEFORE running the loop code. This test is used to compare a variable. This variable starts from a start value and it is modified in the loop until it reaches the value that triggers the exit from the loop.

The syntax of WHILE statement is as follows:

```
<Initialize the variable to its start value>
WHILE <Compare the variable to its end value>
    Process to run
    <Modify the variable>
END
```

For example:

```
Counter is int
Counter = 0
WHILE Counter<10
    // Process to run
    Counter = Counter + 1
END
```

## The procedures

A procedure is used to associate an identifier with a code section in order to re-use it.

In this lesson, we are going to present the different types of procedures available in WLanguage, their creation mode, how to call them, pass parameters and retrieve a result.

### Types of procedures

Three types of procedures are available:

- **Global procedure:** can be used in all project processes (declared in a set of procedures).
- **Local procedure** in a Window, Page or Mobile Window: can be used in all processes that depend on the object in which this procedure was declared.
- **Internal procedure** in a process: can only be used in the process where it was declared.



Remark

#### Scope of procedures

The procedures comply with the scope rules presented for the variables (see "The scope of variables").

### Creating and calling a procedure

To create a global procedure, you must:

1. Create (if necessary) a set of procedures (via the "Project explorer" pane, "Procedures" folder). Give a name to the set of procedures.
2. Create a global procedure in the set of procedures (via the "Project explorer" pane, "Procedures, Set name" folder). Give a name to the procedure.
3. Type the code of global procedure. The procedure code has the following format:

```
PROCEDURE <Name of global procedure>()
```

To create a local procedure, you must:

1. Select the element associated with the procedure (window, page, etc).
2. Create a local procedure (via the "Project explorer" pane, expand the element name, "Local procedures" folder).
3. Give a name to the procedure.
4. Type the code of local procedure. The procedure code has the following format:

```
PROCEDURE <Name of local procedure>()
```

To create an internal procedure, type the following code in the requested process:

```
INTERNAL PROCEDURE <Procedure name>()
<Code of internal procedure>
END
```

To call a procedure, use the procedure name (with the possible parameters that will be passed to it).

```
<Procedure name>(<Parameter 1>, ..., <Parameter N>)
```

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

### Procedure parameters

#### What is a parameter?

A parameter is a value sent to a procedure during the call to the procedure.

The following example is used to call the **Multiply10** procedure and to pass in parameter the value that will be handled in the procedure:

```
Multiply10(50)
```

You have the ability to pass from 0 to several values in parameter to a procedure. These values can have any type (as with the variables).

The parameter is specified in the procedure declaration in the format of a variable. For example, for the **Multiply10** procedure, the procedure code is:

```
PROCEDURE Multiply10(P)
P=P*10
```

P is the parameter expected by the procedure.



Remark

To specify the parameter role in the procedure, you have the ability to typecast the parameter in the procedure declaration.

For example, to use numeric values only, you have the ability to declare:

```
PROCEDURE Multiply10(P is numeric)
```

In the following example, the **Multiplication** procedure expects two Integer parameters and returns the multiplication result.

The procedure code is as follows:

```
PROCEDURE Multiplication(Nb1 is int, Nb2 is int)
MyResult is int
MyResult = Nb1 * Nb2
RESULT MyResult
```

The code used to call the procedure is as follows:

```
res is int
res = Multiplication(10, 50)
// Res is equal to 500
```

### How to use the parameters?

By default, **passing parameters in WLanguage is performed by reference** (or by address). The parameter in the procedure represents (references) the variable passed during the call.

Therefore, when a procedure statement modifies the parameter value, the value of the variable corresponding to this parameter is modified.

Example:

- The procedure code is as follows:

```
PROCEDURE Test_address(P1)
P1 = P1 * 2
```

- The code used to call the procedure is as follows:

```
T is int
T = 12 // T is equal to 12 before the call
Test_address(T)
// T is equal to 24 after the call
```

To avoid modifying the value of the variable corresponding to the parameter, the **parameters must be passed by value**. Passing parameters by value allows you to handle a copy of parameter value. If the procedure code modifies the variable value, the value of the variable corresponding to the parameter is not modified.

To force a parameter to be passed by value to a procedure, the **LOCAL** keyword must be used in front of the parameter name in the procedure declaration. This keyword indicates that the following parameter will not be modified by the procedure.

Example:

- The procedure code is as follows:

```
PROCEDURE Test_value(LOCAL P1)
// Local indicates that the parameter will be passed by value
P1 = P1 * 2
```

- The code used to call the procedure is as follows:

```
T is int
T = 12 // T is equal to 12
Test_value(T)
// T does not change
```



Remark

In the same procedure, some parameters can be passed by reference while other parameters can be passed by value. All you have to do is use the **LOCAL** keyword in front of each parameter passed by value.

### Mandatory or optional parameters?

The parameters received in the procedure can be mandatory or optional parameters. A mandatory parameter must be filled during the call to the procedure while an optional parameter can be omitted: in this case, it will take the default value defined when declaring the procedure parameters.



Remark

When declaring a procedure, the optional parameters are the last parameters (they are always specified after all mandatory parameters).

In the following example, the **Multiplication** procedure is using an optional parameter, Nb2. This optional parameter is indicated after the mandatory parameters, by specifying its default value. In this example, the default value of optional parameter is set to 10.

```
PROCEDURE Multiplication(Nb1 is int, Nb2 is int=10)
MyResult is int
MyResult = Nb1 * Nb2
RESULT MyResult
```

The code used to call the procedure is as follows:

```
res is int
res = Multiplication(6)
// Res is equal to 60
```

In this example, the second parameter was not specified. Therefore, its default value will be used.

### Procedure result

The procedures can return one or more results. The result can be typecasted. The **RESULT** keyword must be used to return a value.

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

**PART 5**

Android and iOS:  
Developing an  
application



## LESSON 5.1. OVERVIEW

**This lesson will teach you the following concepts**

---

- What is a multi-platform project?
- Which project is used for this part?
- How to enable a platform?



Estimated time: 10 mn

## Overview

Two separate projects have been created in the previous parts: an iOS project and an Android project.

This new part will allow you to develop an Android and/or iOS application that is using a HFSQL Classic database.



Remark

This part can be followed both by the developers of Android applications and by the developers of iOS applications. The operations performed are identical, no matter whether the project is developed for Android only or for iOS only.

We are going to develop a multi-platform project that can be used both on an Android platform and on an iOS platform.

Most of the time, when developing an application for Mobile, this application must operate both on Android and on iOS.

With WINDEV Mobile, there is no need to develop two different projects, to maintain them and to make them evolve in parallel. All you have to do is create a single project associated with several platforms, Android and iOS for example: it is a **multi-platform** project.

Each project element (windows, queries, ...) can be associated with one or more platforms.

If a window is common to several platforms, the specific features of the platform can be managed via the **layouts**. The layout is used to define several views of a window in the same project without duplicating this window.

When the project is developed, all you have to do is select the requested configuration to create and deploy the application on the selected platform.

## The sample project used

In this lesson, we are going to develop an Android and iOS application that is using a HFSQL Classic database. The creation of projects was already presented in the previous part therefore we will be working on an existing project, containing the database and the data used by the application.



Remark

In iOS and Android, only the HFSQL database is accessible in native mode. Both the Classic mode and the Client/Server mode are available.

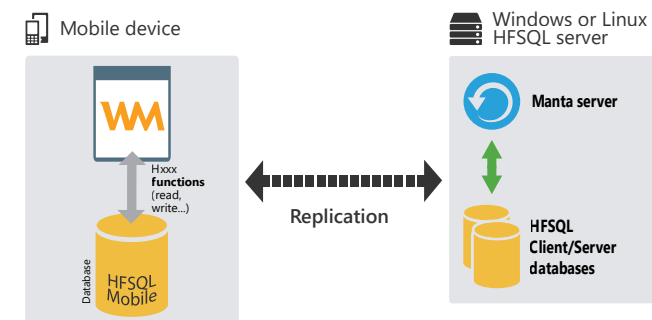
### HFSQL Classic

In HFSQL Classic mode, the data files are stored on the device (iPhone, iPad, Android phones or tablets).

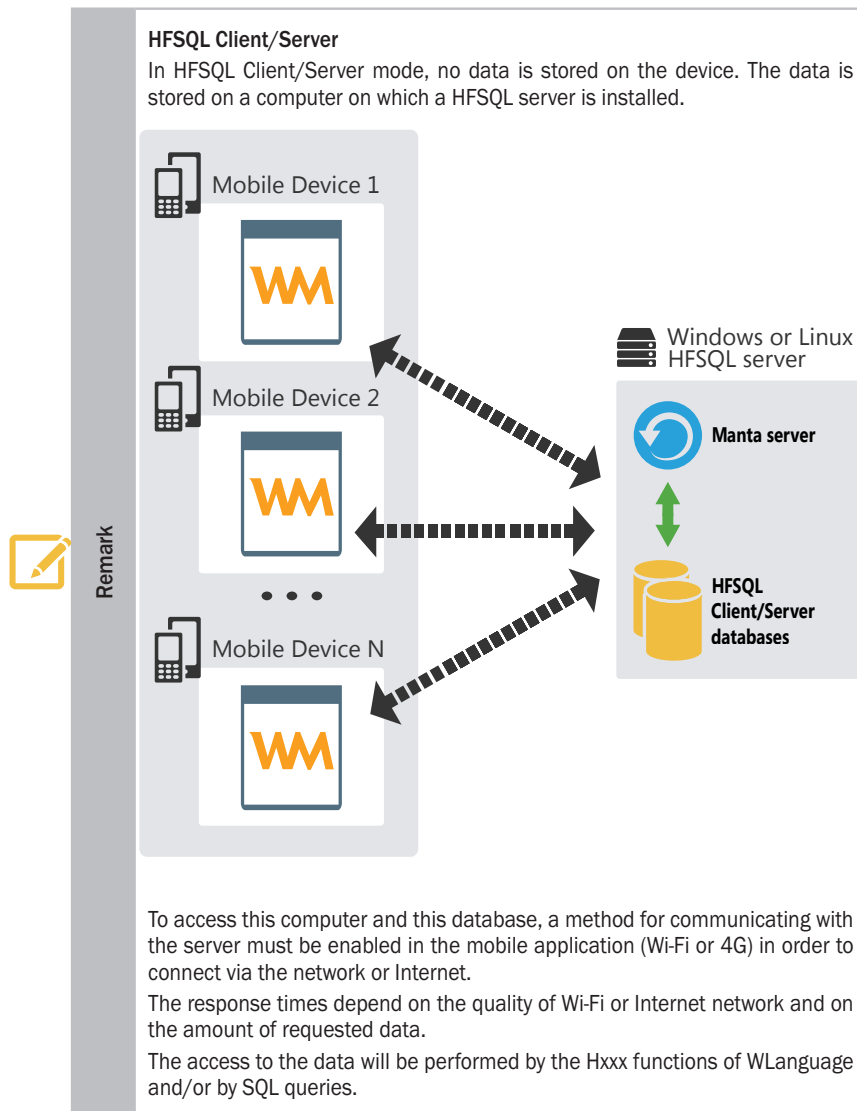
In this case, the application is stand-alone. No Wi-Fi or 4G connection is required.

The data is stored in the device memory. The maximum storage size depends on the amount of memory on the device.

If the data typed on the mobile device must be synchronized with a database found on a server (HFSQL or other), the replication must be implemented (see the online help).



In this part, we will develop an application that is using a HFSQL Classic database.



### Opening project

- Start WINDEV Mobile 25 (if necessary).
- Display the home page if necessary (Ctrl + <).
- Open the "WM Managing Products" project.  
To do so, in the home page, click "Tutorial" and select "iOS/Android application (Exercise)".



Answer

A corrected project is available. This project contains the different windows created in this lesson. To open the corrected project, in the home page, click "Tutorial" and select "iOS/Android application (with windows)".

### Choosing the platform

If you want to develop the application for the **Android** platform, go to lesson 5.2 - Developing the application.

If you want to follow this part for **iOS** only, you must add the iOS platform (see next paragraph "Enabling the iOS platform").



Remark

All operations performed in this part can be performed on a project associated with an Android platform, on a project associated with an iOS platform or on a multi-platform project.

For information, the different images illustrating the lessons of part 5 have been created with the Android configuration. Slight differences may appear if you are using the iOS configuration.

### Enabling the iOS platform

- To enable the iOS platform, simply create a project configuration:
  1. On the "Project" pane, in the "Project configuration" group, expand "New configuration" and select "iPhone/iPad application".



Remark

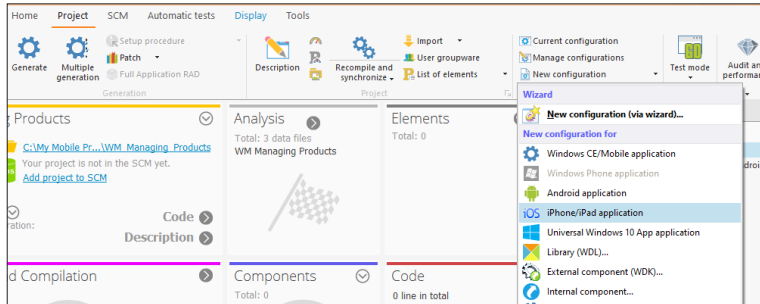
**The project configurations** are used to define the different types of generations supported by the project: Android, iOS, component, ...

The project elements can be:

- common to several configurations (a window used in iOS and Android for example).
- specific to a configuration (a class used by a component for example).

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

See the online help for more details.



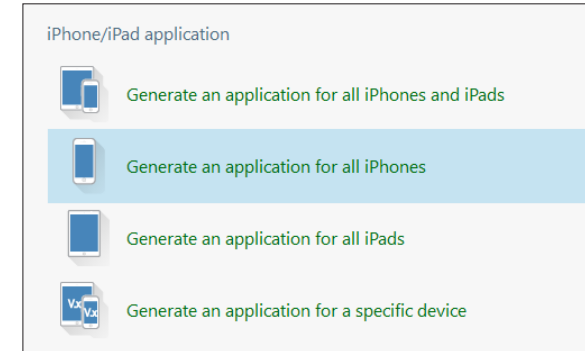
2. The wizard for creating a project configuration starts. The wizard proposes to create an iOS platform. Go to the next step.
3. You can enter information regarding the platform. Keep the default options and go to the next step.
4. The wizard proposes to choose the type of devices affected by the project:
  - Generate an application for all iPhones and iPads.
  - Generate an application for all iPhones.
  - Generate an application for all iPads.
  - Generate an application for a specific device.



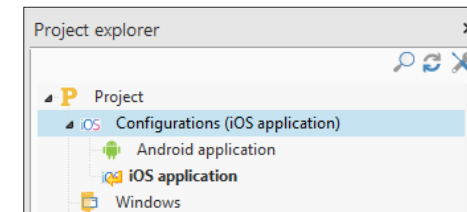
Remark

If the application is intended to operate on several iOS devices (phones with different sizes or with different resolutions for example), we advise you to use one of the following options: "Generate an application for all iPhones and iPads", "Generate an application for all iPhones" or "Generate an application for all iPads".

In this case, WINDEV Mobile proposes the smallest resolution to create the application windows. Using anchors (see "Managing anchors in the Product form", page 127) will allow the application to operate on all devices.



5. For this example, select "Generate an application for all iPhones".
  6. Go to the next step and validate the wizard. The iOS configuration is automatically created.
- You can check the new project configuration in the "Project explorer" pane:
1. Expand "Configurations (iOS application)" at the top of the "Project explorer" pane.
  2. The two configurations are displayed:



3. To select a specific configuration, simply double-click the name of the requested configuration.
4. The iOS configuration that was just created is automatically selected.

Let's now start developing the application.

- For the rest of this tutorial, we recommend that you activate the Android configuration.
1. Expand "Configurations (iOS application)" at the top of the "Project explorer" pane.
  2. Double-click on the configuration name "Android application".
  3. The Android configuration is automatically selected.



Remark

For information, the different images illustrating the lessons of part 5 have been created with the Android configuration. Slight differences may appear if you are using the iOS configuration (especially in the Action Bar).

## LESSON 5.2. DEVELOPING THE APPLICATION

### This lesson will teach you the following concepts


- Creating a window containing a loop.
- Specific controls: Looper control, Multiline Zone control, Map control, ...
- Handling the database.
- Features specific to the device used (GPS, Photo, ...).

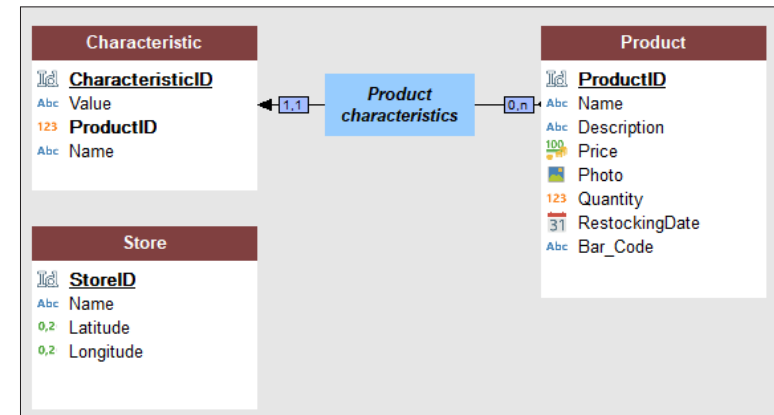


Estimated time: 30 mn

### Project analysis

Let's take a look at our initial project. This project contains no window. It only contains the analysis describing the HFSQL Classic data files that will be used. The corresponding data files are supplied with data in order to run the different tests.

- To view the analysis associated with the project:
  1. Click  among the quick access buttons of WINDEV Mobile menu.
  2. The data model editor is displayed.



3. This analysis includes 3 data files:
  - A "Product" data file, which contains the product description: name, price, quantity, ...
  - A "Characteristic" data file, which contains the different product characteristics. For example, if the product is a tee-shirt, its characteristics will correspond to the size, the color, ... The "Characteristic" data file is therefore linked to the "Product" data file.
  - A "Store" data file, which contains the GPS coordinates of each store.
4. Close the data model editor (click the cross at the top right of editor).

### Displaying the list of products

We are going to create a window used to list the different products. These products will be displayed in a "Looper" control.

This window will be created with the window creation wizard.




Remark

The window creation wizard offers several preset windows. These windows propose modern interfaces for your applications. Most of these windows can be generated from your data, by including the WLanguage code required for them to operate. They can be used straightaway.

### Creating the window

► To create the window used to list the products:

1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Window" then "Window".
2. In the wizard, select the "Standard" tab. In the "Phone" area, choose "Looper" and validate.



#### Remark

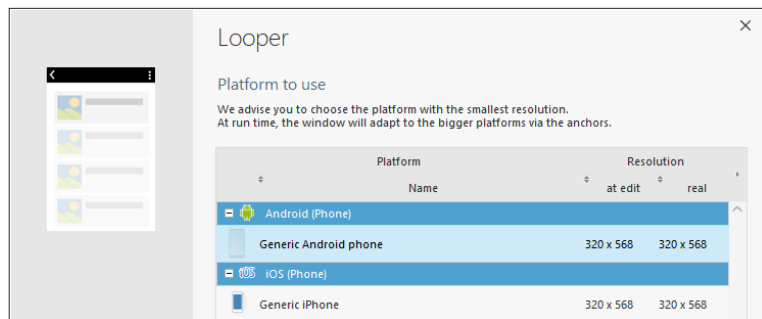
If you are using an iOS configuration, the window for creating a new window contains additional options, specific to iOS.

A "Looper" window can be created for an iOS platform and for an Android platform.

Reminder: The images in part 5 of this tutorial use the Android configuration. Slight differences may appear if you are using the iOS configuration.

3. The window creation wizard starts.

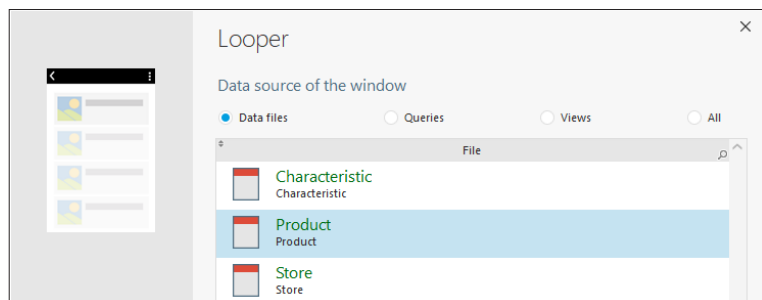
4. If the **iOS platform was added**, several platforms are proposed. We advise you to choose the platform with the smallest resolution.



Go to the next wizard step.

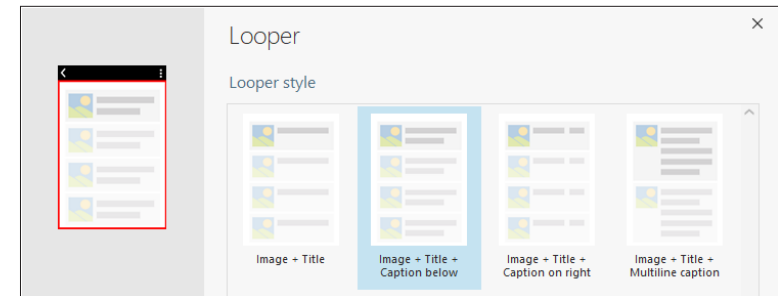
5. The wizard proposes to choose the data source associated with the window. In our case, it will be the "Product" data file:

- The "Data files" option is selected by default.
- Select the Product data file.



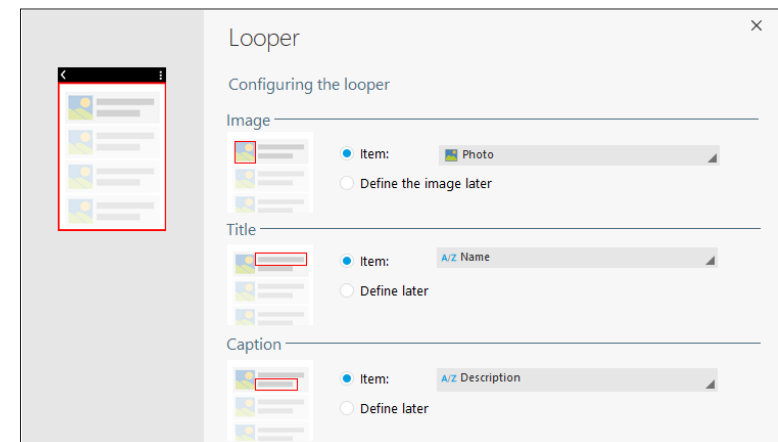
6. Go to the next step.

7. Select the style of the Looper control: "Image + Title + Caption below". This template is used to get an interface containing the image of product, its name and its description.



