



# 10

*Express*

DEVELOP 10 TIMES FASTER



PCSOFT

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 2014: This publication may not be reproduced in part or in whole in any form without the express permission of PC SOFT.

# TABLE OF CONTENTS

## License

---

## Introduction

---

Preliminary points .....	15
Overview of the tutorial .....	15
How do I access the online help? .....	16
Legend of symbols.....	18
If you are familiar with WinDev Mobile 18... ..	19
What is WinDev Mobile used for? .....	19

## PART 1 - OVERVIEW OF WINDEV MOBILE

---

### Lesson 1.1. Discover WinDev Mobile

---

Overview .....	24
Starting WinDev Mobile .....	24
Development environment.....	25
The editor.....	25
The menu bar (ribbon) in details.....	26

## PART 2 - ANDROID APPLICATION

---

### Lesson 2.1. My first Android project

---

Overview .....	32
Necessary configuration for Android .....	32
My first project .....	32
Creating the project.....	32
My first window .....	35
Overview .....	35
Creating the window.....	35
Displaying a message .....	36
First test.....	38
First deployment on the device .....	39
Principle.....	39
Implementation.....	40

### Lesson 2.2. Interface (GUI)

---

Overview .....	43
Choosing the resolution according to the device .....	43
Orientation of the window.....	43
Practical example.....	44

Management of touchscreen .....	45
Practical example .....	46
The different types of controls .....	47

### Lesson 2.3. Databases

The available databases .....	49
Overview .....	49
HFSQL database .....	49
SQLite database .....	49
The synchronization .....	50
Accessing data via a Webservice .....	50

### Lesson 2.4. Advanced programming

Overview .....	52
Opening the project .....	52
Describing the project .....	52
Displaying the list of products .....	53
Creating the window .....	53
Creating the Looper control .....	54
Improving the window .....	56
Creating the form window .....	56
Creating the window .....	56
Creating the controls .....	56
Displaying the form from the list of products .....	58
Managing the creation and the modification of a product .....	59
Modifying the product form .....	59
Creating a new product .....	61
Deploying the application .....	63
Using the camera and displaying an image .....	65
Overview .....	65
Creating the button for taking photos .....	65
Taking photos .....	66
Selecting a photo in the photo album .....	67
Deploying the application .....	67
Managing the bar code of the product .....	68
Overview .....	68
Implementation .....	68
Using the Map control .....	69
Creating the window .....	69
Creating the Map control .....	69
Displaying the stores on the map .....	70
Deploying the application .....	72
Using a Multiline Zone control .....	73
Creating the window .....	74
Creating the Multiline Zone control .....	74
Modifying the Multiline Zone control .....	75
Programming the menu .....	78
Running the test of the application .....	78

## Lesson 2.5. Distributing the application

Overview .....	80
Deployment via Android Market/Google Play .....	80
Deployment via a Web server.....	80
Deployment from the PC via ADB (advanced mode).....	81
Copying the application onto the Mobile device .....	81

## Lesson 2.6. Questions/Answers

Questions/Answers.....	83
How do I use the messaging of the device? .....	83
How do I send an email directly? .....	83
How do I establish the dialog with an FTP server?.....	83
How do I use the GPS of the device and the geo-localization functions? .....	84
How do I send an SMS from my Android application? .....	85
How do I call native JAVA code from my Android application?.....	85

# PART 3 - iOS APPLICATION

## Lesson 3.1. My first iOS project

Overview .....	90
Required configuration for iOS.....	90
My first project .....	91
Creating the project.....	91
My first window .....	93
Overview .....	93
Creating the window.....	93
Displaying a message .....	94
First test.....	96
First deployment on the device .....	97
Principle.....	97
Implementation.....	97

## Lesson 3.2. Interface (GUI)

Overview .....	100
Choosing the resolution according to the device .....	100
Orientation of the window.....	100
Practical example.....	101
Management of touchscreen .....	103
Practical example .....	104
The different types of controls.....	105

## Lesson 3.3. Databases

The available databases .....	107
Overview .....	107
HFSQL database .....	107
The synchronization .....	107
Accessing data via a Webservice .....	108

### Lesson 3.4. Advanced programming

<b>Overview .....</b>	<b>110</b>
Opening the project.....	110
Describing the project.....	110
<b>Displaying the list of products .....</b>	<b>111</b>
Creating the window.....	111
Creating the Looper control .....	111
Improving the window .....	113
<b>Creating the form window.....</b>	<b>114</b>
Creating the window.....	114
Creating the controls.....	114
Displaying the form from the list of products .....	116
<b>Managing the creation and the modification of a product .....</b>	<b>117</b>
Modifying the product form.....	117
Creating a new product.....	119
<b>Using the camera and displaying an image .....</b>	<b>121</b>
Overview .....	121
Creating the button for taking photos .....	121
Taking photos .....	123
Selecting a photo in the photo album .....	123
<b>Managing the bar code of the product.....</b>	<b>124</b>
Overview .....	124
Implementation.....	125
<b>Using the Map control .....</b>	<b>126</b>
Creating the window.....	126
Creating the Map control .....	126
Displaying the stores on the map .....	127
Close the window.....	128
<b>Using a Multiline Zone control .....</b>	<b>129</b>
Creating the window.....	129
Creating the Multiline Zone control .....	129
Modifying the Multiline Zone control.....	130
Programming the menu .....	133
<b>Running the test of the application.....</b>	<b>133</b>
<b>Generating the application .....</b>	<b>134</b>

### Lesson 3.5. Distributing the application

<b>Overview .....</b>	<b>137</b>
<b>Deployment.....</b>	<b>137</b>

### Lesson 3.6. Questions/Answers

<b>Questions/Answers.....</b>	<b>139</b>
How do I use the messaging of the device? .....	139
How do I send an email directly? .....	139
How do I establish the dialog with an FTP server?.....	139
How do I use the GPS of the device and the geo-localization functions? .....	140
How do I display a chart in an iOS application? .....	141
How do I call native Objective C code from my iOS application?.....	141