8. Go to the next step.

9. The wizard automatically proposes the file items corresponding to the different controls of the generated looper.

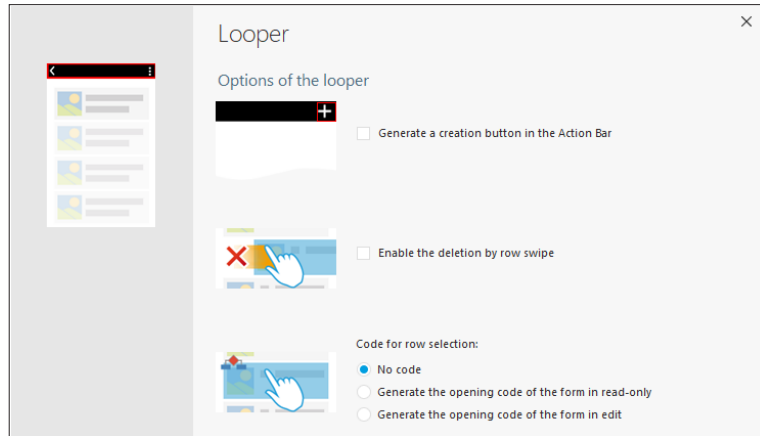


10. Keep the proposed items and go to the next step.

11. Keep the suggested sort on the "ProductID" item. Go to the next step.



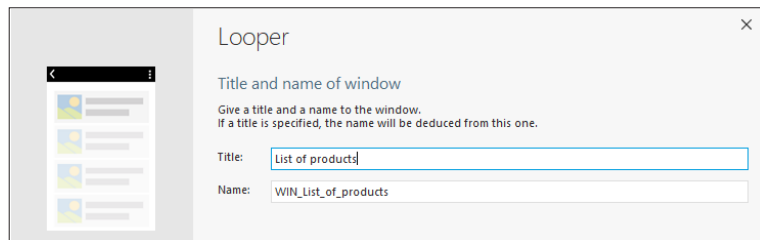
12. The wizard proposes several options for generating the Loopier window:



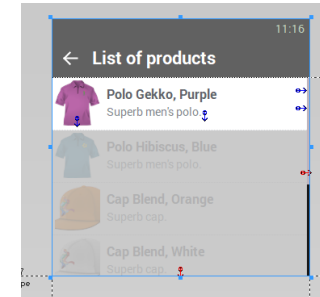
- **Generate a creation button in the Action Bar:** If this option is selected, the wizard proposes to generate an editable form window.
- **Enable the deletion by row swipe:** If this option is selected, the user will be able to delete an element of the Loopier control by swiping the corresponding row.
- **Code for row selection:** If "Generate the opening code of the form in read-only" (or "Generate the opening code of the form in edit") is selected, the wizard proposes to generate a form window in read-only or in edit mode.

13. In our example, keep the default options. Go to the next step.

14. Give a title and a name to the generated window. Type the window title: "List of products". The window name is automatically filled.




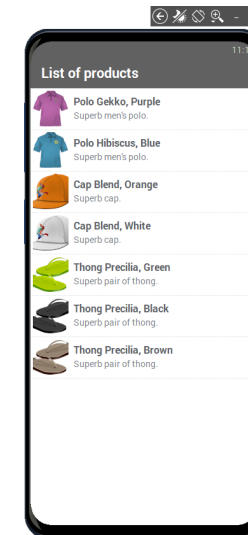
15. Validate the wizard. The window is automatically created, displayed in the editor and saved.



#### Remark

Data automatically appears in the window displayed in the editor. This concept is called "Live Data": you see the data of your data files in real time! This feature is very useful to adapt the size of controls to their content.

- We are going to run a first test in the simulator to see the result. Click  among the quick access buttons (or press F9).



- Close the simulator to go back to the window editor.



Remark

When a project is associated with several platforms, the test of current window is run with the current platform. The shell used for the simulator is also adapted to the current platform.


To perform the test on another platform, simply enable this platform by double-clicking its name in the "Project explorer" pane.

Remark: In this part, the images representing the tests correspond to an Android shell.

## Creating the form window

We are now going to create a new window used to display the product form. Then, this window will be started from the list of products to display the details of selected product.

### Creating the window

- To create the form window:
  1. Create a new blank window.
    - Click  among the quick access buttons.
    - The window for creating a new element is displayed: click "Window" then "Window".
    - In the wizard, select the "Standard" tab, choose "Blank" and validate.
  2. The window for saving an element is displayed. Specify the window title: "Product form". Its name is automatically proposed: "WIN\_Product\_form".

Validate.

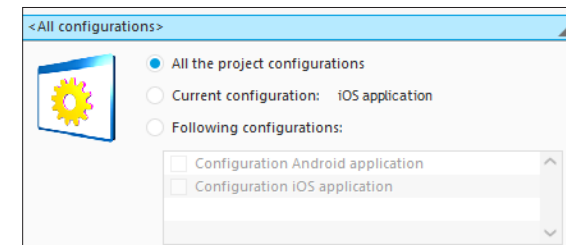
3. The window is added to the project, for all configurations.



Remark

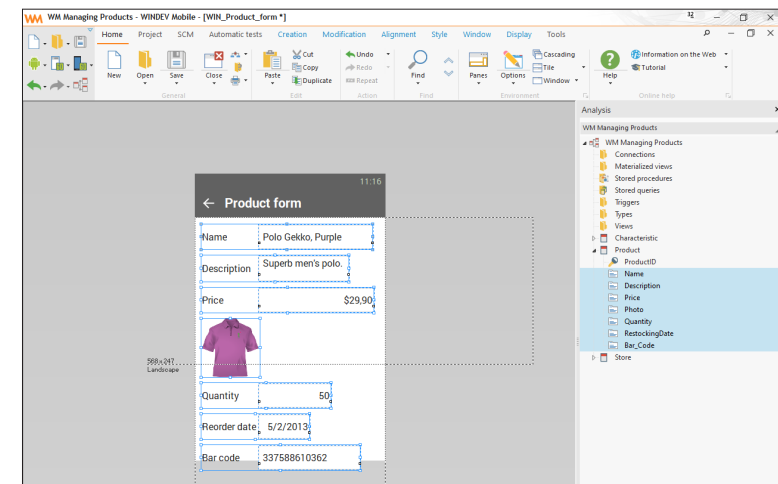
If several platforms are defined in the project, the window for saving an element proposes to associate the new element to all platforms.

To associate the element to a single platform, all you have to do is expand "Configurations" in the save window and choose the requested platform:

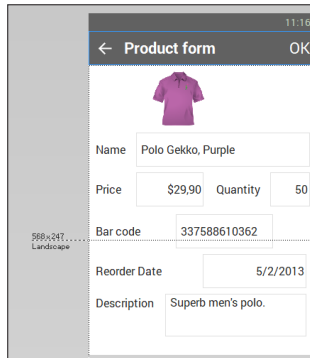


### Creating controls

- To create the different edit controls used to display information about the product:
  1. Display the "Analysis" pane if necessary: on the "Home" pane, in the "Environment" group, expand "Panels" and select "Analysis". The different data files described in the "WM Managing Products" analysis appear in the pane.
  2. With the mouse, select the items of the "Product" data file displayed in the pane (except for the "ProductID" item).
  3. Drag and Drop these items to the window that was just created.

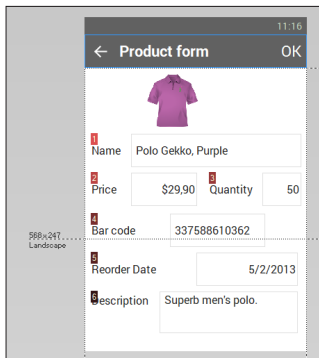


4. Resize the controls ("Name", "Bar code", "Reorder Date" and "Description") so that they become visible in the window.
5. Reorganize the controls in the window. Respect the following order: "Photo", "Name", "Price", "Quantity", "Bar code", "Reorder Date", "Description".



6. We are going to view the tab order in the window:

- Press F5. The numbers that are displayed represent the tab order in the window.



Press F5 again to hide the numbers.

- The tab order is automatically adapted to the order of controls in the window.

7. Save the window.

- ▶ Run the test of "WIN\_Product\_form" window (  among the quick access buttons).

1. The window is displayed with empty controls.
2. Close the simulator.

- ▶ To display the product data:

1. Display the WLanguage events associated with the window:
  - Perform a right mouse click in the area beside the window.
  - Select "Code" from the popup menu that is displayed.
  - The code editor appears.

2. Enter the following WLanguage code in the event "End of initialization of WIN\_Product\_form":

```
FileToScreen()
```

**FileToScreen** is used to display in the controls the data found in the data file, for the current record. In our case, the current record will be the record selected in the Looper control of "WIN\_List\_of\_products". This Looper control is linked to the Product data file.

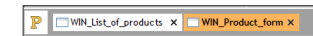
3. Close the code window.
4. Save the window.

### Displaying the form from the list of products

Let's now see how to display the form of selected product in the list of products.

- ▶ Perform the following operations:

1. Position on the "List of products" window: click the "WIN\_List\_of\_products" button in the open documents bar:



2. Select the Looper control.  
Caution: make sure you select the Looper control and not one of its controls.
3. Display the popup menu of Looper control (right click) and select "Code".
4. In the code window that appears, write the following WLanguage code in the event "Selecting a row of...":



```
OpenMobileWindow(WIN_Product_form)
```



#### Remark

The assisted code input is going to help you: as soon as you type the opening bracket "(", a drop-down list proposes the name of all existing windows found in the project. All you have to do is select the window with the keyboard or with the mouse.

If the window name is not displayed in the list, it means that this window was not saved beforehand.

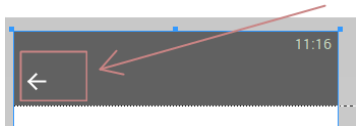
5. Save the modifications by clicking  among the quick access buttons.
  6. Close the code window (click X at the top right corner of code editor).
- ▶ Run the test of "WIN\_List\_of\_products" window again in the simulator (  among the quick access buttons).
    - In the list of products, click one of the products with the mouse.
    - The detailed product window is displayed.
  - ▶ Close the simulator.

## Managing the product creation and modification

We are now going to modify our two windows in order to manage the product addition and modification.

### Modifying the product form

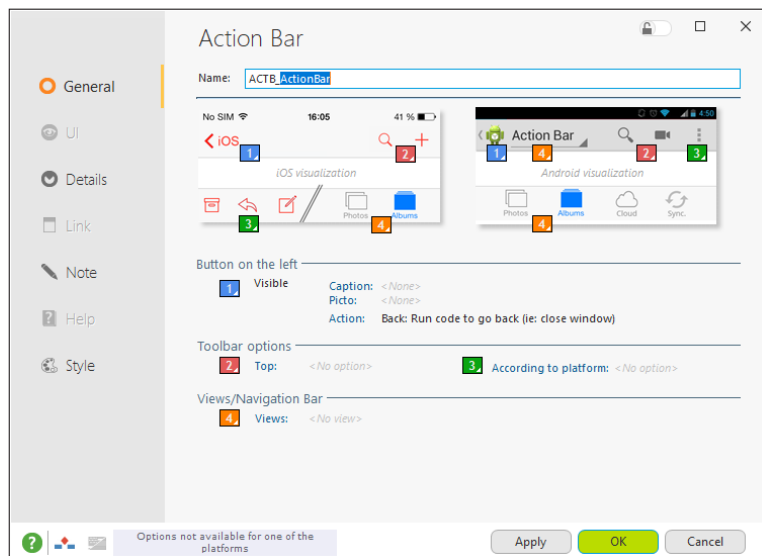
When creating the form window, an Action Bar was automatically created. This Action Bar contains a left button used to cancel the current input and to go back to the previous window. In our case, this button will be used to go back to the list of products.



There is no modification to perform.

We are going to add a validation button into the Action Bar of "WIN\_Product\_form" window in order to manage the validation of modifications.

- ▶ First of all, display (if necessary) the "WIN\_Product\_form" window in the editor: click the corresponding button in the open documents bar.
- ▶ To add a validation option into the Action Bar of window:
  1. Select the Action Bar (at the top of window).
  2. Open the Action Bar description window: right-click and select "Description" from the popup menu.



### Remark

The description window of Action Bar is adapted to the platforms used in the project. Indeed, the Action Bar used in an Android application differs from the Action Bar used in an iOS application.

**If your project is using an Android configuration**, only the options corresponding to the Action Bar for Android are displayed.

Similarly, **for an iOS configuration**, only the options corresponding to the Action Bar for iOS are displayed.

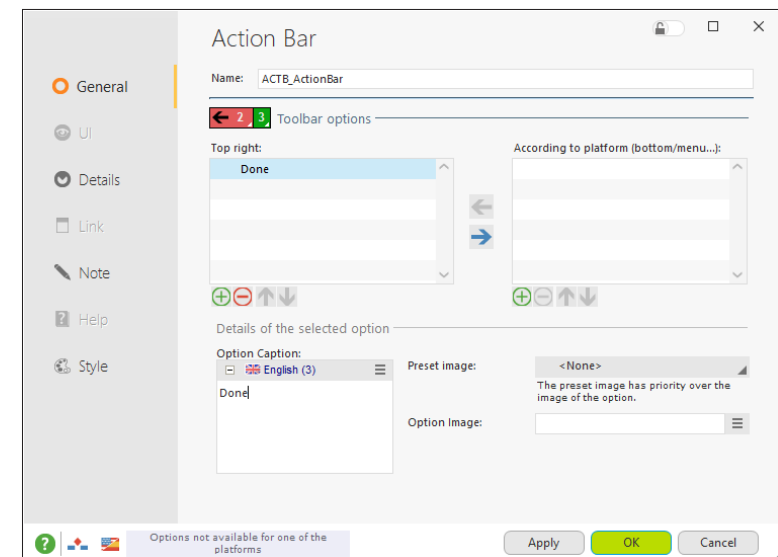
**If your project is using both an Android configuration and an iOS configuration**, the window displays a preview of the Action Bar for the two platforms.

Numbers allow you to access the options to configure.

We are going to present the options required by our example.

See the online help for more details.

3. Click on number 2 (any number 2 in the window). The interface for typing an option in the toolbar is displayed.
4. Click on "+" (below the "Top right" area) to add an option. A new default option is created at the top right.
5. Modify the characteristics of this option:
  - In the "Option caption" area, type "Done".

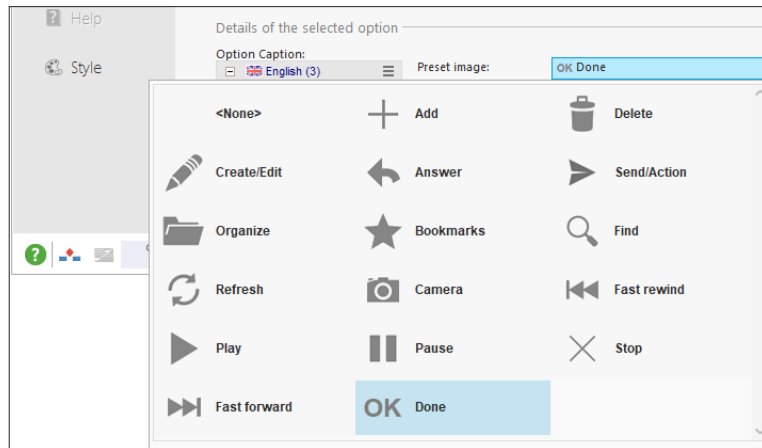


### Remark

At run time, the option caption appears in the Action Bar:

- If no image is associated with the option.
- If the option is transferred into menu at the bottom.

- In the "Preset image" area, expand the list and select "Done".



- Validate the description window.

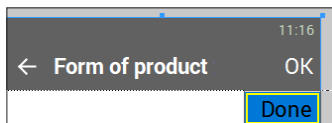


Remark

To go back to the interface for describing the Action Bar, all you have to do is click the  button.

6. The WLanguage code of this option will be used to save the modifications made in the "WIN\_Product\_form" window. To write this code:

- Select (if necessary) the Action Bar control of window.
- Click on "OK".
- A drop-down menu with the "Done" option is displayed.



Remark

This drop-down menu is visible in edit to type the WLanguage code associated with the option. This drop-down menu will not be displayed at run time.


- Right-click the option.
- Select "Code" from the popup menu that is displayed.

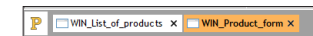
- Write the following WLanguage code in the event "Selecting the menu":

```
ScreenToFile ()
HSave (Product)
Close ()
```

Let's take a look at these lines of code:

- **ScreenToFile** is used to initialize the items with the values of linked controls, for the current record.
- **HSave** is used to update the file data for the current record.

7. Save the modifications by clicking  among the quick access buttons.
  8. Close the code window (click X at the top right corner of code editor).
- When closing the product form, the content of the product list found in the "WIN\_List\_of\_products" window must be refreshed to take into account the modifications performed in the form. To do so, use the event "Closing a child window" of "WIN\_List\_of\_products".
1. Click the "WIN\_List\_of\_products" button in the open documents bar:




2. Right-click the window background and select "Code" from the popup menu. The following WLanguage code is automatically displayed in the event "Closing a child window":

```
LooperDisplay (LOOP_Product, taCurrentSelection)
```

Let's take a look at this WLanguage code:

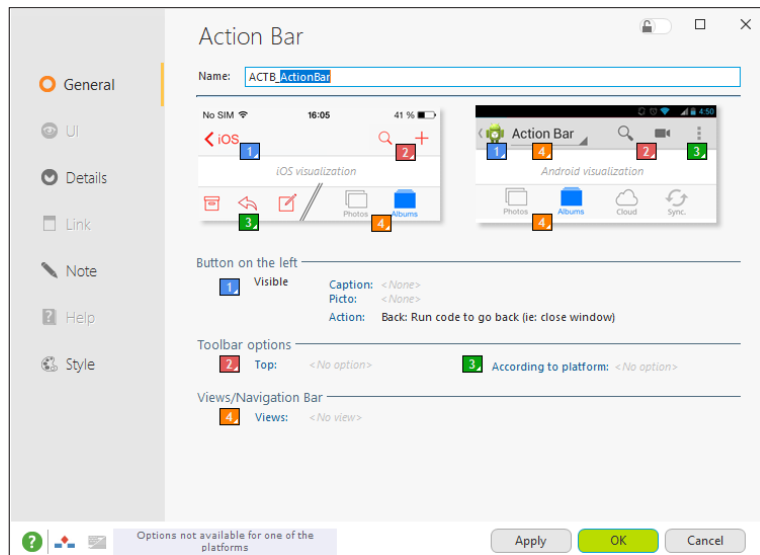
- The event "Closing a child window" is run whenever a child window of the current window is closed. In our case, it is run when the "WIN\_Product\_form" window is closed.
  - **LooperDisplay** is used to update the data found in the Looper control of "WIN\_List\_of\_products" window. The **taCurrentSelection** constant is used to update data from the selection bar.
- This WLanguage code was automatically generated when the wizard created the window.

3. Close the code window (click X at the top right corner of code editor).
- Run the test of "WIN\_List\_of\_products" window in the simulator (  among the quick access buttons).
- In the list of products, click one of the products with the mouse: for example, the "Polo Hibiscus Blue" product.
  - The detailed window of product is displayed. Modify the product name and type "Polo Hibiscus Light blue" and click the "OK" button.
  - When going back to the list of products, you will notice that the name of this article was updated.
- Close the simulator. The WINDEV Mobile editor is displayed.

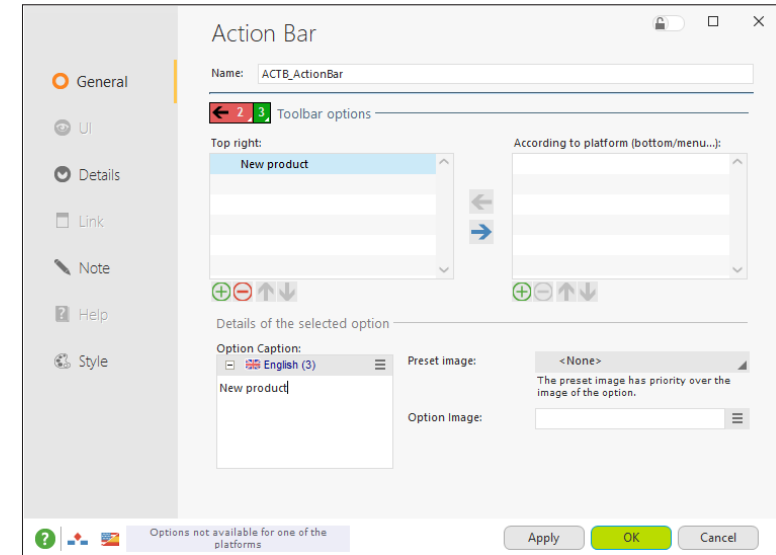
## Creating a new product

The principle for creating a product is as follows:

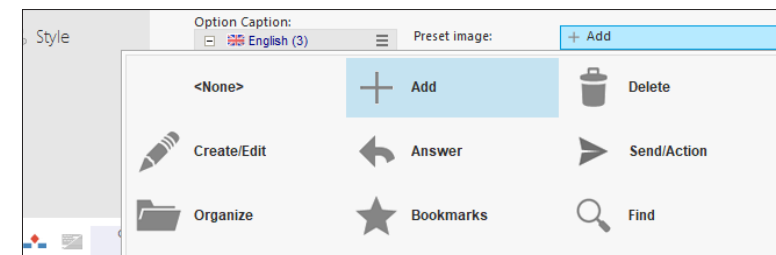
- In the window for the list of products, we are going to add option into the Action Bar of window in order to open the "Product form" window.
  - Then, we will modify the WLanguage code of "Product form" window to manage the addition into the Product data file.
- To add an option into the Action Bar of window:
1. First of all, display (if necessary) the "WIN\_List\_of\_products" window in the editor: click the corresponding button in the open documents bar.
  2. Select the Action Bar (at the top of window).
  3. Open the Action Bar description window: right-click and select "Description" from the popup menu.



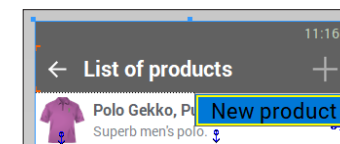
4. Click on number 2 (any number 2 in the window). The interface for typing an option in the toolbar is displayed.
5. Click on "+" (below the "Top right" area) to add an option. A new default option is created at the top right. Modify the characteristics of this option:
  - In the "Option caption" area, type "New product".



- In the "Preset image" area, expand the list and select "Add".



- Validate the description window.
6. The WLanguage code of this option is used to open the "Product form" window and to reset its controls. To enter this WLanguage code:
    - Select (if necessary) the Action Bar control of window.
    - Click the "+" button.
    - A drop-down menu with the "New product" option is displayed.






- Right-click the option.
- Select "Code" from the popup menu that is displayed.
- Write the following WLanguage code in the event "Selecting the menu":

```
HReset (Product)
OpenMobileWindow (WIN_Product_form)
```

**HReset** initializes the item variables in the "Product" data file with the default values to manage a new record.


7. Save the modifications by clicking  among the quick access buttons.
8. Close the code window (click X at the top right corner of code editor).

► Let's now check the management of record addition into the window of product form.

1. Open the "WIN\_Product\_form" window in the editor: click the corresponding button in the open documents bar.
2. Display the code of validation option in the Action Bar:
  - Select the Action Bar.
  - Click on "OK".
  - A drop-down menu with the "Done" option is displayed.
  - Select "Code" from the popup menu of "Done" option (right mouse click).
  - The WLanguage code of the "Click" event does not change:

```
ScreenToFile ()
HSave (Product)
Close ()
```

Let's look at this code again:

- By default, **HSave** is equivalent to **HModify** (current record modified in the data file).
  - If **Hreset** has been previously called (as is the case when adding a new product), **HSave** adds a new record in the data file (equivalent to **HAdd**).
3. Close the code window (click X at the top right corner of code editor).
- Open "WIN\_List\_of\_products" in the window editor and test it in the simulator (  among the quick access buttons).
- Click the "+" button found in the Action Bar.
  - Type a new product.
  - Validate. The new product is displayed in the list of products.
  - Close the simulator.



#### Remark

If your project is using several platforms, a GO must be run for each platform. The differences regarding the display and the operating modes will appear in the simulator.

## Using the camera and displaying an image



#### Caution!

This paragraph requires a device equipped with a camera. If this is not the case, go to the next paragraph directly.  
Furthermore, this paragraph requires the application to be installed on the device because it is using hardware resources that are not accessible in Simulation mode.

### Overview


We are going to manage the product photo by using the device camera.

To do so, we are going to:


- create a Button control to launch the camera. The photo will be retrieved in the format of an image in memory and displayed in the Image control of product.
- create a Button control to select a photo from the mobile device's album.

### Creating the button for taking photos

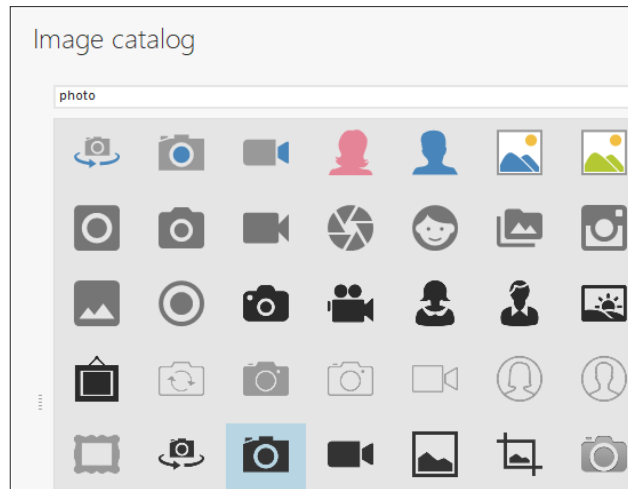
► To create the Button control used to take photos:

1. Open the "WIN\_Product\_form" window in the editor.
2. Add a Button control in the window:
  - On the "Creation" pane, in the "Usual controls" group, click  : the control shape appears under the mouse.
  - Then click on the top right of the Image control to create the Button control.

► We will modify the Button control to associate it with an image representing a camera:

1. Select the control and open its popup menu (right click).
2. Select "Description". The description window of control is displayed.
3. In the "General" tab, click the  button on the right of "Image" control. Select "Catalog" from the popup menu that is displayed.
4. The image catalog of WINDEV Mobile is opened. This catalog contains hundreds of images in different fields, formats and sizes.

5. Enter "photo" in the search box at the top and validate. Several images are displayed:



6. For example, double-click  to select it.

7. In the window that appears, you can select the image size, color, orientation, format and name.

8. Keep all default options and specify the image name ("Camera").

9. Validate the window.

10. The path of the image is displayed in the Button control description.

11. Give a name to the control: "BTN\_Camera".

12. Clear the control caption.

13. Validate the description window.

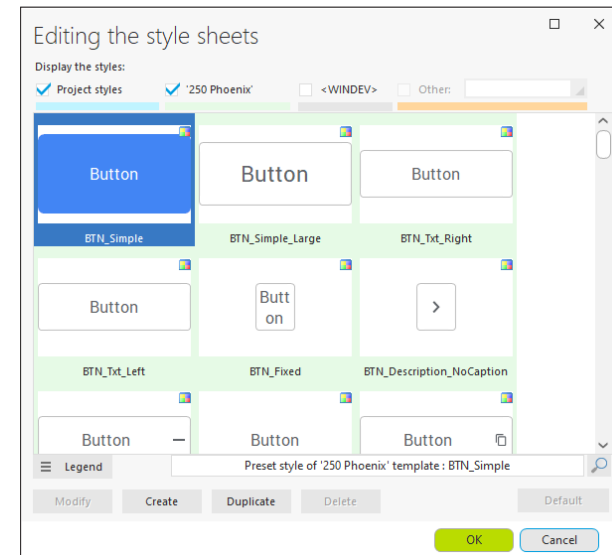
14. In the editor, reduce the control size.

- This Button control uses the default style from the "Phoenix" skin template associated with the project: the control background is blue. In our case, the blue background is not necessary. We will modify the style associated with the control:

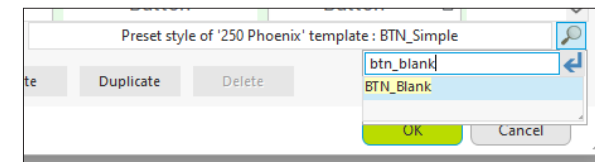
1. If necessary, select the Button control you have just created.

2. Open the popup menu and select "Choose a style".

3. The window that appears lists all styles that can be used for the Button controls in the Phoenix skin template.



4. Click the magnifier and enter "BTN\_Blank".



This name corresponds to a style without background color. This is the style we are going to use.

5. Press Enter to validate. The "BTN\_Blank" style is automatically selected.

6. Validate the window for style selection.

7. The chosen style is automatically applied to the Button control.

## Taking photos

We will now enter the WLanguage code to take a photo and to display it in the Image control of the product form.

► To take photos:

1. Select "Code" from the popup menu of the Button control (right click).
2. Write the following WLanguage code in the "Click" event:

```
// Local variable
sPhoto is string
// Start the camera
sPhoto = VideoStartApp(viPictureCapture)
IF sPhoto <> "" THEN
    IMG_Photo = sPhoto
END
```

In this WLanguage code, **VideoStartApp** is used to launch the native camera application on the device in order to record video or to take a photo.

3. Save the modifications by clicking  among the quick access buttons.
4. Close the code window (click X at the top right corner of code editor).

## Selecting a photo in the photo album

We are going to add a Button control used to select a photo in the device album and to associate it with the product.

- To create the Button control for selecting the photo, we are going to "Copy - Paste" the photo capture Button control that was just created:
1. Select the "BTN\_Camera" Button control that was just created.
  2. Press Ctrl + C: the Button control is copied to the clipboard.
  3. Press Ctrl + V: the mouse cursor changes and the control shadow appears under the cursor.
  4. Click in the window beside the "BTN\_Camera" control: the new Button control is automatically created.
  5. Open the control description window (double-click the control):
    - Give a name to the control: "BTN\_Photo\_Album".
    - Select an image in the image catalog.  
Remark: Don't forget to change the default image name.
  6. Validate the description window.


Remark: This button control corresponds to a "Copy - Paste" of the photo capture Button control, the style characteristics are automatically the same. There is no need to modify them!

- The WLanguage code associated with this Button control is used to open the device's photo album and select an image to display it in the Image control of the product form.
1. Select "Code" from the popup menu of button (right mouse click).

2. Replace the WLanguage code of the "Click" event with the following WLanguage code:

```
// Local variable
sPhoto is string
// Start the selection
sPhoto = AlbumPicker(albumImage)
IF sPhoto <> "" THEN
    IMG_Photo = sPhoto
END
```

In this WLanguage code, **AlbumPicker** is used to retrieve the selected photo from the photo album.

3. Save the modifications by clicking  among the quick access buttons.
4. Close the code window (click X at the top right corner of code editor).

## Managing the bar code of product



Caution!

This paragraph requires a device equipped with a camera.

Otherwise, you will not be able to test its use.


Furthermore, this paragraph requires the application to be installed on the device because it is using hardware resources that are not accessible in Simulation mode.

## Overview

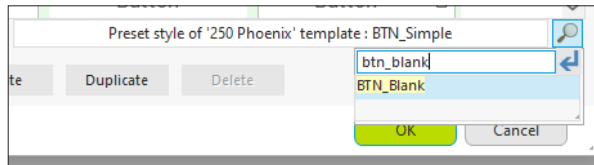
The "Product" data file contains a "Bar\_Code" item. This item is used to store the value of a bar code. Some devices (especially the ones equipped with a camera) can scan a bar code to retrieve its value.

The bar code will be managed via the device camera and by a specific WLanguage function.

## Implementation

- To create a Button control to manage bar codes:
1. Display (if necessary) the "WIN\_Product\_form" window in the editor.
  2. Add a Button control in the window:
    - On the "Creation" pane, in the "Usual controls" group, click .
    - The control shape appears under the mouse.
    - Create the control next to the "Bar code" control (resize the edit control if necessary).
  3. Open the Button control description window (double-click the control):
    - Give a name to the control: "BTN\_Bar\_Code".
    - Clear the control caption.
    - Select an image of bar code in the image catalog (type the "Code" keyword for example).
  4. Validate the description window.
  5. In the editor, reduce the control size.

- As for the Button control used to take a photo, we will modify the style associated with the control:
  1. If necessary, select the Button control you have just created.
  2. Open the popup menu and select "Choose a style".
  3. In the window that lists the available styles, click on the magnifier and specify "BTN\_Blank".



4. Press Enter to validate. The "BTN\_Blank" style is automatically selected.
  5. Validate the window for style selection.
  6. The chosen style is automatically applied to the Button control.
- The WLanguage code of this Button control will make it possible to scan the bar code.
    1. Select "Code" from the popup menu of the control (right click).
    2. Write the following WLanguage code in the "Click" event:

```
// Local variable
bc is BarCodes
// Start the scan
bc = BCCapture()
IF bc..Content <> "" THEN
    EDT_Bar_code = bc..Content
END
```

In this WLanguage code, **BCCapture** allows you to decode the information stored in a bar code using the camera of the device.

3. Save the modifications by clicking  among the quick access buttons.
4. Close the code window (click X at the top right corner of code editor).


## Managing anchors in the Product form

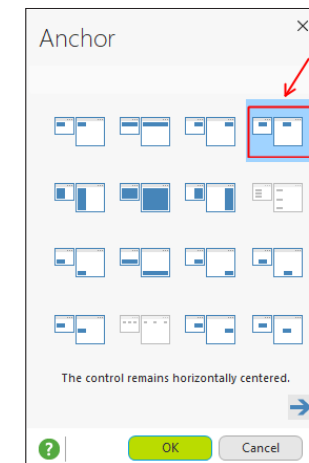
All controls have been positioned in the Product form. The vertical and horizontal resolution may differ according to the devices.

The anchoring is used to adapt the size of controls to the resolution and to avoid displaying "empty" areas in the window (especially at the bottom right).


If the application's target device is known since the creation of the project, the windows adopt the size of the target device. No anchoring is required.


If the target device of application is not known or if several devices are used, the smallest common resolution must be chosen as soon as project creation. The anchoring is required in this case.

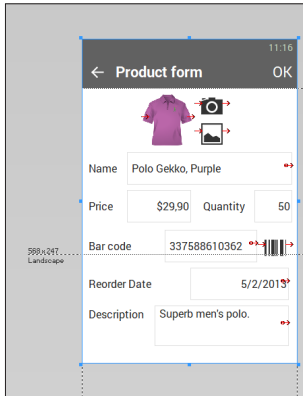
- To define the anchoring that will be applied to the different window controls:
  1. Select the following controls (click each control while keeping the Ctrl key down):
    - the Image control that displays the product image,
    - the Button control used to take photos,
    - the Button control for selecting a photo from the album.
  2. Display the popup menu of selection (right mouse click) and select "Anchor".
  3. Select "Centered horizontally" (  ).



4. Validate.

- To define the anchoring that will be applied to the edit controls:
  1. Select the following edit controls (click each control while keeping the Ctrl key down):
    - Name,
    - Bar code,
    - Reorder date,
    - Description.
  2. Display the popup menu of selection (right mouse click) and select "Anchor".
  3. Select "Width" (  ).
  4. Validate.

- Repeat this operation for the Button control that allows managing barcodes. In this case, select "Right" (  ).
- All anchors have been defined in the window. The red arrows indicating the orientation of anchors are displayed:




## Using the Map control

We will now present the Map control and the GPS functions of WLanguage.

Our database contains a "Store" data file. This data file contains the addresses of stores that will be localized on a map via the mapping functions.


### Creating the window

We are going to create a new window and add a Map control into it.

- To create the window:
  1. Create a new blank window. Click  among the quick access buttons. The window for element creation is displayed: click "Window" then "Window". In the wizard, choose "Blank" and validate.
  2. The window for saving an element is displayed. Specify the window title: "Map of stores". Its name is automatically proposed: "WIN\_Map\_of\_stores". Validate.

### Creating the Map control

- To create the Map control:
  1. On the "Creation" pane, in the "Graphic controls" group, click "Map". The control shape appears under the mouse.
  2. Click inside the window to create the control.
  3. A message regarding the management of scrollbars in the window is displayed. Indeed, both the window and the Map control include their own scrollbar. Therefore, a conflict occurs. A single scrollbar must be enabled. We advise you to:
    - disable the scrollbar in the window because the window has a fixed size,
    - keep the scrollbar enabled in the Map control.

4. Click "Disable the automatic scrollbar". The Map control appears in the window editor.
5. Open the "Map" control description window (double-click the control for example).
6. In the control description window, specify the control name ("MAP\_STORE") and validate.
7. Save the modifications by clicking  among the quick access buttons.

## Displaying stores on the map

### Principle

We are now going to add the code used to display all stores of the "Store" data file on a map.

To do so, we will browse the "Store" data file with a FOR EACH loop. Then, the Marker variables will be used. A Marker variable is used to define a marker that will be displayed on a map.

A marker contains various information. We will be using the following information:



- Name,
- Latitude,
- Longitude.

**MapAddMarker** is used to add a marker on the Map control. Then, simply adjust the zoom level to see all markers in the Map control. If the zoom is not properly adjusted, some markers may not be visible or may overlap on the control.

### Implementation

- To enter the WLanguage code to display the stores:
  1. Right-click outside the window. Select "Code" from the popup menu that is displayed. The different WLanguage events associated with the window are displayed.
  2. Write the following WLanguage code in the event "Global declarations of...".

```
// Global variables
gMarker is Marker
// Load the stores
FOR EACH Store
  // Marker coordinates
  gMarker.Position.Latitude = Store.Latitude
  gMarker.Position.Longitude = Store.Longitude
  // Marker name
  gMarker.Name = Store.Name
  // Add the marker
  MapAddMarker (MAP_STORE, gMarker)
END
// Best zoom to view all markers on the map
MAP_STORE..Zoom = zoomAdaptSize
```

3. Save the modifications by clicking  among the quick access buttons.
  4. Close the code window (click X at the top right corner of code editor).
- Run the test of "WIN\_Map\_of\_stores" window in the simulator (  among the quick access buttons). Then, close the simulator.

## Using a Multiline Zone control

The "Multiline zone" control is often used on the mobile platforms. This control is used to group several controls:

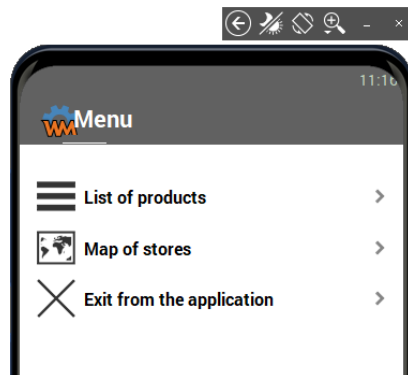
- options on a category,
- group of information about a contact, ...

This control can contain at the same time:

- rows defined in edit mode (static rows),
- rows defined at runtime, by programming (dynamic rows).

A Multiline Zone control will be used to create the main menu of our application. We are going to create a new window and insert a Multiline Zone control into it.

The window that will be created is as follows:



### Creating the window

We are going to create a window and add a Multiline Zone control into it.

► To create the window:

1. Create a new blank window:
  - Click among the quick access buttons.
  - The window for creating a new element is displayed: click "Window" then "Window". In the wizard, choose "Blank" and validate.
2. The window for saving an element is displayed. Specify the window title: "Menu". Its name is automatically proposed: "WIN\_Menu". Validate.

## Creating the Multiline Zone control

► To create the Multiline Zone control:

1. On the "Creation" pane, in the "Data" group, click "Multiline Zone". The control shape appears under the mouse.
2. Click inside the window to create the control.
3. Open the control description window (double-click the control for example).
4. In the description window, specify the control name ("MZ\_MENU") and validate.

The Multiline Zone control contains an empty line. We are going to add as many lines as the number of options found in our menu.

Our menu will include 3 options:

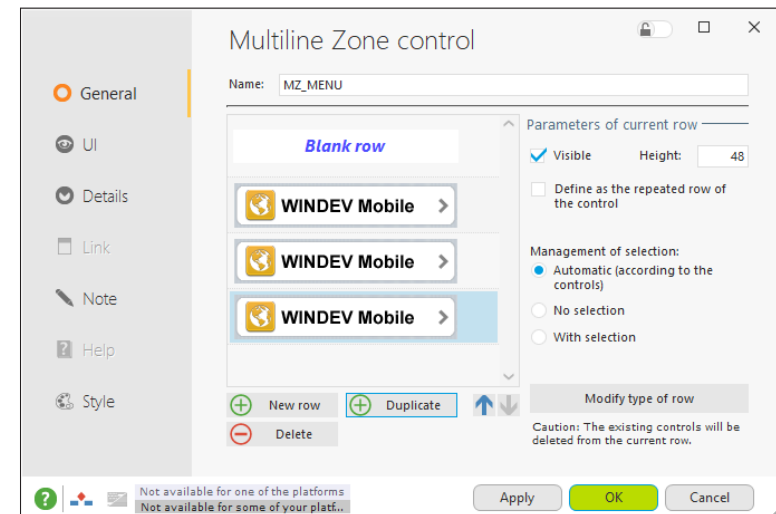
- List of products.
- Map of stores.
- Exit from the application.

### Modifying the Multiline Zone control

► To modify the Multiline Zone control:

1. Open the control description window (double-click the control for example).
  2. Click "New row". A window opens: this window contains all preset line templates.
  3. Select the "Simple row with picto" template and validate. Repeat this operation twice.
- Now, the multiline zone contains:

- a "blank" row,
- 3 "simple rows with picto".



4. We are going to delete the blank row that is useless in our example:


- Select the blank row (the first row).
- Then click "Delete".

5. Validate the description window. Your menu is created.

Each row includes an Image control, a Static control and an arrow image. We are now going to modify the Image control and the Static control of each row in order to represent the menu action.

#### Modifying the 1st row: access to the list of products

► To modify the image in the first row of the Multiline Zone control:

1. Click the Image control in the first row.
2. Open the image description window (double-click the control).
3. In the description window:
  - Give a name to the image ("IMG\_ListOfProducts" for example).
  - Click on the button . Select "Catalog" from the popup menu that appears to choose an image that represents the action.
  - In the image catalog window, type "List" in the search control and press Enter.
  - Select an image via a double click.
  - In the window for configuring the generated image, select a size (80 for example, W (Width) = 80 and H (Height) = 80), give a name to the image ("ListOfProducts" for example) and validate.



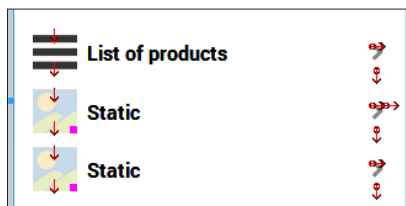
**Remark**

The image used in a Multiline Zone control is in "Homothetic extended centered" mode by default. The image will be automatically "resized" to the proper dimensions.

4. Validate the description window.


► To modify the Static control in the first row of the Multiline Zone control:

1. Click the Static control in the first row.
2. Open the description window (double-click the control).
3. In the description window "General" tab:
  - Give a name to the control ("STC\_ListOfProducts" for example).
  - Change the caption ("List of products" for example).
4. Validate the description window.



#### Modifying the 2nd row: access to the map of stores

► To modify the image in the second row of the Multiline Zone control:


1. Click the Image control in the second row.
2. Open the image description window (double-click the control).
3. In the description window:
  - Give a name to the image ("IMG\_MapOfStores" for example).
  - Click on the button . Select "Catalog" from the popup menu that appears to choose an image representing the action.
  - In the image catalog window, type "Map" in the search control and press Enter.
  - Select an image via a double click.
  - In the window for configuring the generated image, select a size (80 for example, W (Width) = 80 and H (Height) = 80), give a name to the image ("MapOfStores" for example) and validate.
4. Validate the description window.

► To modify the Static control in the second row of the Multiline Zone control:

1. Click the Static control in the second row.
2. Open the description window (double-click the control).
3. In the description window "General" tab:
  - Give a name to the control ("STC\_MapOfStores" for example).
  - Change the caption ("Map of stores" for example).
4. Validate the description window.

#### Modifying the 3rd row: exit from the application

► To modify the image in the third row of the Multiline Zone control:

1. Click the Image control in the third row.
2. Open the image description window (double-click the control).
3. In the description window:
  - Give a name to the image ("IMG\_Exit" for example).
  - Click on the button . Select "Catalog" from the popup menu that appears to choose an image representing the action.
  - In the image catalog window, type "Close" in the search control and press Enter.
  - Select an image via a double click.
  - In the window for configuring the generated image, select a size (80 for example, W (Width) = 80 and H (Height) = 80), give a name to the image ("Close" for example) and validate.
4. Validate the description window.



► To modify the Static control of third line found in the Multiline Zone control:

1. Click the Static control in the third row.
2. Open the description window (double-click the control).
3. In the description window "General" tab:
  - Give a name to the control ("STC\_Exit" for example).
  - Change the caption ("Exit from the application" for example).
4. Validate the description window.