## PART 4 - WINDOWS STORE APPS APPLICATION

---

### Lesson 4.1. My first Windows Store apps project

---

Overview .....	146
Configuration required for the Windows Store apps applications.....	146
<b>My first project</b> .....	146
Creating the project.....	146
<b>My first window</b> .....	148
Overview .....	148
Creating the window.....	148
Displaying a message .....	150
<b>First test</b> .....	151
<b>First local deployment</b> .....	152
Principle .....	152
Implementation.....	153

### Lesson 4.2. Interface (GUI)

---

Overview .....	156
<b>Orientation of the window</b> .....	156
Practical example .....	157

### Lesson 4.3. Databases

---

<b>The available databases</b> .....	160
Overview .....	160
HFSQL database .....	160
<b>The synchronization</b> .....	160
<b>Accessing data via a Webservice</b> .....	161

### Lesson 4.4. Advanced programming

---

Overview .....	163
Opening the project.....	163
Describing the project.....	163
<b>Displaying the list of products</b> .....	164
Creating the window.....	164
Creating the Looper control .....	164
Improving the window .....	166
Creating a button to close the window .....	166
<b>Creating the form window</b> .....	167
Creating the window.....	167
Creating the controls.....	168
Displaying the form from the list of products .....	169
<b>Managing the creation and the modification of a product</b> .....	170
Modifying the product form.....	170
Creating a new product .....	172
<b>Using the camera and displaying an image</b> .....	173
Overview .....	173
Creating the button for taking photos .....	173
Taking photos .....	175

Running the test of the application.....	175
Generating the application .....	176

### Lesson 4.5. Distributing the application

Overview.....	179
Deployment.....	179

## PART 5 - WINDOWS MOBILE APPLICATION

### Lesson 5.1. My first Windows Mobile project

Overview.....	184
My first project .....	184
Creating the project.....	184
My first window .....	186
Overview .....	186
Creating the window.....	187
Displaying a message .....	188
First test.....	190
First deployment on the device .....	191
Principle.....	191
Implementation.....	191

### Lesson 5.2. Databases

Format of the databases.....	194
HFSQL database .....	194
CEDB .....	195
AS/400 .....	195
Sharing data between two applications .....	196
Handling the same data files.....	197
Copying the data files onto the mobile device (Pocket PC for example) .....	197

### Lesson 5.3. Advanced programming

Overview.....	199
Opening the project.....	199
Describing the project.....	199
Displaying the list of products .....	200
Creating the window.....	200
Creating the Table control.....	200
Creating the form window.....	202
Creating the window.....	202
Creating the controls.....	203
Displaying the form from the list of products.....	205
Managing the creation and the modification of a product .....	206
Modifying the product form.....	206
Creating a new product.....	207

<b>Creating a menu window .....</b>	<b>209</b>
Creating the window.....	209
Customizing the menu .....	210
Programming the menu .....	211
<b>Running the test of the application.....</b>	<b>212</b>
<b>Creating the executable .....</b>	<b>212</b>

## **Lesson 5.4. Distributing the application**

---

<b>Overview .....</b>	<b>215</b>
<b>Setup in CAB format .....</b>	<b>216</b>
Initial setup .....	216
Update in CAB format.....	217
<b>Setup in MSI format.....</b>	<b>217</b>
Initial setup .....	217
Update in MSI format .....	217

## **Conclusion**

---



# LICENSE

**Caution: By installing and using WINDEV Mobile Express Version, you have agreed with the terms of the license agreement:**

SOFTWARE shall mean the right to use WinDev Mobile Express Version (also called Trial Version, Evaluation Version or Limited Version).

TRIAL USER shall mean the original customer (either individual or company) who installs and/or uses the software.

APPLICATION shall mean a program or a set of programs developed with the SOFTWARE.

## 1. Express Version

The SOFTWARE is proposed in an “evaluation” version.

This trial version is intended for evaluation purpose only. It can also be used by the students of any educational establishment recognized by the relevant authorities during the duration of their studies. Any commercial use, in part or in full, is strictly forbidden.

Several features are crippled, limited or different than the ones in the “commercial” version.

This version cannot be used to create commercial applications. To do so, a “Commercial” version must be purchased.

Creating applications, in part or in full, is strictly prohibited except for evaluation purpose or for practical work in the course of studies.

Applications can be created for evaluation purpose during the authorized period of time.

Unauthorized use of this software and/or use beyond the authorized period of time will be prosecuted to the maximum extent possible under the law.

The software (even in this evaluation version) shall not be duplicated or released without PC SOFT written agreement.

## 2. License

This SOFTWARE shall be used on a single computer by a single person at a time. The SOFTWARE is considered used as soon as it is loaded in the “computer memory” (also called RAM).

This SOFTWARE can be used during the authorized period of time only.

## 3. SOFTWARE ownership

This software is the sole and exclusive property of PC SOFT. The TRIAL USER has only purchased the rights to use the SOFTWARE according to these conditions.

## 4. Documentation

Reproducing the documentation, in part or in full, is strictly forbidden.

## 5. Use

The TRIAL USER agrees to use the SOFTWARE only for the purpose for which it is intended. The CUSTOMER shall not modify, or try to modify, nor disassemble, or try to disassemble this SOFTWARE. The CUSTOMER shall not remove (or try to remove) the copyright mentions that may be displayed and/or included in the SOFTWARE.

The SOFTWARE shall not be rented, leased or shared.

WinDev Mobile “Express” shall not be used to create and distribute all or part of a development tool, SOFTWARE for application creation and program generator, or any other SOFTWARE of a type similar to WinDev Mobile. Contact us if you have any questions.

The photographs, illustrations and images contained in the examples, templates, skins and skin templates can only be used in their initial implantation. Any use, reproduction, representation, adaptation or copy, in full or in part, in any format, by any existing or forthcoming process, is forbidden.

The images and the animations available from the image catalog must be exclusively used in the APPLICATIONS developed with the SOFTWARE. These images shall not be sold or distributed. The use in products or sites other than the APPLICATION is forbidden.

The images cannot be separated from the APPLICATION that uses them.

## 6. Absence of liability

SOFTWARE and documentation are sold WITHOUT ANY WARRANTY as to performance, merchantability or fitness for any particular purpose. PC SOFT shall not be liable for damages of any nature whatsoever, including data loss or deterioration, financial or operational loss, discrepancies between information supplied in the documentation and SOFTWARE behavior, in case the SOFTWARE behaves differently than expected. The entire risk as to results and performance of programs created with the SOFTWARE is assumed by the TRIAL USER.