### Programming the menu


► Finally, we will write the WLanguage code necessary to perform each action in the menu:

1. Right-click the Multiline Zone control and select "Code".

**Caution:** make sure you select the Multiline Zone control and not one of the controls belonging to it.

2. In the code editor, type the following WLanguage code in the event "Selection (click) of a line in...":

```
SWITCH MZ_MENU
CASE 1 // List of products
  OpenMobileWindow(WIN_List_of_products)
CASE 2 // Map of stores
  OpenMobileWindow(WIN_Map_of_stores)
CASE 3 // Exit from the application
  Close()
END
```

3. Save the modifications by clicking  among the quick access buttons.
4. Close the code window (click X at the top right corner of code editor).

### Application test


The last step consists in specifying that the menu window is the first application window. To do so, we are going to run a full project test and define the first project window.

► To define the first project window:

1. Select the "WIN\_Menu" window in the "Project explorer" pane.
2. Display the popup menu.
3. Select "First project window". A specific icon (with a small 1) is displayed in front of the window name, in the "Project explorer" pane.

Until now, the test of windows was run individually by clicking  among the quick access buttons.

► To start the project test on the simulator:

1. Click  among the quick access buttons.
2. Your project starts with the menu window. Click an option of your menu to check whether the different links are correct.

## LESSON 5.3. MANAGING THE ORIENTATION OF WINDOWS WITH THE LAYOUTS

### This lesson will teach you the following concepts

- The benefit of anchors.
- Creating a layout.
- Using a layout.



Estimated time: 15 mn

## Overview

In most cases, an application for mobile can be used in portrait mode and in landscape mode. The interface of a window used in portrait mode can be changed to be used in landscape mode. WINDEV Mobile manages the change of device orientation via anchors. Most of the time, the anchors are sufficient to adapt the display to the different orientations. However, if the interface must change according to the orientation (different positioning of controls in portrait mode and in landscape mode for example), the anchors are not sufficient. To manage a different interface in portrait mode and in landscape mode, WINDEV Mobile gives you the ability to use layouts.

- To follow this lesson, open (if necessary) the "WM Managing Products" project that was created in the previous lesson.



Answer

If the "WM Managing Products" application was not created, a corrected project is available. This project allows you to follow this lesson.

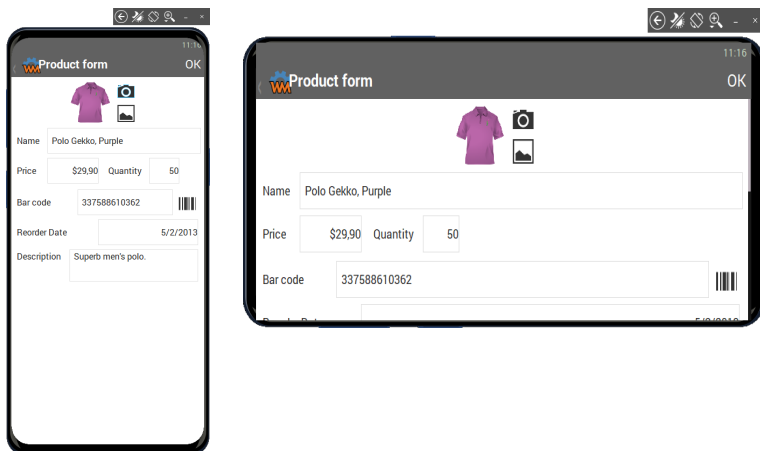
To open this intermediate project, in WINDEV Mobile's home page (Ctrl + <), click "Tutorial" and select "iOS/Android application (With windows)".

**A corrected project is also available.** This project groups all windows developed in this lesson.

To open this corrected project, in WINDEV Mobile's home page (Ctrl + <), click "Tutorial" and select "iOS/Android application (Answer)".

## Using anchors

- The anchors have already been used in the "WIN\_Product\_form" window. In test mode, we get the following interfaces:



All controls found in portrait mode are also displayed in landscape mode but a lot of space is not used and the window must be scrolled to access all controls.

We are going to improve the display in landscape mode via layouts.



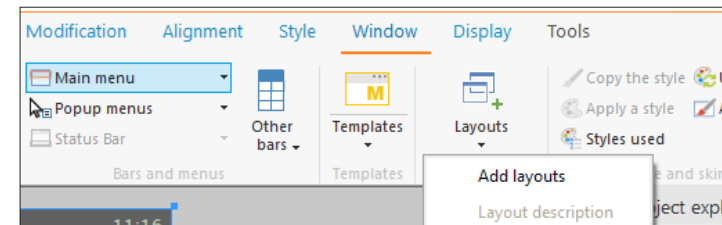
Remark

The layout is used to define several views of a window in the same project without duplicating this window.

This gives you the ability to define:

- a specific view for the portrait mode,
- a specific view for the landscape mode,
- a view specific to the phone,
- a view specific to the tablet,
- ...

- To create a new layout:
  1. Open the "WIN\_Product\_form" window if necessary.
  2. On the "Window" pane, in the "Layouts" group, expand "Layouts" and select "Add layouts".



3. The wizard for creating a layout starts.
4. Go to the next step of the wizard.

5. The wizard proposes to manage the differences between the phone and the tablet. The tablets are not used in our example. Keep the option "This window will be displayed on phone only".

**Phone/Tablet**

Do you want to manage the UI difference between tablet and phone?

☒ This window will be displayed on phone only

☐ Create layouts  
I need to modify the layout of controls between phone and tablet.

☐ Use the anchors only  
The anchors are sufficient to reposition the controls.

6. Go to the next step of the wizard.

7. The wizard proposes to manage the Portrait/Landscape orientation of application. That's what we are going to do in this example: check the option "Create layouts".

**Portrait/Landscape**

Do you want to use anchors or layouts to manage the orientation of the device?

☒ Create layouts  
I need to modify the layout of controls according to the orientation.

☐ Use the anchors only  
The anchors are sufficient to reposition the controls.

8. Go to the next step of the wizard.

9. The wizard proposes to use a different presentation for each platform used. In this example, the same presentation will be used for the Android platform and for the iOS platform. Select "No specific layout for each OS".

**Operating system (OS)**

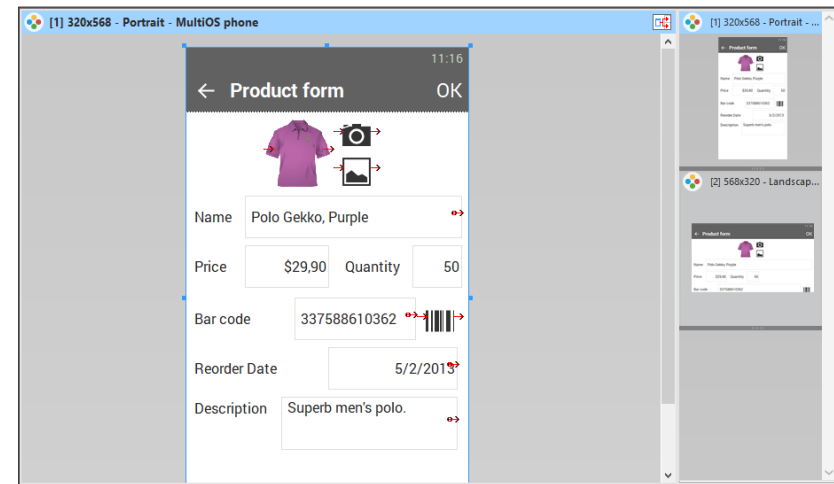
Layouts are required to create OS-specific UIs (Android, iOS...).

☐ Create layouts for each OS  
I am using features specific to one of the OS


☒ No specific layout for each OS  
I am using cross-platform features only

10. Validate the wizard.

► Two layouts are created in our example. These layouts are displayed in thumbnail format, on the right of the "WIN\_Product\_form" window.

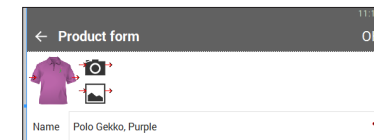


► We are going to modify the layout of landscape mode:

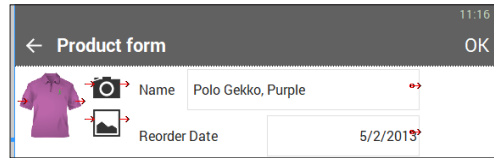
1. Double-click the thumbnail of "Landscape - MultiOS Phone" layout: the window corresponding to this layout is displayed in the middle of the editor.
2. Click  (top right corner of main window, in the title bar) to enable the "automatic dissociation" mode. Via this option, any modification performed on one of the specific windows of layout will not be applied to the other ones.

3. We are going to modify the position of controls in the layout:

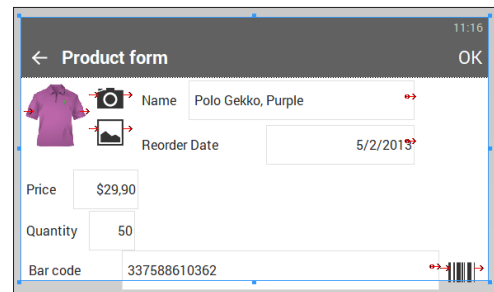
- Select the Image control corresponding to the product photo as well as the two Button controls and move them to the left of the window.



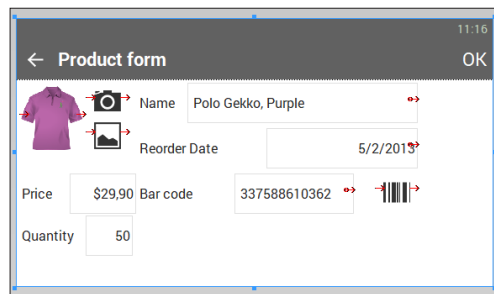
- Select the "Name" and "Reorder date" controls and place them to the right of the Image control and its Button controls. Resize the controls if necessary.



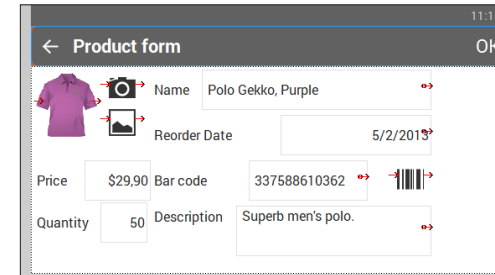
- Select the "Price" and "Quantity" controls and move them under the Image control on the left.



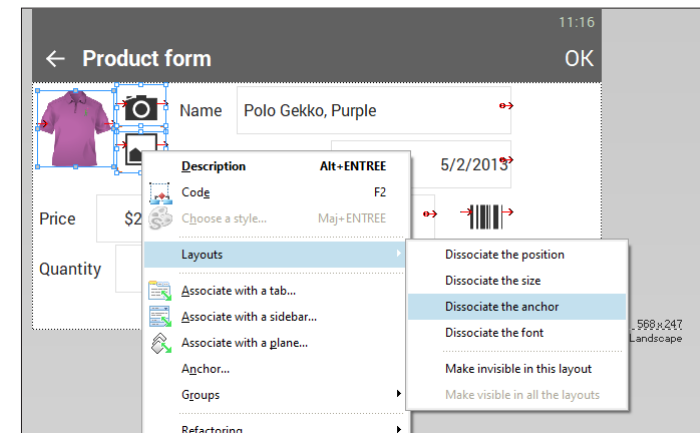
- Select the "Bar code" control and the associated Button control and place them to the right of the price.




- Select the "Description" control and position it on the right of quantity. Reduce the control height if necessary.

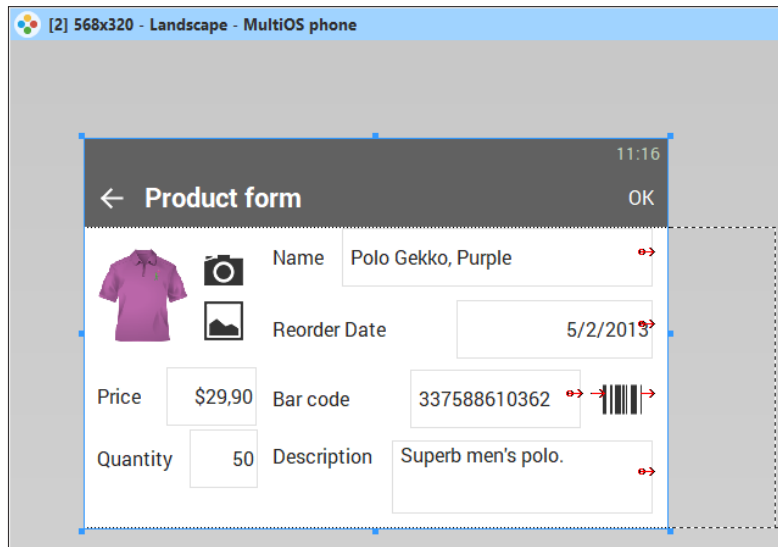


- Select the Image control. The Image control as well as the two Button controls are enclosed by a red line. Indeed, an anchoring conflict occurs with the Name control.
- Therefore, we are going to modify the anchor of these controls:
  1. Select the Image control and the two Button controls.
  2. Display the popup menu and select "Layouts .. Dissociate the anchor".



3. Define the new anchor that will be applied to the controls:
  - Display the popup menu of the controls and select "Anchor".
  - In the window that is displayed, select .

► You get the following interface:



## LESSON 5.4. WINDOW WITH SEARCH

This lesson will teach you the following concepts



- Creating a query with parameters.
- Creating a window via the wizard.
- Implementing the search.
- Managing "Pull to refresh".
- Adding a sliding menu.

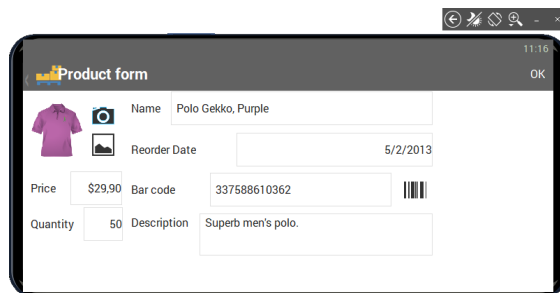


Estimated time: 30 mn

### Running the application test

► To run the project test:

1. Click  among the quick access buttons.
2. Your project starts with the menu window.
3. Click the list of products.
4. Click a product. The product form appears in portrait mode.
5. Change the window orientation by clicking  in the simulator menu.
6. The layout defined for the landscape mode is automatically displayed:



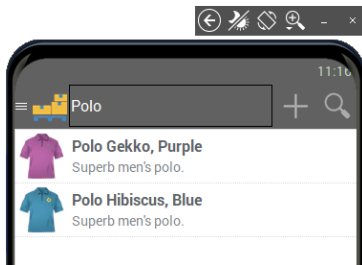
7. Close the test window of application.

## Overview

In the lesson 5.2, we have explained how to create a window used to list the products in a loop. We are going to create a window based on the same principle but proposing a search on the product name:

- the window displays the list of products in a loop.
- when the user types a product name in the search area, the corresponding products are displayed in the loop.

In our example, this search will be performed on the "Product" data file.  
The interface of "WIN\_List\_of\_products\_Advanced" will be as follows:



To create this window, we are going to:

- Create the query for selecting records in the Product data file.
- Create and configure the search window.



Remark

### What is a select query?

A select query is a query that will "choose" the records corresponding to the specified criteria.  
This type of query is called a select query because the SELECT command is used in SQL language.




Answer

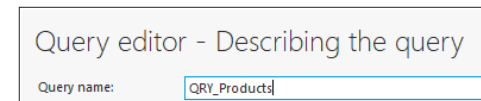
If the "WM Managing Products" application was not created in the previous part:

- an intermediate project is available. This project contains the windows created in the previous parts. This project allows you to follow this lesson and to create the new windows.  
To open this project, in WINDEV Mobile's home page (Ctrl + <), click "Tutorial" and select "iOS/Android application (With windows)".
- a corrected project is available. This project contains all windows created in this part and it allows you to check your operations.  
To open this corrected project, in WINDEV Mobile's home page (Ctrl + <), click "Tutorial" and select "iOS/Android application (Answer)".

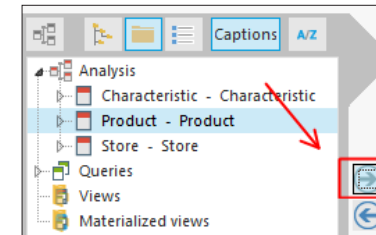
## Creating the query used to find the products

### Creating the query

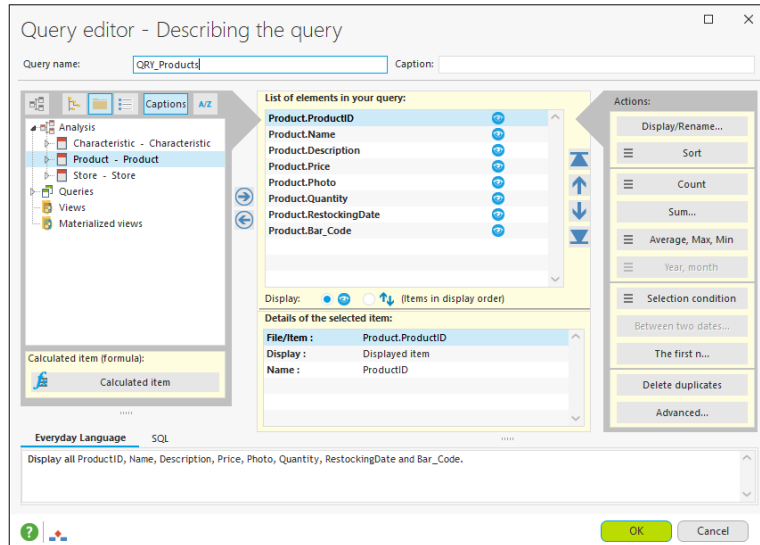
- The query editor will be used to create the query.
  1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Query". The query creation wizard starts.
  2. Select the "Select" option.  
Indeed, this query will be used to select records. Go to the next step.
  3. The query description window is displayed.
  4. Give a name and a caption to your query: type "QRY\_Products" instead of "QRY\_NoName1" in the "Query name" area and "Find products on the name" in the "Caption" area:



- To build the query, we are going to select the elements that will be displayed in the result.
  1. The query must be used to display the characteristics of selected product:
    - Select the "Product" data file in the "Analysis" window area.
    - Click on the arrow to select all data file items in the query.




2. The description window of query is as follows:



3. Validate the query description window ("OK" button).
4. The graphic representation of the query and the window for saving the query are displayed.
5. Validate the displayed information.








### Query test

Like all the elements found in a WINDEV Mobile project, you have the ability to run the test of query that was just created:

1. Click .
2. The result is displayed in a window:

Preview of QRY\_Products

7 records correspond to your query.

ProductID	Name	Description	Price	Photo	Quantity	Restock
1	Polo Gekko, Purple	Superb men's polo.	29,90		50	
2	Polo Hibbicus, Blue	Superb men's polo.	25,90		40	
3	Cap Blend, Orange	Superb cap.	18,90		30	
4	Cap Blend, White	Superb cap.	18,90		30	
5	Thong Precilia, Green	Superb pair of thong.	16,50		80	
6	Thong Precilia, Black	Superb pair of thong.	16,50		20	
7	Thong Precilia, Brown	Superb pair of thong.	15,50		20	

The result lists ALL products.

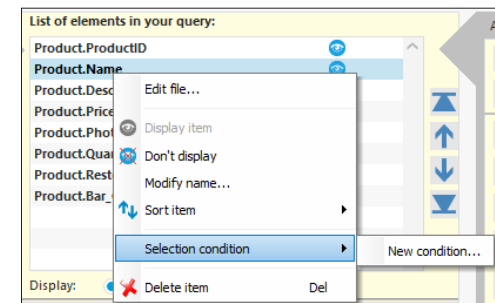
In our case, we want to display the products corresponding to the search criteria, the product name. To do so, we must use a query with parameters.

3. Close the window.

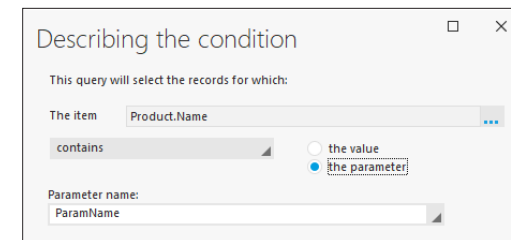
### Adding a selection condition

In our example, the user will be able to select a value for the product name. We must modify the query in order for this search criterion to correspond to a query parameter.

- To define a query parameter, display the description window of query: double-click the background of graphic query representation (or select "Query description" from the popup menu).
- To manage the "Product name" parameter:
  1. In the middle of window, select the Product.Name item.
  2. Display the popup menu and select "Selection condition .. New condition".

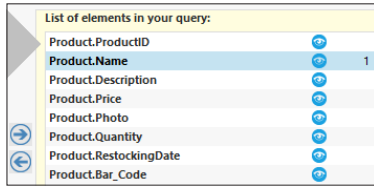


3. In the window that is displayed, we are going to specify that the selection condition corresponds to a parameter:

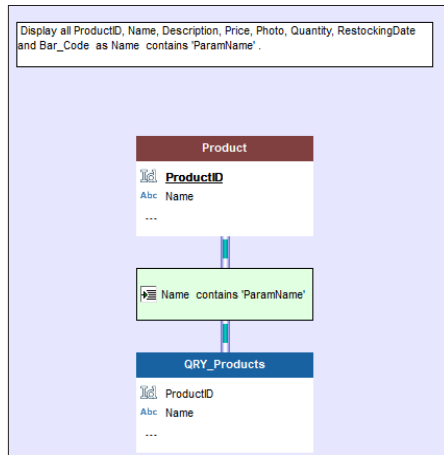


- Select "contains".
  - Select "the parameter".
  - Keep the parameter name automatically proposed: "ParamName".
4. Validate the condition description window. The number "1" appears on the right of "Product. Name" item, indicating that a selection condition was defined.






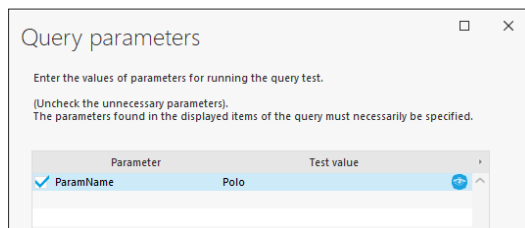
5. Validate the description window of query.
6. The query graph is modified to take into account the selection condition that was defined.



7. Save the query by clicking  among the quick access buttons.

### Test of query with parameters

- To run the test of query with parameters:
  1. Click .
  2. A window is displayed, allowing you to type the different query parameters.
  3. Select the ParamName parameter. In the bottom section of window, type "Polo".



4. Validate the window. The query result corresponding to the specified parameters is displayed.
5. Close the window.

We are now going to create the interface of our window based on this query.