## 7. Special cases

### TomTom component

If the SOFTWARE is supplied with the TomTom component, the rights of the TomTom software belong to the TomTom company. The TomTom component can only be used with a valid license of TomTom Browser.

### Features linked to services

Warning: before using a feature linked to services or applications (including Google), we strongly recommend that you check the license for using this service or this application. Some restrictions may apply. The content of licenses may change over time.

PC SOFT is in no case responsible for the way the native access functions are used. Make sure that you comply with the license of the service provider.

### Applications developed for iPhone and iPad

**Warning:** Any APPLICATION developed with the SOFTWARE and intended to operate on a device designed by the Apple company may have to comply with one or more licenses belonging to the Apple company.

## 8. Third-party software

Fonts installed

The “Open Sans”, “Roboto” and “Droid font” fonts installed and used by the SOFTWARE are subject to the “Apache License” license. The text of this license is found in the “Help” sub-directory of the SOFTWARE.

The “PCSxxxx” fonts used by the SOFTWARE and found in the “Fonts” directory of the SOFTWARE are subject to the “Apache License” and “SIL Open Font License” licenses. The text of these licenses is found in the “Help” sub-directory of the SOFTWARE.

### Acceptance of the license:

By installing and using this software, you have agreed with the terms of license agreement.

This license shall not be modified by the representations of anyone unless a written amendment has been signed by a corporate officer of PC SOFT.

This license agreement is governed by French Laws; any disputes that may arise are to be settled within the jurisdiction of the courts where PC SOFT has its headquarters.

PC SOFT,

3, rue de Puech Villa

BP 44 408

34197 Montpellier Cedex 5

France



# INTRODUCTION

## Preliminary points

---

**Caution:** This manual is a tutorial. We advise you to refer to the online help when you are using WinDev Mobile.

The aim of the tutorial is to help you discover WinDev Mobile, become familiar with the editors and teach you the concepts of WinDev Mobile. This manual does not cover all the features of WinDev Mobile.

**This manual is intended for the developers who are already familiar with WinDev.** This manual only presents the main concepts required to develop an application for a mobile device (operating in Android, iPhone, iPad, Windows Mobile, Windows Phone, ...).

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

Note: To receive the WinDev tutorial, get in touch with the sales department of PC SOFT.

You should plan on spending a few hours to follow this course and to learn WinDev Mobile: you'll find it well worth it !

WinDev Mobile evolves all the time, so the screen shots found in this course may differ from the screen shots found in your product.

## Overview of the tutorial

---

The tutorial was intended to help you develop on the main mobile platforms:

- Android.
- iOS (iPhone/iPad).
- Windows Store apps (applications that operate in the new interface of Windows 8: Windows 8/RT tablets and PC running Windows 8 and later).
- Windows Mobile/CE.

A lesson is proposed for each one of these platforms. You chose your development platform and you follow the corresponding lesson.



Notes

WinDev Mobile can also be used to develop Windows Phone applications. See the online help for more details.

The tutorial was designed to progressively teach you how to use WinDev Mobile. By following this course:

- 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 published. Don't forget to see the online version of the tutorial. The PDF file is accessible from the menu of WinDev Mobile: on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Tutorial (PDF)".

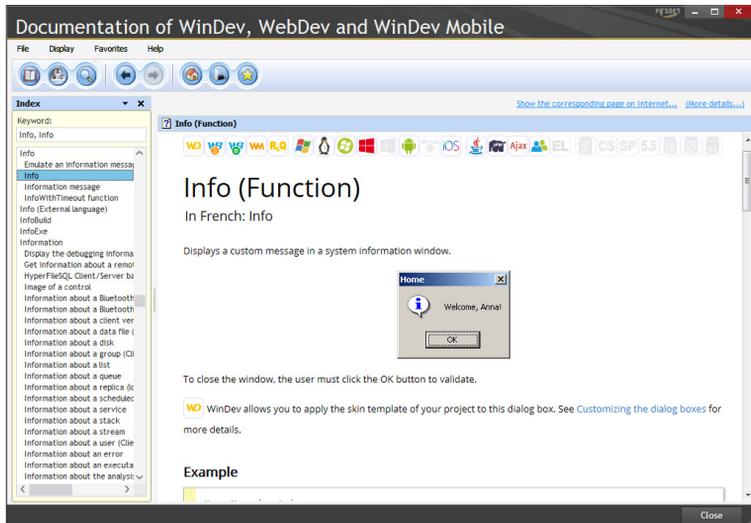
## How do I access the online help?

The online help of WinDev Mobile allows you to get detailed information about the 2500 WLanguage functions. The online help also contains the help about the editors and the controls, tips, ...

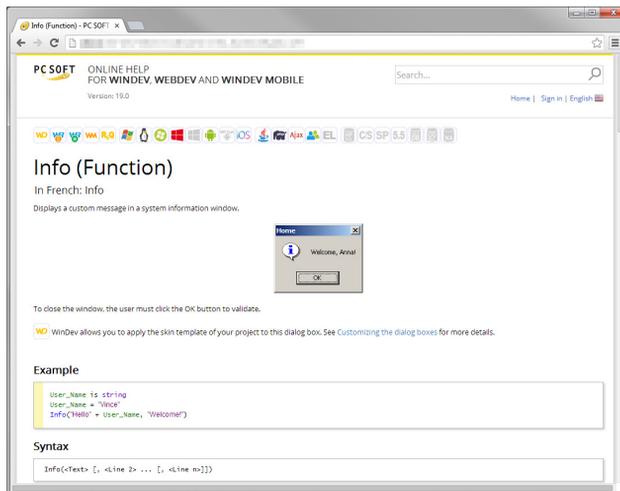
The online help is available at any time in WinDev Mobile:

- 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 a specific "help browser":



- in an Internet browser, if you have access to Internet:





Notes

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. This help is updated on a regular basis.  
 Each Web user can add comments about the documentation pages: personal notes, examples, links..

**To start the Internet online 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 content of the help: help common to WinDev, WebDev and WinDev Mobile or help for the product currently used.