## Creating the interface


The search window will be created via the wizard for window creation.

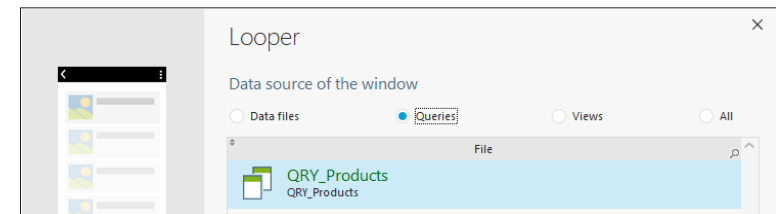


### Remark

The window creation wizard offers many preset windows. These windows propose modern interfaces for your applications. Most of these windows can be generated from your data.

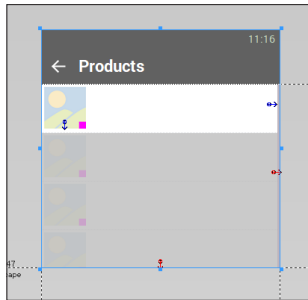
## Creating the window

- To create the search window:
  1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Window" then "Window".
  2. In the wizard, choose "Looper" and validate.
  3. The window creation wizard starts.
  4. Choose the platform to use: "Generic Android phone". Go to the next step of the wizard.
  5. The wizard proposes to choose the data source associated with the window. In our case, it is a query:
    - Click "Queries".
    - Select the query that was just created: "QRY\_Products".



6. Go to the next step.
7. Select the looper style: "Image + Title + Caption below". Go to the next step.
8. The wizard automatically proposes the query items corresponding to the generated looper. Keep the proposed options and go to the next step.
9. Keep the sort item proposed by default ("Name"). Go to the next step.
10. The wizard proposes several options for generating the Looper window. In our example, keep the default options. Go to the next step.
11. Give a title and a name to the generated window. In our case:
  - For the title, type "Products".
  - For the name, type "WIN\_List\_of\_products\_Advanced".

12. Validate the wizard. The window is automatically created, displayed in the editor and saved.



► We are now going to modify the "WIN\_List\_of\_products\_Advanced" window in order to display the Product form that was created in a previous lesson.

1. Right-click the Looper control and select "Code".

**Caution:** make sure you select the Looper control and not one of the controls it contains.

2. In the code editor, type the following WLanguage code in the "Initializing..." event:

```
QRY_Products.ParamName = Null
```

3. This code line is used to initialize the value of parameter found in the "QRY\_Products" query used by the Looper control. By default, the value of this parameter is set to "Null", allowing you to ignore the parameter. Therefore, all products will be displayed in the window.

4. In the code editor, type the following WLanguage code in the event "Selecting a row ...":


```
HReadSeek(Product, ProductID, QRY_Products.ProductID)
OpenMobileWindow(WIN_Product_form)
```

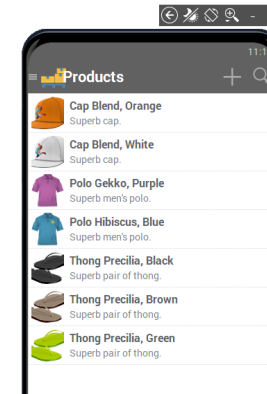
Let's study this code:

- The Looper control is based on the QRY\_Product query. When selecting the product in the Looper control, the selected record is the one found in the query.
- During a click on the control row, we want to open the form window that was created beforehand. This window is based on the Product data file.
- The record selected by the query must be found in the "Product" data file in order to load the buffer of the selected data in memory. The operation is performed by **HReadSeek**.
- Then, the form window named "WIN\_Product\_form" is opened by **OpenMobileWindow**.

5. Save the modifications by clicking  among the quick access buttons.

6. Close the code window (click X at the top right corner of code editor).

7. Test the window that was just created in the simulator (  among the quick access buttons).



8. Click one of the products: the form window is displayed.

9. End the test.

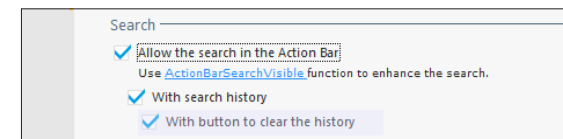
### Managing the search

We are now going to manage the search. To do so, we are going to:

- Allow the search in the Action Bar.
- Create a search button in the Action Bar.

► To allow the search in the Action Bar:

1. Display (if necessary) the "WIN\_List\_of\_products\_Advanced" window in the editor.
2. Display the description window of Action Bar (double-click the Action Bar).
3. In the "Details" tab, check "Allow the search in the Action Bar".



► To create a search button in the Action Bar:


1. In the "General" tab of the description window of Action Bar:
2. Click number 2. The interface for typing an option in the toolbar is displayed.
3. Click the "+" button to add an option. A new default option is created at the top right. Modify the characteristics of this option:

- In the "Caption" area, type "Find".
- In the "Preset image" area, expand the list and select "Find".
- Validate the description window.

4. The code of this option is used to make the search area visible. To write this code:

- Select (if necessary) the Action Bar of window.
- Click the search button.
- A drop-down menu with the "Find" option is displayed.
- Right-click the option.
- Select "Code" from the popup menu that is displayed.
- Write the following WLanguage code in the event "Selecting the menu":

```
ActionBarSearchVisible (True)
```

5. Save the modifications by clicking  among the quick access buttons.
6. Close the code window (click X at the top right corner of code editor).
7. Select the Action Bar and display the associated code (press F2 or select "Code" from the popup menu).
8. In the code editor, type the following WLanguage code in the event "Validating the search...":

```
QRY_Products.ParamName = ACTB_ActionBar..SearchValue  
LooperDisplay (LOOP_QRY_Products, taReExecuteQuery)
```


9. Let's take a look at this WLanguage code:

- The query parameter is initialized with the search value typed in the Action Bar.
- Then, the Looper control is redisplayed by **LooperDisplay**. The **taReExecuteQuery** constant is used to re-run the base query of Looper control and therefore to take the new parameter into account.

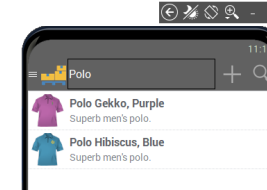
10. Save the modifications by clicking  among the quick access buttons.

11. Close the code window (click X at the top right corner of code editor).

12. A button for product addition can also be created in this window. This operation was already performed in the lesson 5.2 "Creating a new product". The same operations must be performed. Only the code of "+" button must be adapted.

13. Test the window that was just created in the simulator (  among the quick access buttons).

- Click the magnifier.
- Type "Polo" in the search area.
- Validate (ENTER key).
- The list of products containing "Polo" is displayed.



14. Close the simulator.

## Improving the window

### Managing "Pull to refresh"

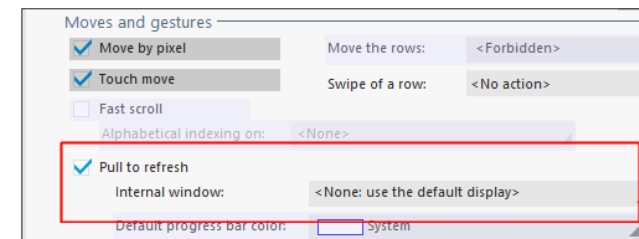
A new feature will be added to our window: the management of "Pull to refresh". This feature allows the user to "pull" a Table or Looper control in order to refresh its content.

During this action, a refresh bar automatically appears in the exposed area:

- The bar indicates that you must pull to refresh.
- Then, the bar indicates that you must release to refresh.
- The bar indicates that the refresh operation is in progress. A progress bar is displayed during the refresh duration.
- The control is updated.

► To use the "Pull to Refresh" feature:

1. Display (if necessary) the "WIN\_List\_of\_products\_Advanced" window in the editor.
2. Select the Looper control and display the description window of control.
3. In the "Details" tab of the description window of control, in the "Moves and gestures" area, check "Pull to refresh".




Remark


A specific internal window can be used to manage the "Pull to refresh" feature. In this example, we will be using the default window. See the online help for more details.

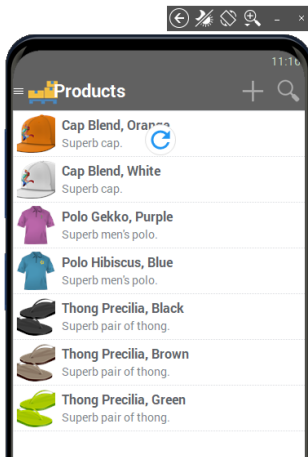
4. Validate the control description window.

- The setting of "Pull to refresh" has added:
  - the refresh bar that will be displayed to the user during the operation.
  - the "Refreshing by pull/release" event among the Looper control events. This event is automatically called during the refresh gesture. We are now going to modify the WLanguage code of this event in order to manage how the control is refreshed.
- To modify the WLanguage code of the event "Refreshing by pull/release":
  1. Select the Looper control and display the associated event (press F2 for example).
  2. In the code editor, type the following WLanguage code in the event "Refreshing by pull/release":

```
LooperDisplay(LOOP_QRY_Products, taReExecuteQuery)
```

3. As already seen for the search management, **LooperDisplay** is used to redisplay the Looper control. The **taReExecuteQuery** constant is used to re-run the base query of Looper control and therefore to take into account the new records entered in the database.
4. Save the modifications by clicking  among the quick access buttons.
5. Close the code window (click X at the top right corner of code editor).

- Test the window that was just created in the simulator (  among the quick access buttons).
  1. Click the top of looper with the mouse and move the mouse to the bottom.
  2. Release the mouse. The looper is updated.



3. Close the simulator.

This example allows you to understand the implementation of "Pull to refresh" feature.

This feature can be used for example in the same application in HFSQL Client/Server where other users would update or add products. These modifications could be displayed by the "Pull to refresh" feature.

### Using a sliding menu

In several mobile applications, the menu does not correspond to a "static" window. It corresponds to a sliding window displayed via an option of Action Bar and/or via a window swipe.


We are going to modify the "WIN\_List\_of\_products\_Advanced" window in order to add a "sliding menu". This menu will be using the Multiline Zone control of "WIN\_Menu" window that was created beforehand.

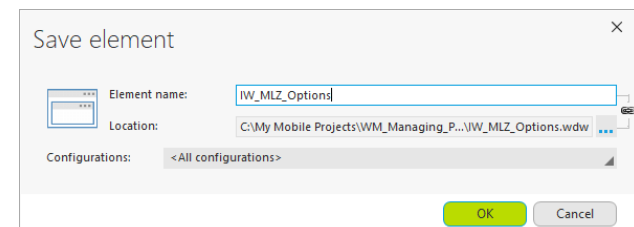
- To create a sliding menu, we are going to:
  - Create an internal window. This internal window will contain the menu options.
  - Modify the "WIN\_List\_of\_products\_Advanced" window to display the menu.



Remark

An internal window is a simple window with no Action Bar and no toolbar. An internal window is used to easily include a set of controls in another window.

- To create the internal window containing the menu:
  1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Window" then "Internal window".
  2. The internal window is automatically opened in the editor.
  3. The window for saving an element is displayed. Type the name of internal window: "IW\_MLZ\_Options".



4. Validate.
5. Display the description window of internal window ("Description" from the popup menu).

6. In the "UI" tab, specify the dimensions of this internal window:

- Width: the one that suits you. It must be sufficient to entirely see the controls of drop-down menu. In theory, the sliding menu must be narrower than the window above which it is displayed (260 for example).
- Height: This height must correspond to the height of window above which the sliding menu is displayed. In our case, this height is set to 248.



7. Specify a background color for the internal window in the "Style" tab. Indeed, for the menu to be transparent, the parameter "Background color of internal window" must correspond to "White".

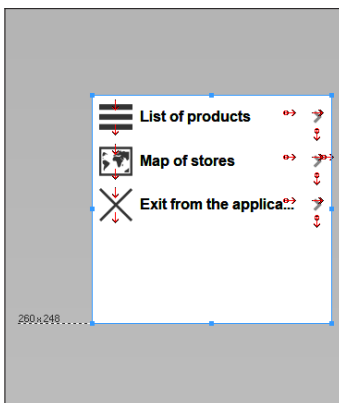
8. Validate.

9. Save the window by clicking among the quick access buttons.

► To add the menu options into the internal window:

1. Open the "WIN\_Menu" window previously created (double-click on its name in the "Project explorer" pane, for example).
2. Copy the controls found in the "WIN\_Menu" window to the "IW\_MLZ\_Options" internal window:
  - Select all elements found in the "WIN\_Menu" window (Ctrl + A).
  - Copy the elements (Ctrl + C).
  - Display the "IW\_MLZ\_Options" window (click its name in the bar of opened documents).
  - Paste the elements (Ctrl + V).

3. Via the selection handles, modify the width of Multiline Zone control so that it is entirely displayed in the internal window. Via the anchors, all controls found in the Multiline Zone control are also modified. You get the following window:



- We are now going to modify the WLanguage code used to open the list of products. Indeed:
  - we work with the "WIN\_List\_of\_products\_Advanced" window and not with the "WIN\_List\_of\_products" window anymore.
  - the "WIN\_List\_of\_products\_Advanced" window contains the sliding menu. Therefore, the "List of products" option must not re-open this window.
- We are going to change the WLanguage code used to select the Multiline Zone control.
  1. Select the Multiline Zone control.
  2. Display the associated WLanguage events (F2).
  3. Replace the line:

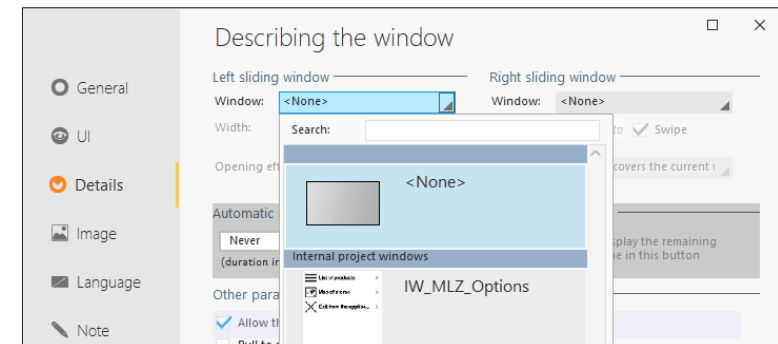
```
CASE 1 // List of products
  OpenMobileWindow(WIN_List_of_products)
```

by:

```
CASE 1 // List of products
  WinSlidingVisible(swLeft, False)
```

In this WLanguage code, **WinSlidingVisible** is used to make the sliding window displayed from the left invisible. Therefore, the list of products is displayed.

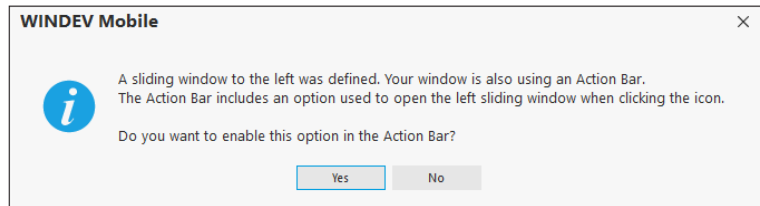
4. Save the modifications by clicking among the quick access buttons.
  5. Close the code window (click X at the top right corner of code editor).
- To associate the internal window with the "WIN\_List\_of\_products\_Advanced" window:
1. Display the window "WIN\_List\_of\_products\_Advanced" (click its name in the bar of opened documents).
  2. Display the description window.
  3. In the "Details" tab, in "Left sliding window", select "IW\_MLZ\_Options".



The "Swipe" option is used to automatically manage how the sliding window is displayed during the swipe.


4. Validate.

5. The window being associated with an Action Bar, the editor proposes to enable the option used to open the sliding window.



6. Accept.

► We are going to test the operating mode of the sliding menu in the simulator:

1. In the "Project explorer" pane, define the "WIN\_List\_of\_products\_Advanced" window as first project window (indeed, the "WIN\_Menu" window is now useless).
  - Select the window "WIN\_List\_of\_products\_Advanced" in the "Project explorer" pane.
  - Display the popup menu.
  - Select "First project window". A specific icon (with a small 1) is displayed in front of the window name, in the "Project explorer" pane.
2. Click  among the quick access buttons.
3. When clicking the menu of Action Bar, the sliding window of menu is displayed.

## LESSON 5.5. WINDOW WITH SCROLL

### This lesson will teach you the following concepts

---

- Creating an internal window by refactoring.
- Changing the window content by swipe.



Estimated time: 20 mn

## Overview

In the Windows applications, all the operations are performed via the mouse or the keyboard.

In mobile, the interfaces must be configured differently. With the touchdown screens, all the operations are performed with the fingers.

In the applications for mobile devices, WINDEV Mobile proposes several features to manages the specific moves of fingers (pull to refresh, double touch, sliding, ...).

In this lesson, we are going to create a new form window, used to view the products.

In this window, you will have the ability to go from a product form to another one by swipe.



Answer


If the "WM Managing Products" application was not created in the previous part, a corrected project is available. This project contains all windows created in this part and it allows you to check your operations.

To open this corrected project, in WINDEV Mobile's home page (Ctrl + <), click "Tutorial" and select "iOS/Android application (Answer)".

## Creating the Form window with swipe


### Creating the Form window

► To create the form window:

1. Create a new blank window.
  - Click  among the quick access buttons.
  - The window for creating a new element is displayed: click "Window" then "Window".
  - In the wizard, select the "Standard" tab, choose "Blank" and validate.
2. In the window for saving an element, specify:
  - the window title: "Product form".
  - the window name: "WIN\_Product\_form\_Advanced".
3. Validate.

We are now going to create an Internal Window control in the "WIN\_Product\_form\_Advanced" window. This control will host an internal window that will display the data to scroll.

► To create the Internal Window control:

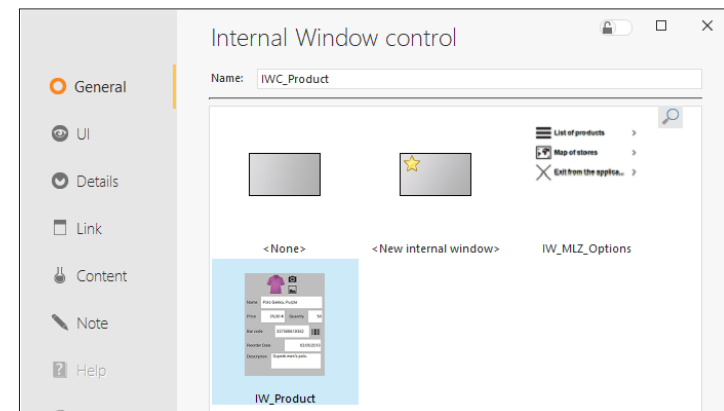
1. On the "Creation" pane, in the "Containers" group, expand "Internal window" and select "Swipe area (Internal window)".
2. Click the position where the control will be created in the window (top left corner).
3. Resize the Internal Window control so that it occupies the entire available space in the window.
4. Modify the anchor of Internal Window control ("Anchor" option from the popup menu): anchor the control in height and in width (  ).
5. Validate.

### Creating the internal window

- To simplify the creation of internal window, we are going to create it from the "WIN\_Product\_form" window found in our project:
1. If necessary, open "WIN\_Product\_form" in the editor (double-click its name in the "Project explorer" pane).
  2. Select all controls found in the portrait layout (Ctrl + A).
  3. Display the popup menu and select "Refactoring .. Create an internal window from the selection".
  4. The internal window is automatically created and the save window is displayed. Give the "IW\_Product" name and validate.
  5. Display the created internal window in the editor: click the "IW\_Product" button in the bar of opened elements.
  6. Display the description window of internal window. In the "UI" tab, modify the window width: 320. This width corresponds to the width of Internal Window control that was created in the "WIN\_Product\_form\_Advanced" window.
  7. Validate.

### Managing the swipe in the form window

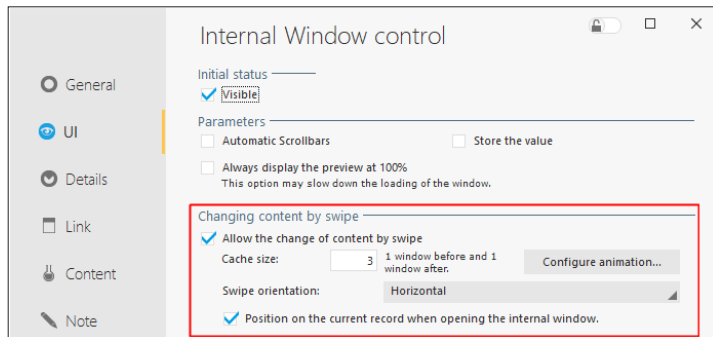
- To associate the internal window with the Form window:
1. Display the "WIN\_Product\_form\_Advanced" window in the editor (click its name in the bar of opened elements).
  2. Select the Internal Window control and display its description ("Description" from the popup menu).
  3. In the "General" tab, select the internal window that was just created ("IW\_Product").





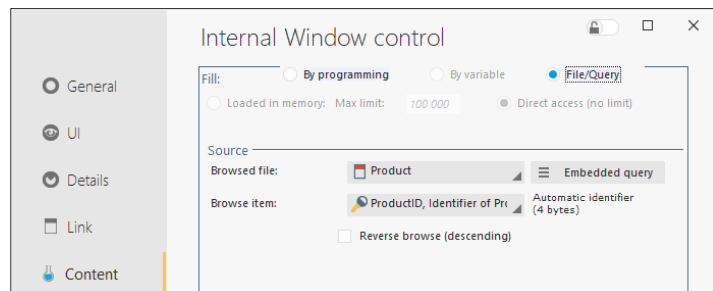
#### 4. In the "UI" tab, in "Changing content by swipe":

- The option "Allow the change of content by swipe" is already checked because we have chosen the Internal Window control for swipe.
- Check "Position on the current record when opening the internal window".
- The swipe orientation is "Horizontal" by default.



#### 5. In the "Content" tab, we are going to configure the mode for filling the internal window. This window will display the data of the Product data file:

- Click "File/Query".
- In the source, for the "Browsed file" option, select the "Product" data file.



#### 6. Validate the description window.

The swipe is implemented.

#### ► To avoid being bothered by the keyboard appearance during the swipe, we are going to use **SIPVisible**:

1. Display the WLanguage events of the Internal Window control in "WIN\_Product\_form\_Advanced".
2. Write the following code in the WLanguage event "Selection by swipe ...":

```
SIPVisible(False)
```

#### 3. Close the code editor.

#### ► To finish this window, we are going to add the validation option into the Action Bar of the window. We already did this in the lesson 5.2 ("Managing the product creation and modification"). Here, we will only present the main points:

1. Select the Action Bar (at the top of window).
2. Open the Action Bar description window.
3. Click number 2. The interface for typing an option in the Action Bar appears.
4. Click the "+" button to add an option. A new default option is created at the top right.
5. Modify the characteristics of this option:
  - In the "Caption" area, type "Done".
  - In the "Preset image" area, expand the list and select "Done".
  - Validate the description window.
6. To enter the code used to save the modifications performed in the "WIN\_Product\_form\_Advanced" window:
  - Select (if necessary) the Action Bar control of window.
  - Click on "OK".
  - A drop-down menu with the "Done" option is displayed.
  - Right-click the option.
  - Select "Code" from the popup menu that is displayed.
  - Write the following WLanguage code in the event "Selecting the menu":

```
ScreenToFile ()
HSave (Product)
Close ()
```

#### 7. Save the modifications by clicking among the quick access buttons.

#### 8. Close the code window (click X at the top right corner of code editor).

## Window test


Before running the window test, we are going to modify the "WIN\_List\_of\_products\_Advanced" window to directly open the form window that was just created.

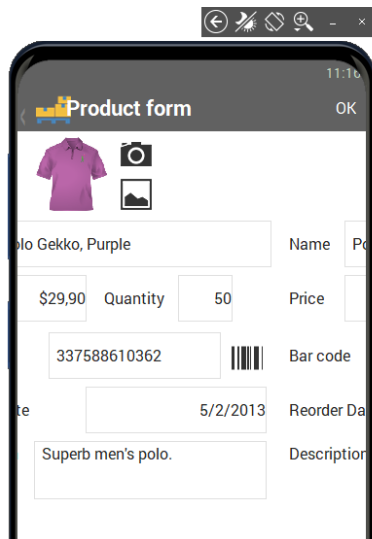
- To open the window with swipe from the list of products:
  1. Display the window "WIN\_List\_of\_products\_Advanced" in the editor (click its name in the bar of opened documents).
  2. Display the WLanguage events related to the Looper control.
  3. In the event "Selecting a row ...", replace the WLanguage code:

```
HReadSeek(Product,ProductID,QRY_Products.ProductID)
OpenMobileWindow(WIN_Product_form)
```

by:

```
HReadSeek(Product,ProductID,QRY_Products.ProductID)
OpenMobileWindow(WIN_Product_form_Advanced)
```

- We are going to test the operating mode of project in the simulator:
  1. Click  among the quick access buttons.
  2. In the Looper control that is displayed, click a product to display its form.
  3. Click the product form and move the mouse to the right or to the left. The product form automatically changes.



## Conclusion

This part explained the main concepts for developing Android or iOS applications.

Several themes have not been presented in this part:

- managing emails,
- managing notifications,
- using the visualization panel,
- using the debugger,
- ...

Don't hesitate to see the online help to discover and check new features.

## LESSON 5.6. DEPLOYING THE APPLICATION

### This lesson will teach you the following concepts

- Generating the Android application.
- Generating the iOS application.



Estimated time: 20 mn

### Overview

That's it, our application is created, we must now compile it and install it on the device in order to run its test.


If you have chosen to develop the application for Android only, you can follow the generation of Android application.

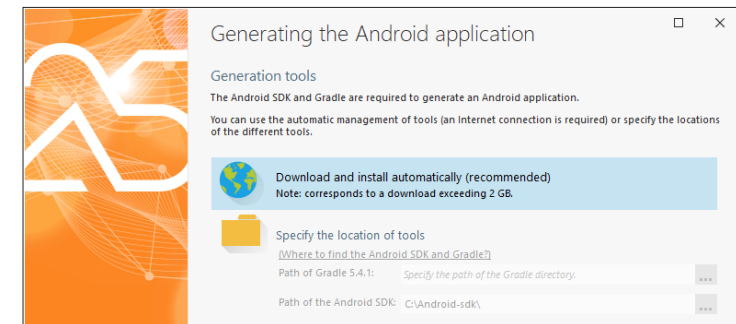
If you have chosen to develop the application for iOS, go to "Generating the iOS application".

If you want to develop the application for the two platforms, follow the generation for Android then the generation for iOS.

### Generating the Android application

► To generate the Android application:

1. If necessary, select the Android platform in the "Project explorer" pane.
2. On the "Project" pane, in the "Generation" group, click "Generate" (you can also click  among the quick access buttons).
3. The Android application generation wizard starts.
4. The first step of wizard allows selecting the application generation mode. Indeed, the generation format may change depending on the mode selected. For this tutorial, select "Generate for free deployment (private store, corporate fleet, etc)". Go to the next step.
5. The next step of the wizard is to check the tools needed to generate the Android application.



The wizard proposes to:

- **Download and automatically install the Android SDK and Gradle:** in this case, you will only have to validate the license. WINDEV Mobile takes care of everything. If an update is available, the generation wizard will automatically propose to perform the necessary updates. Only an Internet connection is required.
- **Use the tools automatically installed.** This option is available if the tools have been downloaded and installed during a previous generation.
- **Specify the location of tools already installed on your computer.** In this case, all you have to do is specify the setup paths of Gradle and Android SDK.

6. Select the option corresponding to your configuration and go to the next step.

Remark: If you have chosen to download and install the tools, going to the next step may be quite long and you may have to validate the license.

7. Go to the next step by clicking the arrow keys at the bottom of wizard. The next wizard step is used to:

- define the application name (displayed below the icon used to start the application) and the corresponding package.
- select the application icon in the image catalog of WINDEV Mobile ("stock" in the "Flat Soft" theme for example).
- define the email address used by default to send an error report if necessary.

**Generating the Android application**

**General information**

Name and package of the application

Name: WM\_Managing\_Products

Package: com.mycompany.wm\_managing\_products

The package is used to identify the generated application. By convention, we recommend that you use the domain name of the company, inverted and followed by the application name in lowercase.

Icon of the application

The visible icon in the menu is an image in PNG format.

Path: <Multiple values>

Preview:

Error report

Address used by default to send the error reports.

Email: myemail@mycompany.com

8. Go to the next step. This step is used to define:

- the splash screen of application,
- the information saved in the manifest,
- the start mode of application (when starting the device or not).

9. Go to the next step. The wizard is used to define the version number of application.

10. Go to the next step. This step is used to sign the application. The wizard proposes a generic signature that can be used for the application tests. A specific signature is required to distribute the application.

See the online help for more details.

11. Go to the next step. As we have chosen to generate the application for a fleet of devices, the wizard proposes to choose the type of devices. Keep the default options.



**Remark**

If you have a device connected to the PC, use "Detect hardware automatically". Thus, next time an application is generated, this type of device will be automatically selected.

12. The next step is used to include the data files in the application. For our example, pre-filled HFSQL data files are available. They will be supplied with the application. In this case, they must be specified in the "Integrating files" step.

- Click the "Add" button.
- Select the data files (.fic, .ndx and .mmo) found in the "EXE\Android application" directory of project. The list of files is displayed.



**Remark**

If your project is using the Android platform only, the files must be selected in the EXE directory of project.

13. Check the "Write" box for each file (required to be able to modify data from the application).

**Generating the Android application**

**Integrating files**

Specify the files that must be accessible in write mode so that they can be automatically extracted in one of the application subdirectories during the first start. You can also extract these files manually by using fExtractResource.

You can also easily store your HFSQL data in the cloud. [More details](#)

File	Read	APK destination	Write	Application destination
C:\My Mobile Pro...\Product.FIC	<input checked="" type="checkbox"/>	<Automatic>	<input checked="" type="checkbox"/>	<DB directory>
C:\My Mobile Pr...\Product.ndx	<input checked="" type="checkbox"/>	<Automatic>	<input checked="" type="checkbox"/>	<DB directory>
C:\My Mobile Pr...\Product.mmo	<input checked="" type="checkbox"/>	<Automatic>	<input checked="" type="checkbox"/>	<DB directory>
C:\My Mobile Projec...\Store.FIC	<input checked="" type="checkbox"/>	<Automatic>	<input checked="" type="checkbox"/>	<DB directory>
C:\My Mobile Proje...\Store.ndx	<input checked="" type="checkbox"/>	<Automatic>	<input checked="" type="checkbox"/>	<DB directory>
C:\My Mobi...\Characteristic.FIC	<input checked="" type="checkbox"/>	<Automatic>	<input checked="" type="checkbox"/>	<DB directory>
C:\My Mob...\Characteristic.ndx	<input checked="" type="checkbox"/>	<Automatic>	<input checked="" type="checkbox"/>	<DB directory>

14. Go to the next step.

15. Validate the other steps until you reach the "Configuration" step that is used to configure the options of Android SDK and the setup location.

**Generating the Android application**

**Configuration**

Android SDK

Minimum version required to run the application: Android 5.0

Information about the setup

Location of setup: Internal memory/Movable to the SD card

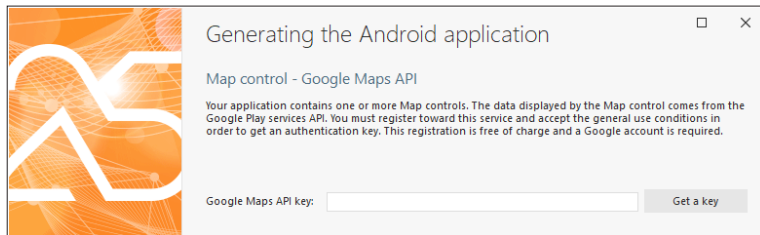
Advanced configuration

The advanced configuration is used to modify the manifest or the compilation and signature commands.


[Advanced configuration](#)

16. Go to the next step.

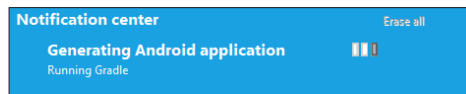
17. This step is specific to the use of Map control. It is used to enter the key required to use the Map control. If you own a Google Maps API key, type it. Otherwise, click "Get a key".



18. Go to the next step.

19. End the wizard. The generation is automatically performed in background task. The  icon indicates that the generation is in progress.

20. When the generation is ended, a popup window is displayed in the editor:



21. To copy and run the application on the device linked to the computer or on an emulator, click "Deploy".

22. A new window is displayed, allowing you to select the runtime device. If you own an Android device connected to the development computer, select the device connected to the PC.




## Remark

If the device is not connected to the development computer, you must:

1. Connect the device to the development computer.
2. In the explorer, open the generation directory of apk file corresponding to the Android application.
3. Copy the apk file into the "Download" directory of device.
4. Unplug the device.
5. Start the application for file management on the device.
6. Go to the "Download" directory and run the apk file. The application is automatically installed.

## Generating the iOS application

► To generate the iOS application:

1. If necessary, select the iOS platform in the "Project explorer" pane.
2. On the "Project" pane, in the "Generation" group, click "Generate" (you can also  among the quick access buttons).
3. The wizard for generating an iOS application starts.
4. The wizard is used to:
  - define the application name, the company and the copyright.
  - enter the bundle identifier.

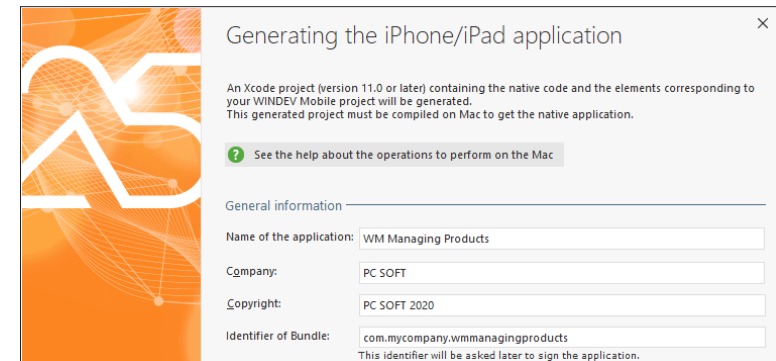


## Remark

This identifier is the unique identifier of your application beside Apple. It is defined and saved on the Apple developer account.

This identifier will be used to save your application in order to run its test and to deploy it. By default, WINDEV Mobile automatically proposes an identifier that respects the development standard of Apple. This identifier can be modified.

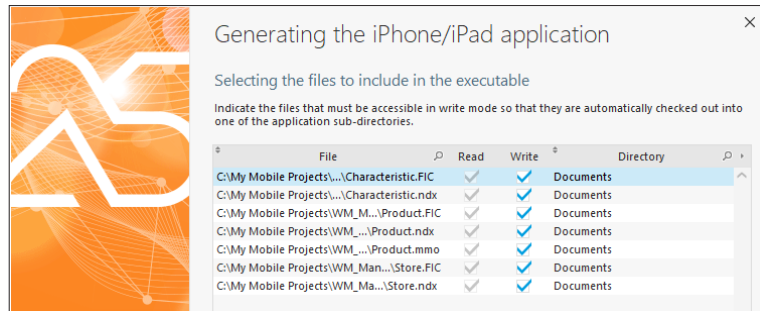
- define the email address used by default to send an error report if necessary.



5. Go to the next step.
6. You have the ability to specify the path of different application icons for each type of device.
7. Go to the next step.
8. Specify the characteristics of the splash screen.
9. Go to the next step.
10. Specify the version number of the generated application.
11. Go to the next step.

12. This step is used to include specific files (data files, images, ...).

- Click the "Add" button.
- Select the data files (.fic, .ndx and .mmo) found in the "EXE\iOS application" directory of project. The list of files is displayed.



13. Indicate that these data files must be in write mode: check the "Write" box.

14. Go to the next step.

15. The wizard allows you to specify:

- whether the application is allowed to be resized or not (Split View). The option allows the end user to display two windows of two different applications side by side on some devices.
- whether files can be shared with iTunes. If this option is checked, you will have the ability to retrieve the application files on Mac during the synchronization. For example, if data files have been supplied with the application, the iTunes application will allow you to retrieve these files.
- the minimum version of iOS required to run the application.



16. Go to the next step. This step is used to specify the signature options for Xcode. See the online help for more details.

17. Validate the wizard.

18. The generation is performed in the EXE folder of project directory. The directory containing the source code that will be used on Mac is named "Project\_name.xcode.gen" ("WM Managing Products.xcode.gen" in our example). This is the directory that must be copied onto Mac. Click the "Open the generation directory" button.

The other operations must be performed on Mac. You must:

- Transfer the WINDEV Mobile project onto Mac.
- Compile the project in Xcode.

### Transferring the WINDEV Mobile project onto Mac

► To transfer the WINDEV Mobile project onto Mac:

1. Copy the entire folder generated in the EXE directory onto an external media (USB key, external hard disk, shared directory with Mac on network). This directory is named <Project Name>.xcode.gen.
2. Paste this folder on the Mac that will compile the application.
3. Open the folder on Mac and open the file named "Project\_name.xcodeproj".
4. The project is opened in Xcode.

### Compiling the project in Xcode

WINDEV Mobile automatically generates an Xcode project for your iOS compilations. To simplify the implementation of applications, WINDEV Mobile generates a "Scheme" for Xcode.

► To compile the project in Xcode:

1. In the drop-down list found in the top left corner, select the compilation options. Select the scheme corresponding to your application then the compilation target (device currently connected or a simulator).
2. To start the compilation, select "Product .. Clean" then "Product .. Build".
3. A status report of compilation is displayed at the top ("Succeeded", otherwise the number of warnings and errors). You have the ability to click these symbols to see the list of errors/warnings.
4. Once the program is compiled without error, you can start the simulation ("Product .. Run"). The simulation window appears with the application.

Then, you have the ability to run the test of your application on your iPhone or iPad.



# Windows Mobile application





## LESSON 6.1. MY FIRST WINDOWS MOBILE PROJECT

**This lesson will teach you the following concepts**

---

- Creating Windows Mobile project.
- My first window.
- My first test.
- First deployment.



Estimated time: 1 h

## Overview

To start developing with WINDEV Mobile for a Windows Mobile platform, we are going to create a first project. This project will contain a window used to display a message.

This first example will present the main concepts of development for Windows Mobile with WINDEV Mobile.



### Caution

The setup of a Windows Mobile application can be created with a 32-bit WINDEV Mobile editor only.

## My first project

### Creating the project

We are going to create our first project for Windows Mobile. If you own the mobile device (Smartphone or Pocket PC) on which the application must be run, we advise you to connect this device to the development computer. Therefore, the device characteristics will be automatically detected and proposed when creating the Windows Mobile project.

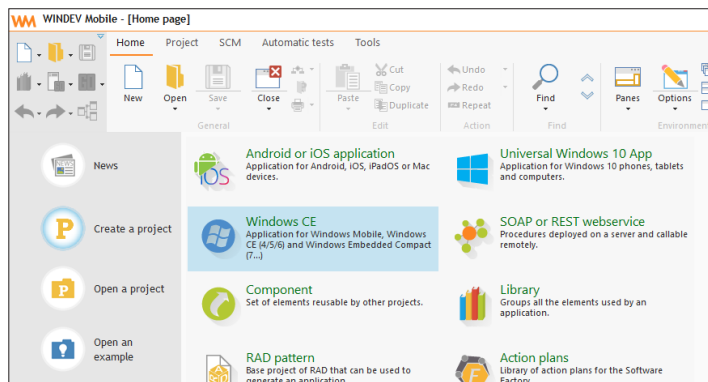


### Answer

A corrected project is available. To open this project, in WINDEV Mobile's home page (Ctrl + <), click "Tutorial" and select "My Pocket project (Answer)".

► To create a project:

1. Start WINDEV Mobile 25 (if necessary).
2. Display the WINDEV Mobile home page if necessary (Ctrl + <).
3. In the home page, click "Create a project" then "Windows CE".




The project creation wizard starts. The different steps of the wizard help you create your project. The information specified in this wizard can be modified later.



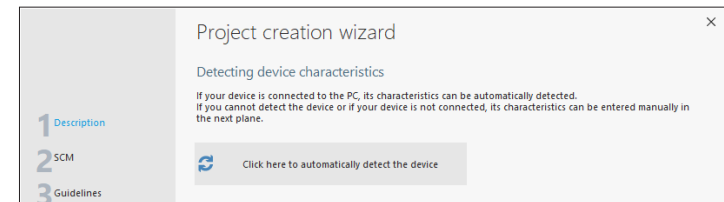
### Remark

Other method for creating a project:

1. Click  among the quick access buttons of WINDEV Mobile menu.
2. The window for creating a new element is displayed: click "Project".

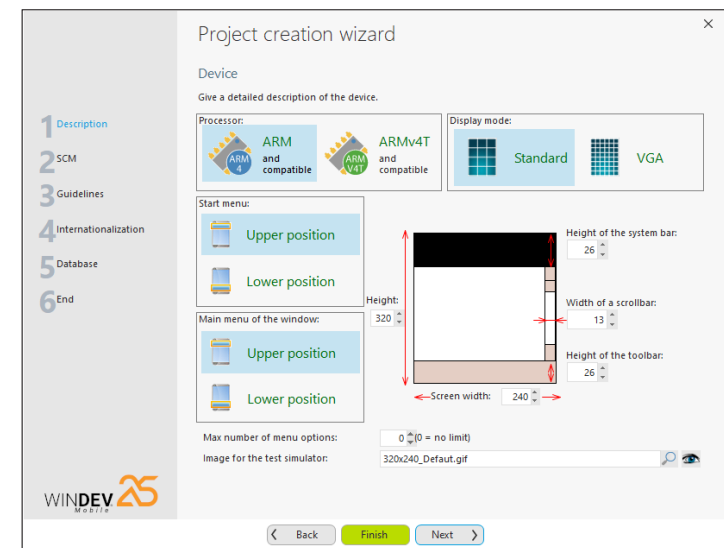
4. The wizard proposes to create a blank project or a project based on an example. Select "Create a blank project" and go to the next step.

5. The next step is used to detect the parameters of device connected to the PC.



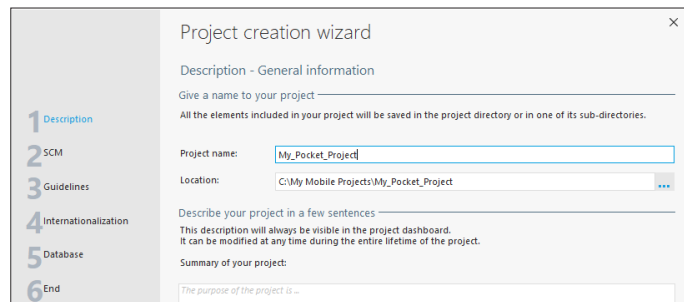
- If your device is connected, click "Click here to automatically detect the device".
- If no device is connected, go to the next step to manually define the characteristics of device used.

6. The next step is used to describe your device.



Go to the next step.

7. The wizard proposes to type the name of project, its location and its description. In our case, this project will be named "My\_Pocket\_Project". WINDEV Mobile proposes to create this project in the "\\My Mobile Projects\\My\_Pocket\_Project" directory. You can keep this location or modify it via the [...] button.



8. Go to the next step ("Next" button).

9. The wizard proposes to add documents. Go to the next step.

10. This step allows you to use the SCM (Source Code Manager). We will not use this option in this tutorial. Click on "No, do not use SCM".

11. In the left section of the wizard, click "Guidelines". This step is used to define the code style. Don't modify the suggested options. Go to the next step.

12. This step is used to define the style book. Select "Elegant".

Remark: This skin template can be found in "Obsolete skin templates for Windows Mobile and others".

13. The other wizard steps not being important for our first project, click "End" in the left section of wizard.

14. Click on "Finish" at the bottom of the wizard. The project is automatically created.

15. The window for creating a new element is displayed. This window is used to create all elements that can be associated with a project.

## My first window

### Overview

The first window allows the user to display a welcome message via a "Display" button.

You may think this is too basic but we advise you to create this window. You may be surprised by how intuitive and how easy it is to use the editor of WINDEV Mobile. Furthermore, this window will allow you to discover concepts that are fundamental for the rest of this tutorial and to see the entire process for developing a Windows Mobile application with WINDEV Mobile.

### Creating the window

► To create the window:

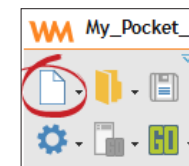
1. In the window for creating a new element, click "Window" then "Window".



Remark

As a new project was created, the window for creating a new element is automatically displayed.

To display the window for creating a new element, all you have to do is click  among the quick access buttons of WINDEV Mobile:



2. The window creation wizard starts.

3. In the list of proposed windows, select "Blank". The skin template used is displayed at the bottom right of wizard. The "Elegant" skin template that was selected when creating the project is selected by default.



Remark

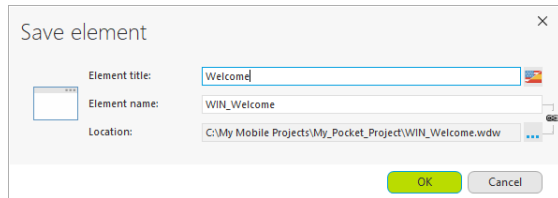
Skin templates allow you to quickly create outstanding interfaces. A skin template defines the window style as well as the style of all controls that will be used in this window. Thus, there is no risk of obtaining an ugly interface.