**Legend of symbols**



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 "Wizards, Examples and Components" pane of WinDev Mobile.



This symbol introduces a "Tip": we strongly advise you to read the associated text.



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



This symbol introduces a "Note": we recommend that you to read the associated text.



This symbol gives the result of a "Test": we advise you to read the associated text.

## **If you are familiar with WinDev Mobile 18...**

---

If you are familiar with WinDev Mobile 18, following this course 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 allow you to develop applications in several 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, the 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!





**PART 1**

# Overview of WinDev Mobile

# 10

**Express**



**DEVELOP 10 TIMES FASTER**



**PCSOFT**



# LESSON 1.1. DISCOVER WINDEV MOBILE

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

- Starting WinDev Mobile



Estimated time: 10 min

## Overview

---

WinDev Mobile is an IDE (Integrated Development Environment) allowing you to develop applications for mobile devices in several fields: management, industry, health care, catering, ... The developed applications can give access to the information stored in 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 19 (if not already done).
- ▶ **If WinDev Mobile 19 was never started before**, a welcome wizard is displayed. This wizard is used to:
  - If you own an earlier version of WinDev Mobile, retrieve the existing configurations.
  - If you are a new user, configure your environment. See the online help for more details.
- ▶ **If WinDev Mobile 19 was already started**, identify yourself if necessary. The development environment and the welcome window appear.

Let's take a look at the development environment of WinDev Mobile.

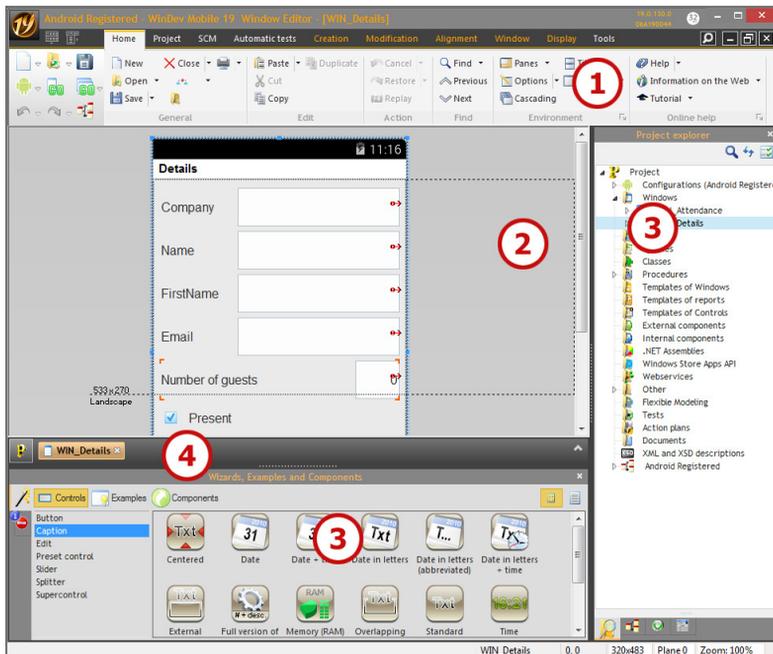
## Development environment

### 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, ...

All the editors use the same environment:



**1. Menu of editors**, displayed in the format of a ribbon (we'll see how to use it in the next paragraph).

**2. Current editor** (window editor here). This space allows you to view 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 the project elements by category.
- The "Wizards, Examples and Components" pane (at the bottom) is used to quickly access the full examples, sample windows and controls that can easily be included in your applications.

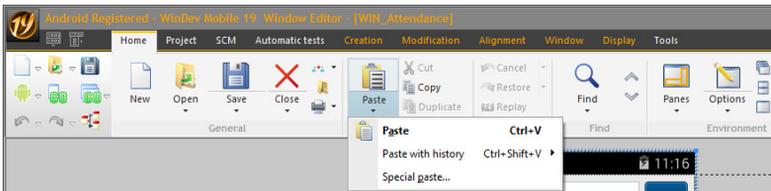
These panes can be hidden by pressing [CTRL] + [W] if necessary.

**4. Bar of opened documents.** This bar is used to quickly view all the 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 the shape of a ribbon. This ribbon includes panes in which the options of the editors are grouped.

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



## The different ribbon elements

The ribbon includes three areas:

- the button area, on the left.
- the pane area, at the top.
- the option area.

Let's take a closer look at these areas.

## The button area

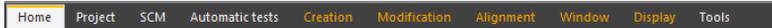


The button area groups the **quick access buttons**. These buttons are used to perform the most usual operations, common to all the editors: save, open, create, ...

The 3 buttons found at the top of this area are specific buttons:

- The logo of the product is used to display the "About" window, the custom menus and the drop-down menus found in the former interface of the editors (version 17 and earlier).
- The 2 other logos are used to restore the toolbars and the drop-down menus found in the former interface of the editors (version 17 and earlier).

### The pane area



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

- the current pane: The name of the current pane is displayed on a white background and an orange line is displayed above the name.
- the popup panes, specific to the current element: The name of the pane is displayed in orange.
- the available panes: The name of the pane appears in white.

### The option area



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 either 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: displaying the description of the current element, displaying 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".





**PART 2**

**Android  
application**

**10**

**Express**



**DEVELOP 10 TIMES FASTER**



**PCSOFT**



## 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: 1h

## Overview

---

To start working with WinDev Mobile, 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 with WinDev Mobile.

Before creating our first project for Android, a configuration of the development computer is required.

## 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 (2.2, 2.3, 4, ...).

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

We advise you to restart the computer after these two setups.

## My first project

---

### Creating the project

We are now going to create our first project for Android. If you own the Android device on which the application must be run, we advise you to connect this device to the development computer. Therefore, the characteristics of the device will be automatically detected and proposed when creating the Android project.



EXAMPLE

Answer

A corrected project is available. To open this project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My Android project (Answer)".

- To create a project:
1. Start WinDev Mobile 19 (if not already done). Close (if necessary) the current project to display the home window.

2. In the home window, click the "Create a project" icon then "Android application".



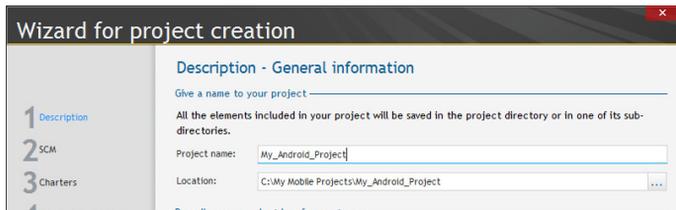
3. The wizard for project creation starts. The different wizard screens help you create your project. The information specified in this wizard can be modified later.

Notes

Tip: To create a project, you can also:

1. Click among the quick access buttons of the WinDev Mobile menu.
2. Click "Project" in the wheel that is displayed.

4. The first wizard screen is used to enter the name of the 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.



5. Go to the next screen via the arrows found at the bottom.
6. The wizard proposes to add documents. Keep the default options and go to the next screen.
7. The wizard proposes to configure the directory of Android SDK. Only the root directory of the different versions of Android platforms is required.

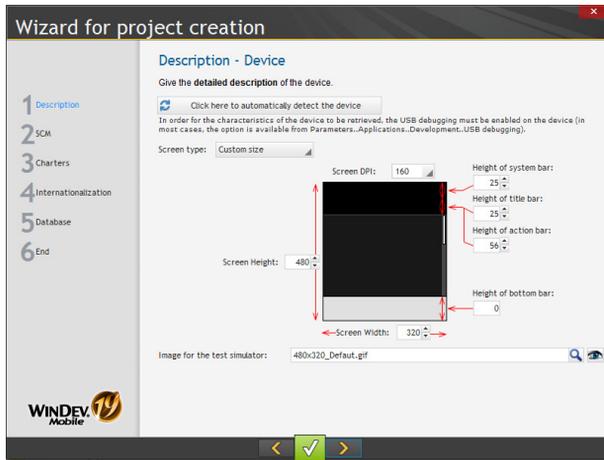


- Note: The path of the SDK is stored for all the forthcoming projects.  
Go to the next screen.
8. Choose the device that will be used to develop your application.
- Caution: this screen is important because the size of your windows will depend on this choice.

If your device is connected, select "Other" in the proposed list and go to the next screen.  
If you have no device, choose one of the devices proposed in the list ("Samsung Galaxy S III" for example).

Go to the next screen.

9. If you have chosen "Other", the screen for detecting your device is displayed.



Click the "Click here to automatically detect the device" button. The characteristics are automatically updated. Go to the next screen.

10. In the left section of the wizard, click "Charters". This step is used to define the programming charter. Don't modify the suggested options. Go to the next screen via the arrows found at the bottom.

11. We are now going to define the style book. Select "ActivAndroid 4-HoloLight".

12. The other wizard steps not being important for our first project, click "End" in the left section of the wizard.

13. Click the validation button at the bottom of the wizard. The project is automatically created. WinDev Mobile displays the different possible actions.

14. Click "WinDev Mobile editor".

## 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, too simple, but we recommend that you 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. Click  among the quick access buttons of the WinDev Mobile:



2. A window shaped like a wheel is displayed. This window is used to create all the elements that can be associated with a project.
3. Click "Window". The wizard for window creation starts.
4. Select "Blank" in the list of windows displayed on the left. In the list of skin templates found on the right, the "ActivAndroid 4-HoloLight" skin template is selected by default. You can choose another skin template proposed in the list.

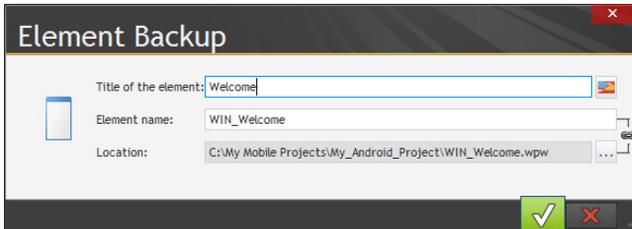


Note

The skin templates allow you to quickly create outstanding interfaces. A skin template defines the style of the window but also the style of all the controls that will be used in this window. No ugly interface anymore.

5. Validate. The window is automatically created in the editor.

- ▶ Save the window by clicking  among the quick access buttons. During the first backup, a specific window is displayed. This window proposes to enter:



- the title of the element: enter "Welcome". In our case, this title will be displayed in the title bar of the window.
- the name of the element is the name of the window. This name will be used in programming. By default, this name includes "WIN\_" that corresponds to the programming charter and "Welcome" that corresponds to the title of the window.



Note

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 uses a programming charter.

The programming charter 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 don't want to use this charter, all you have to do is disable it: on the "Project" pane, in the "Other actions" group, expand "Charter" and uncheck "Use the charter".

- the location that corresponds to the file name created for the window. The window is a file whose extension is "WPW", saved in the project directory.
- ▶ Click the green button to validate the backup of the window.

## Displaying a message

You are now going to create a button used to display a message.

- ▶ To create the "Display" button:
  1. On the "Creation" pane, in the "Usual controls" group, click . The button appears in creation under the mouse.
  2. Move the mouse toward the position where the control must be created in the window (at the top of the window for example). To drop the control in the window, all you have to do is perform a click in the window.
  3. Perform a right mouse click on the control that was just created. The popup menu of the control is displayed. Select "Description" from this popup menu. The description window of the button is displayed.

- ▶ Modify the characteristics of the control by entering the following information:



1. Name of this control: "BTN\_Display".
2. Caption of this control: "Display"

- ▶ Validate the description window of the control (green 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**.



Notes

WLanguage is the programming language supplied with WinDev Mobile. It is a 5th generation language (5GL) that includes highly sophisticated commands.

1. Select the "Display" button with the mouse: all you have to do is click it.
2. Display the popup menu of the control (right mouse click).
3. Select "Code". This option opens the code editor of WinDev Mobile, in which all the WLanguage statements can be entered.
4. Enter the following code in the "Click BTN\_Display" process:

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

**Note about the assisted input:** As soon as the first two characters are typed, WinDev Mobile proposes all the words of the WLanguage vocabulary containing these characters. The assisted development is a very powerful feature. No more mistake when typing 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.



Notes

When typing this code in the code editor, you have noticed that different colors are used by the different elements. 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.