4. Validate. The window is automatically created in the editor. The window for saving an element is displayed. This window displays:

- the element title. For a window, this title will be displayed in the title bar of window.
- the element name that corresponds to the window name. This name will be used in programming.
- the element location. This location corresponds to the directory in which the physical file corresponding to the element is saved. The window is a "WDW" file, saved in the project directory.

5. Specify the name of the element: "Welcome" (here, the element corresponds to the WINDEV Mobile window).

6. The element name ("WIN\_Welcome") is automatically proposed.



## Remark

Let's take a look at the window name proposed by WINDEV Mobile: this name starts with the letters "WIN\_". This prefix is automatically added because the project is using a code style.

The code style is used to define a prefix for each type of object, allowing you to quickly identify the element:

- a window starts with "WIN\_",
- a button starts with "BTN\_",
- etc.

If you do not want to use this code style, you can simply disable it: on the "Project" pane, in the "Other actions" group, expand "Code style" and uncheck "Use code style".

7. Validate by clicking "OK".

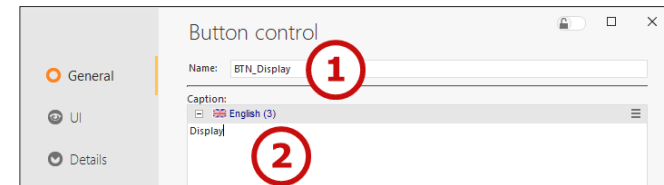
## Displaying a message

You are now going to create a Button control to display a message.

► To create the "Display" Button control:

1. On the "Creation" pane, in the "Usual controls" group, click **Ok**. The control appears in creation mode under the mouse pointer.
2. Move the mouse toward the position where the control will be created in the window (at the top of window for example). To drop the control in the window, simply click again.
3. Right-click the control that was just created. The popup menu of control is displayed. Select "Description" from this popup menu. The description window of the Button control appears.

► Modify the control characteristics by entering the following information:



1. This control is named: "BTN\_Display".

2. The control caption is: "Display"

► Validate the control description window ("OK" button). The control appears in the window editor.

► We are going to display a message in a dialog box (a small window proposed by the system). To do so, we will be using our first WLanguage function: **Info**.



## Remark

The programming language supplied with WINDEV Mobile is named WLanguage. It is a 5th-generation language (5GL) that includes highly sophisticated commands.

1. Select the control if necessary.

Remarks:

- When the control is selected, several handles appear around the control.
- To select a control, simply click on it.

2. Display the popup menu of control (right mouse click).

3. Select "Code". This option opens the code editor of WINDEV Mobile, where all WLanguage statements can be typed.

4. Write the following code in the "Click" event of the "BTN\_Display" control:

```
Info ("Hello")
```

**Note about the assisted input:** As soon as the first two characters are typed, WINDEV Mobile proposes all words of WLanguage vocabulary containing these characters. The assisted development is a very powerful feature. No more mistakes when entering the name of an element: the syntax errors are reduced. All you have to do is select the requested word and press Enter to validate. You can focus on the algorithm.




## Remark

When writing this code in the code editor, you may have noticed that the different elements use different colors. This is the syntactic coloring. The code editor allows you to easily identify the different elements handled by the code:

- the WLanguage functions are colored in blue,
- the character strings (between quotes) are colored in purple,
- the names of controls are colored in cyan.


These colors can be modified element by element in the options of code editor (on the "Home" pane, in the "Environment" group, expand "Options" and select "Options of the code editor").

**Info** displays the message passed in parameter.

- Save the modifications by clicking  among the quick access button (on the left of ribbon) or by pressing Ctrl + S.
- Close the code window (cross in the top right corner of code editor). The window reappears in the window editor.

## First test

For a Windows Mobile application, WINDEV Mobile allows you to run the application test on the development computer via the simulation mode. This test simulates a Windows Mobile device on the development computer. This test is useful when no Windows Mobile device can be used by the developer. However, this test does not allow you to use the hardware components of device (SMS, ...).

- We will now run the window test in simulation mode.
  1. Click  among the quick access buttons (or press F9).
  2. Validate (if necessary) the information message regarding the simulator mode.
  3. The created window is started in execution. The simulator shell corresponds to:
    - the device connected to the development computer,
    - the device chosen in the wizard for project creation.
  4. Click the "Display" button.



5. Validate the system window that is displayed.
- Any developer knows that running a program test can be a long and tiresome job. In WINDEV Mobile, a SINGLE CLICK allows you to run the test of window, report or procedure while you are creating it. This is both simple and fast!
  - Click the "x" button found in the simulator shell to close the window.
  - The WINDEV Mobile editor appears again.


## First deployment on the device

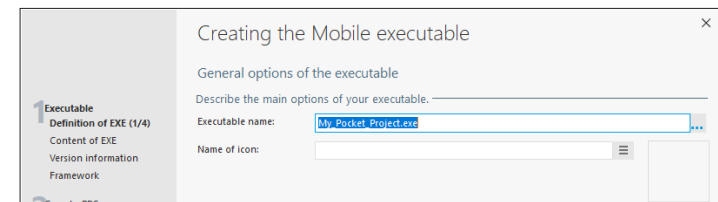
### Principle

To run the application in stand-alone mode on the mobile device, you must:


- Connect the device via a USB port.
- Generate the application.
- Choose to copy and start the executable on the connected mobile. Copying the application can take several seconds.

### Implementation

- To generate the Windows Mobile application:
  1. On the "Project" pane, in the "Generation" group, click "Generate" (you can also click  among the quick access buttons).
  2. WINDEV Mobile proposes to select the first project window. In our example, select "WIN\_Welcome" and validate ("OK" button).
  3. The wizard for creating a mobile executable starts.
  4. The first wizard step is used to define the application name and icon.




The executable icon can be chosen in the image catalog of WINDEV Mobile:

- Click on  on the right of the control "Name of icon".
- Select "Catalog" from the popup menu that is displayed.
- The window of image catalog is displayed.



#### Remark

As soon as an image can be used (in a control, window, report, ...), the "Catalog" button is available (via the menu displayed by the  button). This allows you to select an image among the images supplied in the image catalog of WINDEV, WEBDEV and WINDEV Mobile.

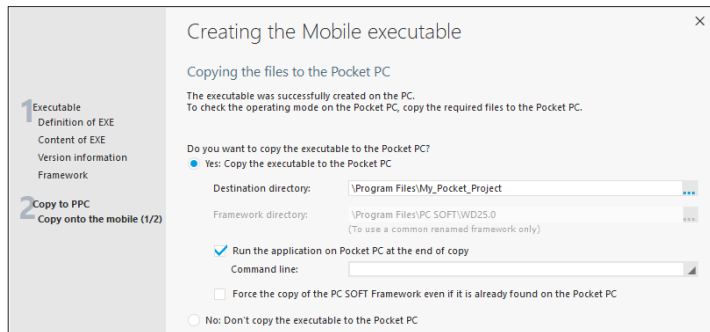
To perform a search in the image catalog:

- specify the keyword corresponding to the search,
- validate. The images found are automatically displayed.

By double-clicking the requested image, this one is generated and included in your project.

5. The other steps are not required by our application. Click the "2- Copy onto the mobile" link found on the left of the wizard.

6. This step is used to define the options for copying files onto the mobile device:



Caution

The setup of a Windows Mobile application can be created with a 32-bit WINDEV Mobile editor only.

7. The options to choose depend on your configuration:

- If a mobile device is connected, select "Yes: Copy the executable to Pocket PC" as well as "Run the application on Pocket PC at the end of copy". In this case, once the executable is generated, the application will be automatically copied and started on the mobile.
- If no mobile is device connected, select "No: Don't copy the executable to Pocket PC". In this case, the application can be deployed on the mobile devices via a setup procedure.

8. Validate and end the wizard.

That's it, our first application is generated and run on the Windows Mobile device.

## LESSON 6.2. DATABASES

This lesson will teach you the following concepts

- Available databases.
- Synchronization.



Estimated time: 30 mn

## Format of databases

A Windows Mobile application can handle data. The format of these databases can be:

- **HFSQL** (in Classic or Client/Server mode), database system supplied with WINDEV Mobile.
- **CEDB**, database system that can be used on mobile devices (Pocket PC).
- **AS/400**, AS/400 database that can be used on mobile devices (Pocket PC).
- ...

### HFSQL database

#### HFSQL Classic

In HFSQL Classic mode, the data files are stored on the device.

In this case, the application is stand-alone. No Wi-Fi or 3G connection is required.

The data is stored in the device memory. The maximum storage size depends on the amount of memory on the device.

#### HFSQL Client/Server

In HFSQL Client/Server mode, no data is stored on the device. The data is stored on a computer on which a HFSQL server is installed.

To access this computer (and therefore the database), a method for communicating with the server must have been enabled in the mobile application (Wi-Fi or 3G) in order to connect via the network or Internet.

The response times depend on the quality of Wi-Fi or Internet network and on the amount of requested data.

The access to the data will be performed by the Hxxx functions of WLanguage and/or by SQL queries.



#### Example

The "Pocket Notes" and "Pocket Telephony" examples (supplied with WINDEV Mobile) are using HFSQL data files. These examples are accessible from the WINDEV Mobile home page (Ctrl + <).



#### Remark

During the test (in simulation mode) of a WINDEV Mobile application that handles HFSQL data files, the data files used are the ones found on the **PC**.

## CEDB

The CEDB format is a database format that can be used on the mobile devices (Pocket PC).

A CEDB database corresponds to a ".CDB" file. A CEDB database can contain several data files (also called "tables").

Two types of CEDB databases are available:

- **the standard CEDB databases**, that correspond to the databases found by default on the mobile device. These databases contain the following data files: "Tasks", "Contacts", "Appointments", ...
- **the other CEDB databases** (called custom databases), that correspond to the Access databases (".MDB" file) previously exported from a PC.

Remark: When an Access database (".MDB" file) is copied onto a mobile device (via the file explorer), this database is automatically changed into a CEDB database (".CDB" file).

A CEDB database can be handled:

- from a WINDEV Mobile application.
- from a WINDEV application.

These operations are performed by the **cdbXXX functions** of WLanguage.



#### Remark

**Caution:** From Windows Mobile 5 onwards, a standard database (tasks, contacts, appointments, etc.) can no longer be accessed from a Window application (and therefore from a standard WINDEV application). CEDB databases can be accessed from the PC, from the simulator, from the mobile device. Standard databases can be accessed from the mobile device only.



#### Remark

**Caution:** The structure of CEDB databases is not adapted to the process of large data amount. Therefore, we recommend that you use HFSQL databases. Furthermore, HFSQL allows you to benefit from all features available in WINDEV Mobile (RAD, file link, ...).



#### Remark

During the test (in simulation mode) of a WINDEV Mobile application that handles a CEDB database, the database used is found on **Pocket PC**.

## AS/400

This database format is accessible via a Native Connector (also know as Native Access) by Windows Mobile applications. To use this Native Connector, a module is required in addition to WINDEV Mobile.

Contact PC SOFT Sales Department for more details.



## Sharing data between two applications

A WINDEV Mobile application for Windows Mobile can share data with a standard WINDEV application.

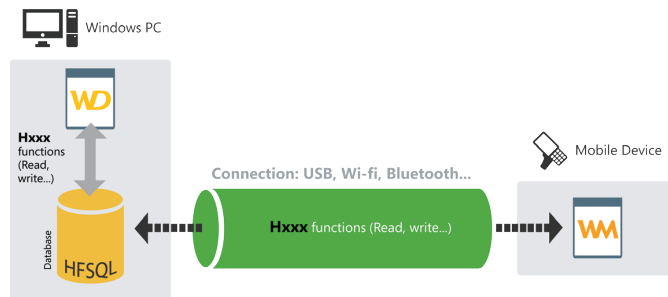
You have the ability to use:

- a standard WINDEV application used to handle the entire database.
- a WINDEV Mobile application used to handle the entire database or part of this database.

When two applications share the same data, the data files can be managed according to two different methods:

- **Handling the same data files:**

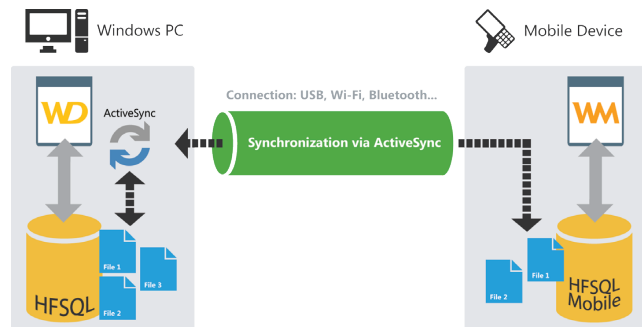
The two applications handle the same data files. These data files are found on the PC. The WINDEV Mobile application accesses the data files by Wi-Fi, by infrared, by GPRS, ... **HSubstDir** allows you to specify the data directory to use.



For example: application for taking orders in a restaurant. The new orders are automatically sent to the database found on the PC.

- **Copying data files on the mobile device:**

All data files (or some of them) are copied onto each mobile device (Pocket PC for example) beforehand. Each application handles its own files. To take into account the modifications performed in each application, the data files must be synchronized (automatically or not).



For example: poll application performed in the street. The answers will be available in the WINDEV application once the data files have been synchronized.

## Handling the same data files

To allow the WINDEV Mobile application to access the data files found on the PC:

- the mobile devices must have network access (Ethernet card, Wi-Fi, etc.)
- the data found on the PC must be accessible in read/write via a UNC path (the directory used must be a shared directory).

Then, the data can be handled (addition, modification and deletion) by the HFSQL functions.



### Example

#### Databases in HFSQL format (on the Pocket PC and on the PC)

The "Network tasks" example (supplied with WINDEV Mobile) includes a project that can be used on a Windows Mobile device (Pocket Network tasks) and a project that can be used on PC (PC Network tasks). These two examples use the data files found on the PC.

## Copying the data files onto the mobile device (Pocket PC for example)

To update the data files found on the PC with the data typed on the mobile devices, all you have to do is synchronize the files.

If the data files used are in HFSQL format, all mobile devices must be connected one by one to the PC. The automatic HFSQL synchronization via ActiveSync takes everything in charge.

If the data files used are not in HFSQL format, you must program the synchronization between the WINDEV Mobile application and the WINDEV application. See the examples supplied with WINDEV Mobile for more details.



### Example

#### Databases in HFSQL format (on the mobile device and on the PC)

The "Sending SMS", "Managing lists of purchases" and "Stocks" examples (supplied with WINDEV Mobile) include a project that can be used on mobile device (Pocket PC for example) and a project that can be used on PC. These examples present the synchronization of data typed in the two projects.

## LESSON 6.3. ADVANCED PROGRAMMING

### This lesson will teach you the following concepts

- Creating a management application.
- Handling the database.



Estimated time: 1 h

### Overview

In this lesson, we are going to develop a Windows Mobile application that is using a HFSQL Classic database.

This application will allow us to present some specific features of the programming for Windows Mobile.

### Opening project

- Start WINDEV Mobile 25 (if not already done).
- Display the WINDEV Mobile home page if necessary (Ctrl + <).
- Open the "Pocket Managing Products" project.  
To do so, in the home page, click "Tutorial" and select "Pocket Managing Products (Exercise)".




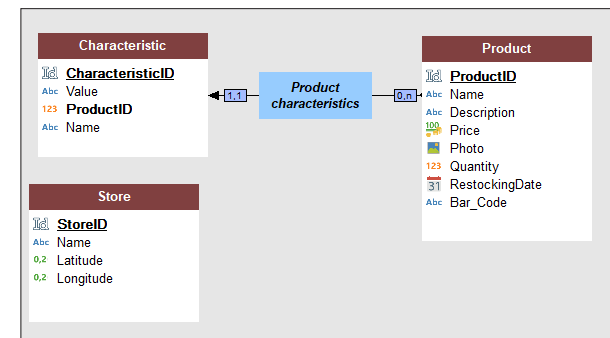
Answer

A corrected project is available. This project contains the different windows created in this lesson. To open the corrected project, in WINDEV Mobile's home page, click "Tutorial" and select "Pocket Managing Products (Answer)".

### Project description

Let's take a look at our initial project. This project is an existing project. It contains no window. It only contains the analysis describing the HFSQL Classic data files that will be used. In the EXE directory, the corresponding data files are supplied with data in order to run the different tests.

- To view the analysis associated with the project:
  1. Click  among the quick access buttons of WINDEV Mobile menu.
  2. The data model editor is displayed.




3. This analysis includes 3 data files:
  - A "Product" data file, which contains the product description: name, price, quantity, ...
  - A "Characteristic" data file, which contains the different product characteristics. For example, if the product is a tee-shirt, its characteristics will correspond to the size, the color, ... The "Characteristic" data file is therefore linked to the "Product" data file.
  - A "Store" data file, which contains the GPS coordinates of each store.
4. Close the data model editor (click the cross at the top right of editor).

We are now going to develop our application.

## Displaying the list of products

We are going to create a window used to list the different products. These products will be displayed in a "Table" control.

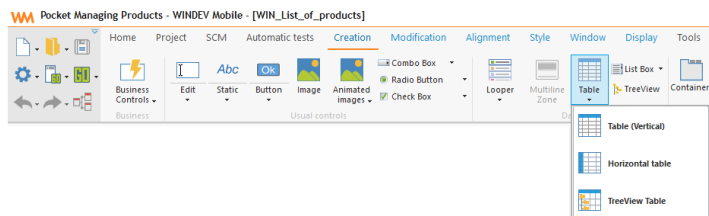
## Creating the window

- To create a new window:
  1. Click  among the quick access buttons. The window for creating a new element is displayed: click "Window" then "Window".
  2. In the wizard, display the "Standard" tab, choose "Blank" and validate.
  3. The window for saving an element is displayed.
  4. In the window that is displayed, type the window title: "List of products". The window name and location are automatically filled. Keep the default values and validate ("OK" button).
  5. The window is added to the project.

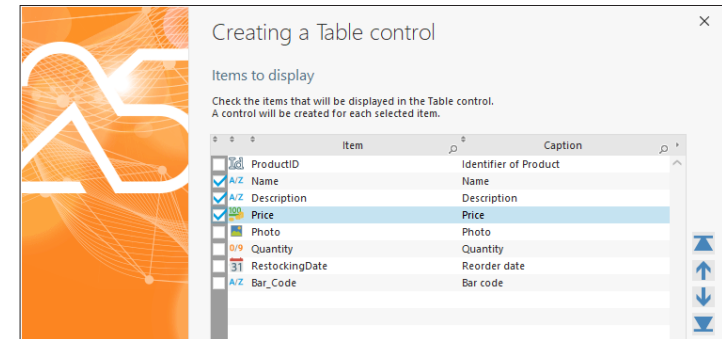
## Creating the Table control

To display the list of products, we are going to use a "Table" control. The main information about the products will be displayed in this control.

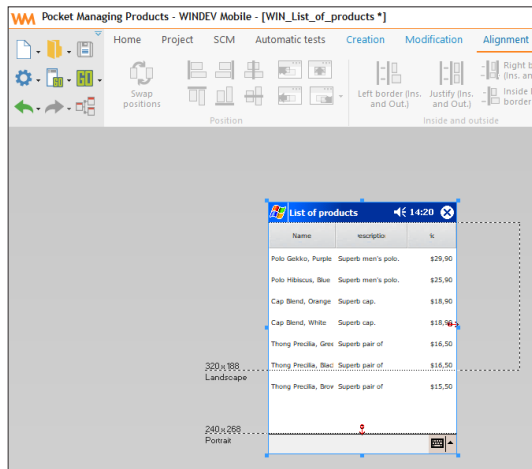
1. On the "Creation" pane, in the "Data" group, expand "Table" and select "Table (Vertical)".





2. The control appears under the mouse.
3. Click the position where the control must be created in the window (at the top for example). The wizard for creating a Table control starts.
4. In the wizard, select "Display the data found in a file or in an existing query". Go to the next step.
5. Select the "Product" data file. Go to the next step.
6. Select the items to display:

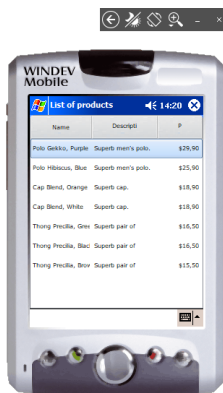


7. Keep the "Name", "Description" and "Price" items (a checkmark must be found in front of these items). Go to the next step.
8. Keep the proposed sort item (ProductID). The products will be sorted in the Table control according to this item. Go to the next step.
9. In the "Additional parameters" step, keep the default options. Go to the next step.
10. Select the "Vertical" orientation and go to the next step.
11. Keep the default name ("TABLE\_Product") and validate.
12. The Table control appears in the window editor.
13. Resize the control and its columns via the handles in order for the content to be displayed in the window.

**Remark**

Data automatically appears in the window displayed in the editor. This concept is called "Live Data": you see the data of your data files in real time! This feature is very useful to adapt the size of controls to their content.

- Save the window by clicking  among the quick access buttons.
- We are going to run a first test in the simulator to see the result. Click  among the quick access buttons (or press F9).




- Close the simulator to go back to the window editor.

## Creating the form window

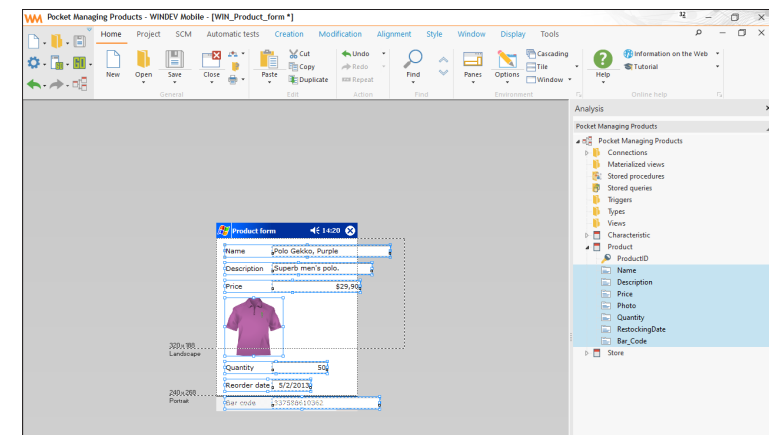
We are now going to create a new window used to display the product form. Then, this window will be started from the list of products to display the details of selected product.