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

- ▶ Save the modifications by clicking  among the quick access buttons or by pressing [Ctrl]+[S].
- ▶ Close the code editor (cross at the top right of the code editor). The window re-appears.

## First test

---

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



Notes

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

To run this test, you must:

- start the AVD emulator and create different configurations corresponding to the requested targets.
- generate the Android application. This operation is explained in the next paragraph.
- select the requested target configuration at the end of generation to run the application test on the emulator.

- ▶ Let's now run the test of the window 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 ("Samsung Galaxy S III" in our example).
  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, a SINGLE CLICK allows you to run the test of the 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 editor of WinDev Mobile is redisplayed.

## First deployment on the device

---

### Principle

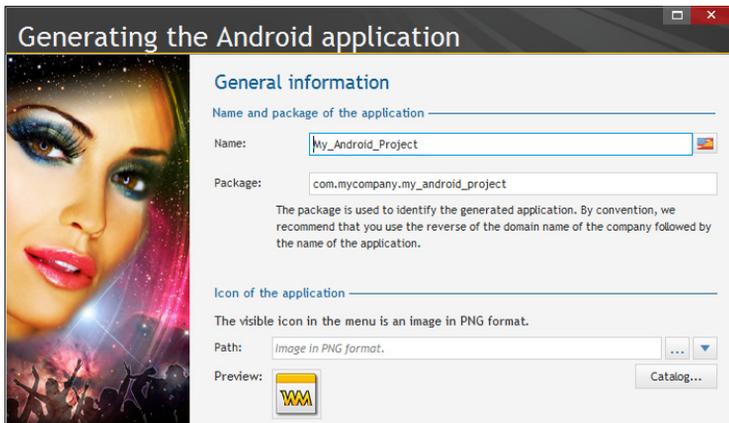
To run the application in stand-alone mode on the Android device, you must:

- Connect the device via a USB port.
- Generate the application.
- Select your device at the end of generation. Copying the application can take several seconds.

Let's take a look at these different steps.

## 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 (green button at the bottom of the screen).
  3. The wizard for generating an Android application starts.



4. The first wizard screen is used to:
  - define the name of the application (displayed below the icon used to start the application) and the corresponding package.
  - select the icon of the application in the image catalog of WinDev Mobile.
5. Go to the next screen by clicking the arrow keys found at the bottom of the screen. This screen is used to define:
  - the splash screen of the application,
  - the information saved in the manifest,
  - the start mode of the application (when the device is started or not).
6. Go to the next screen. This screen is used to define the version number of the application.
7. Go to the next screen. This screen is used to sign the application. The wizard proposes a generic signature that can be used for the tests of the application. A specific signature is required to distribute the application. See the online help for more details.  
**Note:** The signature of applications is not available in Express version.
8. Go to the next screen. This screen is used to include specific files (data files, images, ...). This possibility will not be used in our example. Keep the default options.

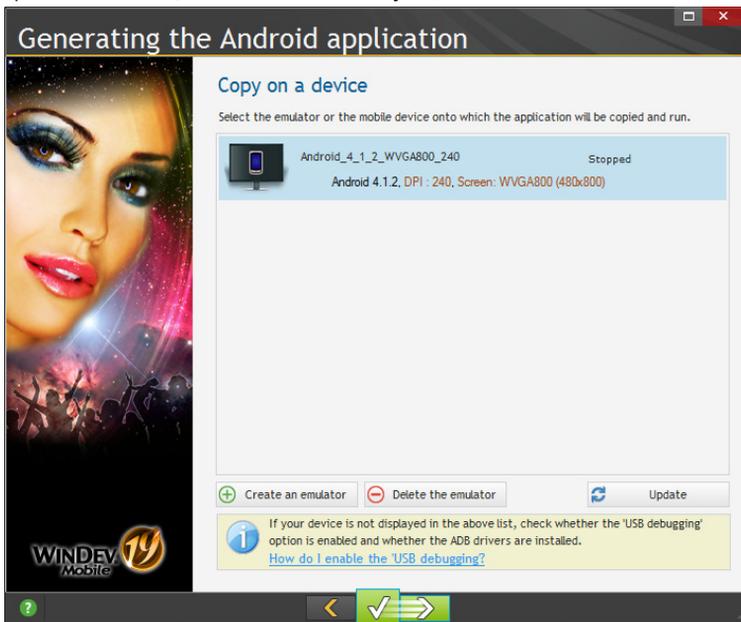
9. Go to the next screen. This screen is used to include specific libraries. Keep the default options.
10. Go to the next screen. This screen is used to define the permissions of the application. By default, according to the WLanguage functions used in the application, WinDev Mobile detects the necessary permissions.
11. Go to the next screen. This screen is used to restrict the download of the application on Google Play store to the devices equipped with the features used. This possibility will not be used in our example. Keep the default options.
12. Go to the next screen. This screen is used to configure the options of Android SDK. Keep the default options.
13. Go to the next screen. The last wizard screen is used to specify whether the application must be copied and run on the device connected to the computer or on an emulator.



Notes

If no device is connected or if no emulator is displayed, you must create an emulator ("Create an emulator" button).

If this option is checked, a new screen allows you to select the runtime device :



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

## LESSON 2.2. INTERFACE (GUI)

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

---

- Choosing the resolution according to the device
- Orientation of the window
- Management of touchscreen



Estimated time: 30 min

## 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:

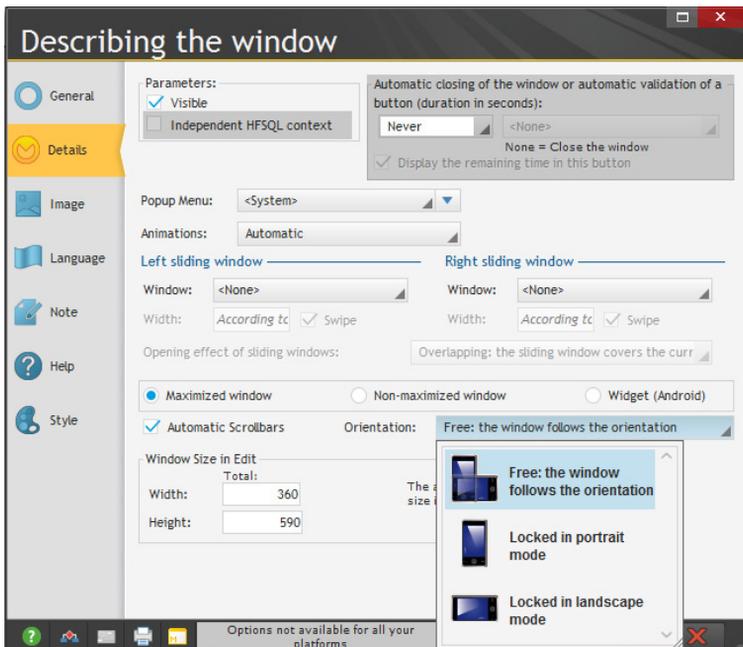
- **You are developing for a single target device:** in this case, all you have to do is select this device in the list proposed by the wizard.
- **You are developing for several target devices with different screen resolutions:** in this case, you must choose the smallest resolution common to all these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

## Orientation of the window

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

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

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



In the two first 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.



EXAMPLE

Answer

A corrected project is available. To open this project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My Android project (Answer)".

In our example, the project was created for a phone and its test was run in portrait mode in the simulator.

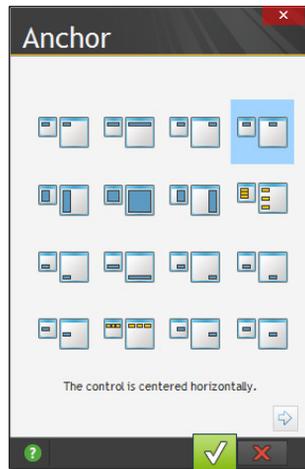
We are now going to run its test in landscape mode in the simulator.

- ▶ Run the test of the project (Go among the quick access buttons).
  1. The window is displayed in portrait mode.
  2. In the simulator, click the arrow in the shell (↵).
  3. The orientation of the window changes on the screen.
  4. In our example, the location of the button does not change: it does not adapt to the orientation of the screen.



- ▶ We are now going to modify our window in order for the "Display" button to be centered in the window and to remain centered regardless of the device orientation.
- ▶ Stop the test and go back to the editor.

- ▶ To center the button in the window:
  1. Select the button (click the button).
  2. On the "Alignment" pane, click "Center in the parent (horz)".
- ▶ In order for the button to remain centered in the window, we are going to use the control anchoring:
  1. Select the button (click the button).
  2. Display the popup menu (right mouse click).
  3. Select "Anchor": the window for defining the anchors is displayed:



4. Select "Horizontally centered" and validate (green button).
- ▶ Run the test of the project (🏃 among the quick access buttons):
    - The button is centered in portrait mode.
    - Change the orientation of the simulator.
    - The button remains centered in landscape mode.

## Management of touchscreen

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

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

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

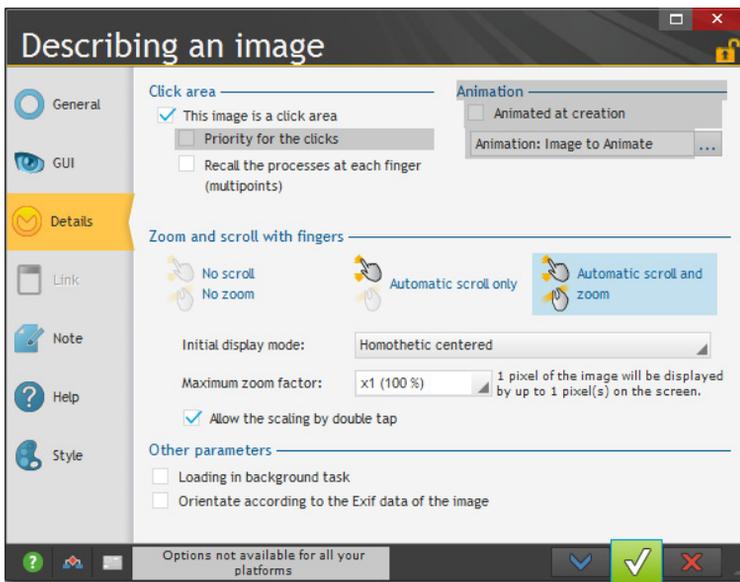
- ▶ Open (if necessary) the "My\_Android\_Project" project that was created in the previous lesson.



Answer

A corrected project is available. To open this project, on the "Home" pane, in the "Online help" group, expand "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 must be created. To drop the control in the window, all you have to do is perform a new left mouse click.
  4. Double-click the Image control: the description window of the control is displayed.
  5. In the "General" tab, select an image found on your disk in the "Image" edit control via the file picker.
  6. Display the "Details" tab: the options for managing the multi-touch are displayed:



7. Select "Automatic scroll and zoom".
8. Validate the description window of the control.
9. Save the window (click  among the quick access buttons).
10. A GUI error appears in the error pane: the automatic scrollbars of the window are in conflict with the scroll features of the Image control.

11. To avoid this GUI error, disable the scrollbars of the window:
  - Display the description window of the window ("Description" from the popup menu).
  - In the "Details" tab, uncheck "Automatic scrollbars".
  - Validate the description window.
12. Save the window (click  among the quick access buttons). The GUI error disappears.
13. Close the project.

## The different types of controls

---

WinDev Mobile proposes several controls to communicate with the user. 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 zones to create GUI 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 these controls are presented in the "Advanced programming" lesson.

## LESSON 2.3. DATABASES

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

---

- Available databases
- Synchronization



Estimated time: 15 min

## The available databases

---

### Overview

Two types of databases are available in Android:

- HFSQL (Classic mode and Client/Server mode).
- SQLite.

### HFSQL database

#### HFSQL Classic

In HFSQL Classic mode, the data files are stored on the device (Smartphone or tablet).

In this case, the application is stand-alone. No Wi-Fi or 3G connection is required.

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

In the "Advanced programming" lesson, we will develop an application that uses a HFSQL Classic database.

#### 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 the 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.

### SQLite database

SQLite is a free database format recognized by the Android system.

This database is stored in the memory of the device. It is accessible from WinDev Mobile via the Native SQLite Access.

The access to the data will be performed by the Hxxx functions of WLanguage and/or by SQL queries.