### Creating the window

- To create the form window:
  1. Create a new blank window. Click  among the quick access buttons. The window for creating a new element is displayed: click "Window" then "Window".
  2. In the wizard that starts:
    - Select the "Standard" tab.
    - Choose "Blank".
  3. Validate the wizard.
  4. The window for saving an element is displayed. Specify the window title: "Product form". Its name is automatically proposed: "WIN\_Product\_form". Validate.

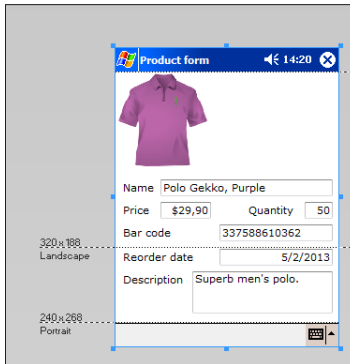
### Creating controls

- To create an edit control:
  1. Display the "Analysis" pane if necessary (on the "Home" pane, in the "Environment" group, expand "Panels" and select "Analysis"). The different data files described in the "Pocket Managing Products" analysis appear in the pane.
  2. With the mouse, select the items of the "Product" data file displayed in the pane (except for the "ProductID" item).
  3. Drag and Drop these items to the window that was just created.



4. Resize the controls ("Name", "Bar code", "Reorder date" and "Description") so that they become visible in the window:
  - Select the requested control.
  - Use the sizing handles (blue squares) to resize the control.


## 5. Reorganize the controls in the window.



Respect the following order: "Photo", "Name", "Price", "Quantity", "Bar code", "Reorder date", "Description".

6. We are going to view the tab order in the window: press the F5 key. The numbers that are displayed represent the tab order in the window. Press the F5 key again to make the numbers disappear. The tab order is automatically adapted to the order of controls in the window.

7. Save the window (Ctrl + S).

► Run the window test (  among the quick access buttons). The window is displayed with empty controls.

► Close the test window to go back to the editor.

► To display the product data:

1. Display the WLanguage events associated with the window:
  - Perform a right mouse click in the area beside the window.
  - Select "Code" from the popup menu that is displayed.
  - The code editor appears.

2. Enter the following code in the event "End of initialization of WIN\_Product\_form":

```
FileToScreen()
```

**FileToScreen** is used to display in the controls the data found in the data file, for the current record.

3. Close the code window (click X at the top right corner of code editor).

4. Save the window.


## Displaying the form from the list of products

Let's now see how to display the form of selected product in the list of products.

► Perform the following operations:

1. Position on the "List of products" window: click the "WIN\_List\_of\_products" button in the open documents bar:



2. On the "Creation" pane, in the "Usual controls" group, click : the shape of the control appears under the mouse pointer. Then, click at the bottom of the window to create the Button control.

3. Select the control and press Enter. The control caption becomes editable. Type "Modify the element" and press Enter on the keyboard.

4. If necessary, resize the control (using the handles) so that the caption is fully displayed.

5. Right-click the control and select "Code" from the popup menu.

6. In the code window that appears, write the following code in the "Click" event:

```
Open(WIN_Product_form)
TableDisplay(TABLE_Product, taCurrentSelection)
```




## Remark

The assisted code input is going to help you: as soon as you type the opening bracket "(", a drop-down list proposes the name of all existing windows found in the project. All you have to do is select the window with the keyboard or with the mouse.

If the window name is not displayed in the list, it means that this window was not saved beforehand.

7. Save the modifications by clicking  among the quick access buttons.

8. Close the code window (click X at the top right corner of code editor).

- Run the window test again in the simulator (  among the quick access buttons).
- In the list of products, click one of the products with the mouse.
- Click the "Modify the element" button.
- The detailed product window is displayed.

► Close the simulator.

## Managing the product creation and modification

We are now going to modify our two windows in order to manage the product addition and modification.

### Modifying the product form

We are going to add a button into the "WIN\_Product\_form" window: a "Validate" button to manage the validation of modifications.



#### Remark

In the applications for Windows Mobile, the use of a "Cancel" button is not required. Indeed, the user can just click the cross (top right of the window) to cancel the input performed.

► To create the "Validate" Button control in the "WIN\_Product\_form" window:

1. Display (if necessary) the "WIN\_Product\_form" window in the editor: click the corresponding button in the open documents bar.
2. On the "Creation" pane, in the "Usual controls" group, click **Ok**: the shape of the control appears under the mouse pointer. Then click in the window to create the control (next to the Image control, for example).
3. Select the control and press Enter. The control caption becomes editable. Type "Validate" and press Enter on the keyboard.




4. If necessary, resize the control (using the handles) so that the caption is fully displayed.


► We will now write the WLanguage code associated with the "Validate" Button control.

1. Right-click on the control and select the "Code" option from the popup menu.
2. Write the following code in the "Click" event:

```
ScreenToFile ()
HModify (Product)
Close ()
```

Let's study this code:

- **ScreenToFile** is used to initialize the items with the values of linked controls, for the current record.
  - **HModify** is used to update the file data for the current record.
  - **Close** is used to close the form and to go back to the "WIN\_List\_of\_products" window.
3. Save the modifications by clicking  among the quick access buttons.
  4. Close the code window (click X at the top right corner of code editor).

► Display the "WIN\_List\_of\_products" window in the window editor and run its test in the simulator (  among the quick access buttons).

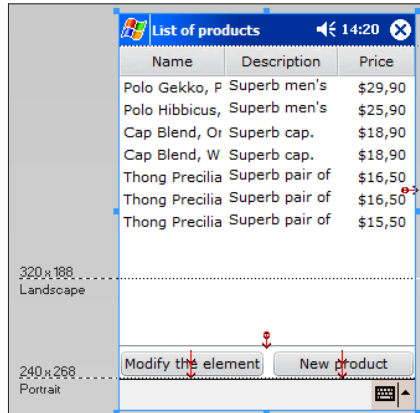
- In the list of products, click one of the products with the mouse: for example, the "Polo Hibbicus Blue" product at a price of 25.90 Dollars.
- Click "Modify the element".
- The detailed window of product is displayed. Modify the price of 25.90 Dollars and type 19.90 Dollars, then click the "Validate" button.
- When going back to the list of products, you will notice that the price was updated for this product.

► Close the simulator. The WINDEV Mobile editor is displayed.

### Creating a new product

The principle for creating a product is as follows:

- In the window for the list of products, we are going to add a "New product" button that will be used to open the "Product form" window.
  - Then, we will modify the code of "Product form" window to manage the addition into the Product data file.
- First of all, display (if necessary) the "WIN\_List\_of\_products" window in the editor: click the corresponding button in the button bar.
- To create a new Button control:
1. On the "Creation" pane, in the "Usual controls" group, click **Ok**: the shape of the control appears under the mouse pointer. Then click in the window to create the control (for example next to the Button control "Modify the element").
  2. Select the control and press Enter. The control caption becomes editable. Type "New product" and press Enter on the keyboard.



3. If necessary, resize the Button control (using the handles) so that the caption is fully displayed.

4. The WLanguage code of this Button control will allow you to open the "Product form" window and to reset its controls. To write this code:

- Right-click the "New product" Button control.
- Select "Code" from the popup menu that is displayed.
- Write the following code in the "Click" WLanguage event:

```
HReset (Product)
Open (WIN_Product_form)
TableDisplay (TABLE_Product)
```

- **HReset** initializes the item variables in the Product data file with the default values to manage a new record.
- **Open** is used to open the form window.
- **TableDisplay** is used to update the content of Table control: the new record will be displayed in the control.

5. Save the modifications by clicking  among the quick access buttons.

6. Close the code window (click X at the top right corner of code editor).

► We are now going to modify the window of product form to manage the addition of a new record.

1. Open the "WIN\_Product\_form" window in the editor.

2. We will modify the code of the Button control used to validate the form:

- Right-click the Button control and select "Code" from the popup menu.
- In the "Click" event, replace the existing WLanguage code with the following:

```
ScreenToFile ()
IF Product..NewRecord THEN
    HAdd (Product)
ELSE
    HModify (Product)
END
Close ()
```

Let's take a look at these lines of code:

- **..NewRecord** is used to find out whether the current record must be created.
- If **HReset** was called beforehand, the property returns **True** (click on "New product") and the record must be created by **HAdd**.
- Otherwise, the current record already exists and it must be modified by **HModify**.




Remark

The test code of new record can be replaced by **HSave**. This function is used to check whether the record is already found in the data file, and it allows you to add it or to modify it. The code becomes:

```
ScreenToFile ()
HSave (Product)
Close ()
```

3. Save the modifications by clicking  among the quick access buttons.

4. Close the code window (click X at the top right corner of code editor).

► Display the "WIN\_List\_of\_products" window in the window editor and run its test in the simulator (  among the quick access buttons).

- In the list of products, click the "New product" button.
- Type a new product.
- Validate. The new product is displayed in the list of products.
- Close the simulator.




## Creating a menu window

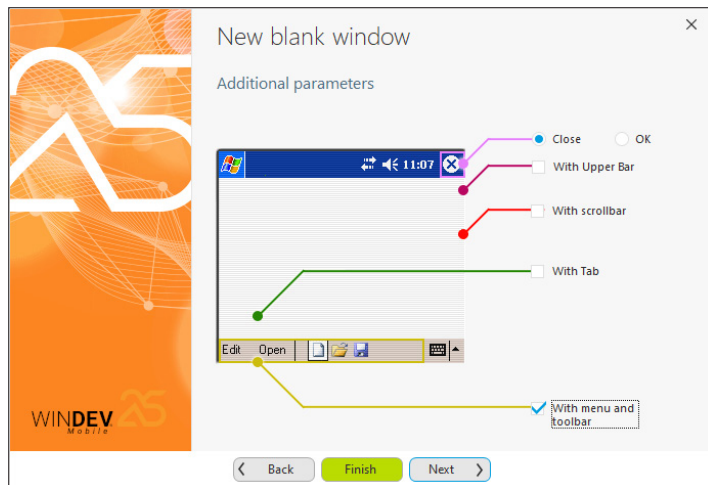
We are now going to create a window for implementing the application menu.

### Creating the window

First of all, we are going to create a blank window with a menu.

► To create the window:

1. Create a new blank window. Click  among the quick access buttons. The window for creating a new element is displayed: click "Window" then "Window".
2. In the wizard for window creation, click the "Standard" tab.
3. Choose "Wizard" and validate.
4. The wizard for creating a Windows Mobile window starts.
5. Select "Maximized" and go to the next step.
6. In the advanced settings, select "With menu and toolbar".



Go to the next step.

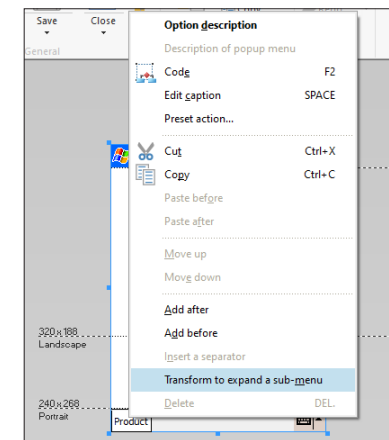
7. Specify the window title: "Menu".
8. Validate the wizard. The window is displayed in the editor.
9. The window for saving an element is displayed. The name entered in the wizard is automatically proposed in the save window. Validate.

## Customizing the menu

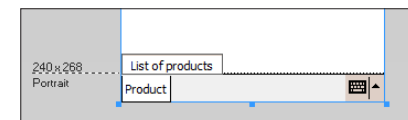
We are going to modify the menu in order to call the window of list of products and to exit from the application.

► To modify the menu option:

1. Right-click the "Menu" option to display the popup menu.  
Remark: This "Menu" option is positioned at the bottom left of window.
2. Select "Option description". The description window is displayed.
3. Modify the caption of the option that becomes "Product". Validate the description window.
4. Select the menu option again.
5. Display the popup menu (right click):



6. Select "Transform to expand a sub-menu".
7. In the input area that is displayed, type "List of products" and validate.



8. Right-click the new option and select "Add before" from the popup menu.
9. In the input area that is displayed, type "Exit" and validate.

## Programming the menu

We are now going to type the code of these two menu options.

► To define the WLanguage code of "Exit" option:

1. In the window editor, display the "Exit" option. All you have to do is expand the menu, just like you did at run time.
2. Display the popup menu of "Exit" option (right mouse click) and select "Code".
3. Write the following code:


```
Close()
```

**Close** is used to close the current window. The "Menu" window being the only window, the application is automatically closed.

► To define the WLanguage code of "List of products" option:

1. In the window editor, display the "List of products" option. All you have to do is expand the menu, just like you did at run time.
2. Display the popup menu of the "List of products" option (right mouse click) and select "Code".
3. Write the following WLanguage code:

```
Open(WIN_List_of_products)
```

4. Save the modifications by clicking  among the quick access buttons.
5. Close the code window (click X at the top right corner of code editor).

## Running the application test


The last step consists in specifying that the menu window is the first application window. To do so, we are going to define the first project window and run a full test of project.

► To define the first project window:

1. Select the "WIN\_Menu" window in the "Project explorer" pane.
2. Display the popup menu.
3. Select "First project window". A specific icon (with a small 1) is displayed in front of the window name, in the "Project explorer" pane.

Until now, the test of windows was run individually by clicking  among the quick access buttons.


► To run the project test:

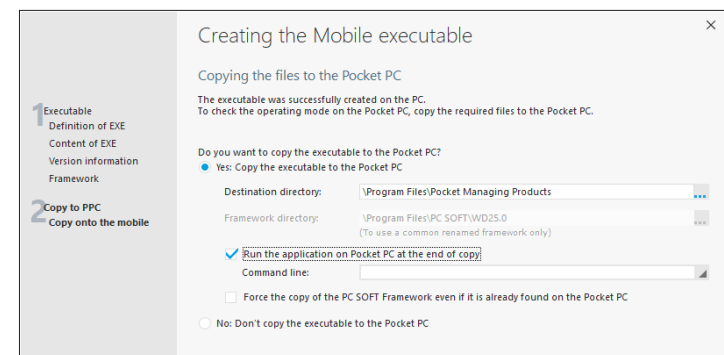
1. Click  among the quick access buttons.
2. Your project starts with the menu window. Click an option of your menu to check whether the different links are correct.

That's it, our application is created, we must now compile it and install it on the device in order to test it in real conditions.

## Creating the executable

► To generate the Windows Mobile application:

1. On the "Project" pane, in the "Generation" group, click "Generate" (you also have the ability to click  among the quick access buttons).
2. The wizard for creating a mobile executable starts.
3. The first wizard step is used to define the application name and icon. The executable icon can be chosen in the image catalog of WINDEV Mobile ("Catalog" button).
4. The other steps are not required by our application. Click the "2- Copy onto the mobile" link found on the left of the wizard.
5. This step is used to define the options for copying files onto the mobile device:

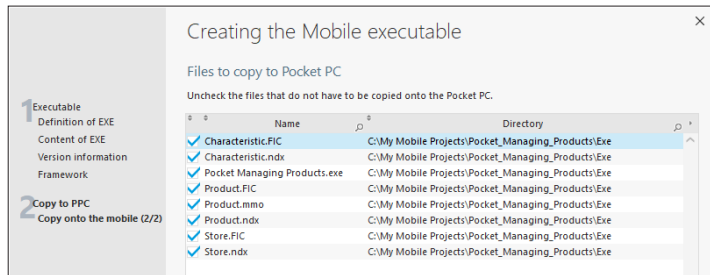


6. The options to choose depend on your configuration:

- If a mobile device is connected, select "Yes: Copy the executable to Pocket PC" as well as "Run the application on Pocket PC at the end of copy". In this case, once the executable is generated, the application will be automatically copied and started on the mobile.
- If no mobile is device connected, select "No: Don't copy the executable to Pocket PC". In this case, the application can be deployed on the mobile devices via a setup procedure.

7. Go to the next step.

8. If the option "Yes: Copy the executable to Pocket PC" was selected, the wizard proposes the files to copy onto Pocket PC:



9. Validate the selection of suggested files.

That's it, your application is generated and run on the Windows Mobile device.

## LESSON 6.4. DISTRIBUTING THE APPLICATION

This lesson will teach you the following concepts

- Available distribution modes.



Estimated time: 20 mn

## Overview

WINDEV Mobile allows you to develop applications for Windows Mobile.

Several methods can be used to install a WINDEV Mobile application on a device running Windows Mobile (Pocket PC for example):

- **setup in CAB format.** This setup program is run on the mobile device directly.
- **setup in MSI format.** This setup program is run on a PC running Windows connected to a mobile device (Pocket PC for example).
- **setup by direct copy** of executable from the PC to the Pocket PC (as already seen in the previous lessons).

These different setup modes are available via the wizard for setup creation.

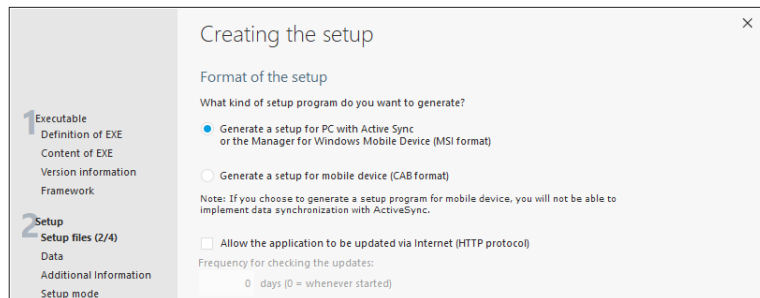


Caution

The setup of a Windows Mobile application can be created with a 32-bit WINDEV Mobile editor only.

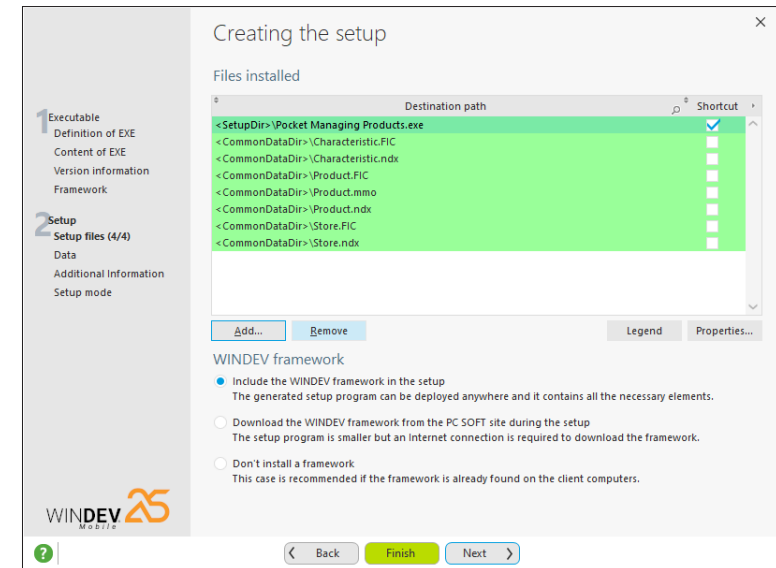
► To start the setup wizard:

1. On the "Project" pane, in the "Generation" group, click "Setup procedure". The wizard for creating the executable and the setup starts.
2. The executable being already created, click "2-Setup" on the left of the wizard.
3. In the message box that is displayed, click "Access the setup options".
4. The description screen of application is displayed. Go to the next step of the wizard ("Next").
5. You can choose the setup mode of your mobile application:



6. For our example, we are going to choose a setup by ActiveSync: check the corresponding option and go to the next step.
7. The wizard asks for the setup directory of application. Keep the default option. Go to the next step.

8. Specify the files to install. The wizard proposes the executable. You also have to install the data files of application. Click the "Add" button and select the HFSQL files (.fic, .ndx and .mmo files) found in the EXE directory of your project.



9. Go to the next step.
10. Uncheck the options for configuring the databases. Go to the next step.
11. The wizard proposes to install additional modules. We won't select any. Go to the next step.
12. The generation directory of setup is proposed.
13. Validate. The setup is generated.
14. WINDEV Mobile proposes to start the setup or to open the generation directory.

## Setup in MSI format

---

### Initial setup

This setup consists in:

- creating the application executable on the development computer via WINDEV Mobile.
- generate the setup program of application on the development computer via WINDEV Mobile. This program corresponds to a ".MSI" file.
- distributing this setup program to the end users.
- running this setup program on PCs. The application will be automatically installed on the mobile device connected to the current computer, via ActiveSync.

Remark: If no mobile device is connected, the setup will be performed during the next synchronization between the PC and the mobile device.

To use this application, start the application on the mobile device (double-click the ".EXE" file for example).

### Update in MSI format

The MSI format proposes two types of updates:

- update by entirely re-installing the application . In this case, the update is identical to an initial setup.
- update by Internet (HTTP protocol) in CAB format. This update consists in:
  - creating the application executable on the development computer via WINDEV Mobile.
  - generate the setup program of application on the development computer via WINDEV Mobile. This setup program corresponds to a ".CAB" file.
  - copying this setup program onto a Web directory by FTP.
  - retrieving and installing the update on the mobile device.

## Setup in CAB format

---

### Initial setup

This setup consists in:

- creating the application executable on the development computer via WINDEV Mobile.
- generate the setup program of application on the development computer via WINDEV Mobile. This setup program corresponds to a ".CAB" file.
- copying this setup program onto the mobile devices of the end users (via a memory card, by GPRS from an Internet site or via the file explorer).
- running this setup program on the mobile devices. This program installs all the files required by the application. At the end of setup, the ".CAB" file is automatically deleted from the device.

To use this application, start the application on the mobile device (double-click the ".EXE" file for example).

### Update in CAB format

The CAB format proposes two types of updates:

- update by entirely re-installing the application . In this case, the update is identical to the initial setup. See the previous paragraph.
- update by Internet (HTTP protocol). This update consists in:
  - creating the application executable on the development computer via WINDEV Mobile.
  - generate the setup program of application on the development computer via WINDEV Mobile. This setup program corresponds to a ".CAB" file.
  - copying this setup program onto a Web directory by FTP.
  - retrieving and installing the update on the mobile device.

# CONCLUSION

The tutorial is over now!

This course covered a range of topics, but not all WINDEV Mobile features, far from it!

You are now familiar with the main concepts.

We recommend that you spend another day exploring the menu options of WINDEV Mobile, for each module.

You can also explore the examples supplied with WINDEV Mobile: some are simple and only address one topic, while others are more complex. These examples will show you the different aspects of WINDEV Mobile as well as the development for the different platforms. Reading the source code is also a good way to learn.

It would take too much time to discuss all available topics (there are hundreds, even thousands!). WINDEV Mobile proposes several features that were not presented in this tutorial:

- sockets and HTTP functions, ...
- queries, queries with parameters, ...
- printing, ...

See the online help for more details.

We wish you great development experiences with **WINDEV Mobile 25**!