The SQLite database is also available in WinDev and WebDev via the same Native Access.

You cannot directly handle a SQLite database found on a PC from a mobile device (tablet or Smartphone) and conversely. The synchronization methods must be used to keep the data updated.

## **The synchronization**

---

The synchronization mechanism is used to "synchronize" the data stored on a mobile device with the data stored on a server. The synchronization uses the mechanism of "universal replication". This technique is available in WinDev, WebDev and WinDev Mobile.

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

## **Accessing data via a Webservice**

---

Another method can also be used to access the data on a server from a mobile device: the call to a Webservice.

In this case, the mobile device must be equipped with a Wi-Fi or 3G connection to connect to the Webservice.

The mobile application does not directly access the database. The application calls the functions of the Webservice. These functions return the data.

It is the Webservice that accesses the database.

This technique is used to have a business layer (the Webservice) common to several types of applications and interfaces (WinDev, WebDev or WinDev Mobile) and different types of operating systems (Windows, Android, iOS, ...).

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

## LESSON 2.4. ADVANCED PROGRAMMING

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

---

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



Estimated time: 1h

## Overview

---

In this lesson, we are going to develop an Android application that uses a HFSQL Classic database. This application will allow us to present some specific features of the Android programming.

### Opening the project

- ▶ Start WinDev Mobile 19 (if not already done). Close (if necessary) the current project to display the home window.
- ▶ Open the "Android Managing Products" project.  
To do so, in the home window, click "Tutorial" and select the "Android Managing Products (Exercise)" project.  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Android Managing Products (Exercise)".



EXAMPLE

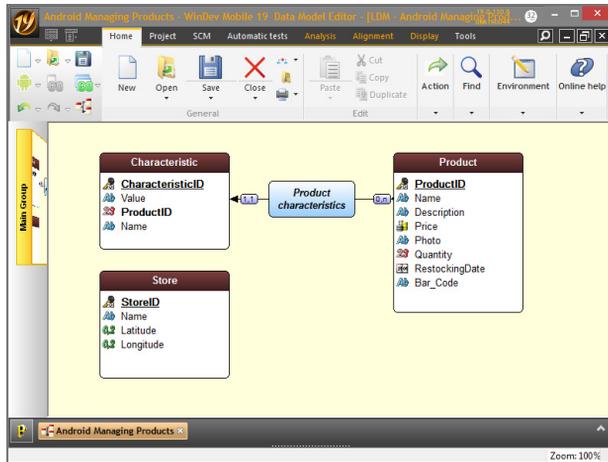
Answer

A corrected project is available. This project contains the different windows created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Android Managing Products (Answer)".

### Describing the project

Let's take a look at our source 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 the WinDev Mobile menu.
  2. The data model editor is displayed.



3. This analysis includes 3 data files:

- A "Product" file, that contains the description of the product: name, price, quantity, ...
- A "Characteristic" file, that contains the different characteristics of the product. For example, if the product is a tee-shirt, its characteristics will correspond to the size, the color, ... Therefore, the "Characteristic" file is linked to the "Product" file.
- A "Store" file, that contains the GPS coordinates of each store.

4. Close the data model editor (click the cross at the top right of the 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 "Looper" control.

### Creating the window

► To create a new window:

1. Click  among the quick access buttons. Click "Window" in the wheel that is displayed. Choose "Blank" and validate the wizard.
2. Save the window by clicking  among the quick access buttons.
3. In the window that is displayed, enter the title of the window: "List of products". The name and location of the window are automatically filled. Keep the default values and validate (green button).
4. The window is added to the project.

## Creating the Looper control

A "Looper" control will be used to display the list of products. The main information about the products will be displayed in this control.

► To create the "Looper" control:

1. On the "Creation" pane, in the "Data" group, expand "Looper" and select "Looper (vertical)". The control appears under the mouse.
2. Click the position where the control must be created in the window (at the top for example). The wizard for creating a Looper control starts.
3. In the wizard, select "Display the data coming from a file or from a query". Go to the next screen.
4. Select the "Product" file. Go to the next screen.
5. Select the items to display:



6. Keep the "Name", "Description" and "Price" items (a checkmark must be found in front of these items). Go to the next screen.
7. Keep the proposed sort item (ProductID). The products will be sorted in the Looper control according to this item. Go to the next screen.
8. In the "Additional parameters" screen, keep the default options. Go to the next screen.
9. Keep the default name ("LOOP\_Product") and validate.
10. A message regarding the management of scrollbars in the window is displayed. Indeed, both the window and the Looper 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 Looper control.

The vertical scroll will be performed in the Looper control and not in the window. Therefore, controls can be positioned above or below the Looper control without scrolling.

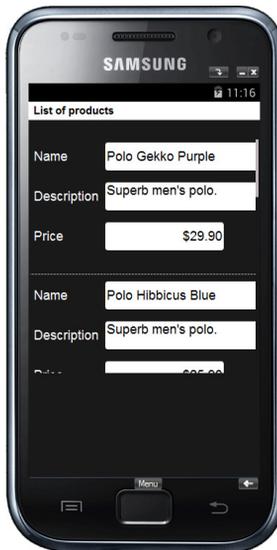
**11.** Click "Disable the automatic scrollbar". The Looper control appears in the window editor.



## Notes

The data automatically appears in the window displayed in the editor. This concept is called "Live Data": you see the data found in your 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 view the result. Click  among the quick access buttons (or press [F9]).



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

## Improving the window

As you have noticed during the test, the controls created in the Looper control are displayed beyond the looper. We are going to resize the controls.

- ▶ To resize the "Name" control:
  1. Select the "Name" control.
  2. Reduce the size of the control with the handles in order for the control to be entirely displayed in the window.
  3. Repeat the operations for the "Description" control.

The importance of anchors was presented in the previous lesson. We are now going to apply an anchor to our Looper control so that it is automatically resized according to the orientation of the screen.

- ▶ To anchor the Looper control:
  1. Select the Looper control.
  2. Display the popup menu of the control (right mouse click) and select "Anchor".
  3. In the window that is displayed, select the option  .
  4. Validate. The red arrows indicating the anchoring options appear in the window.
- ▶ Run the window test to see the result.
  - Click  among the quick access buttons.
  - Modify the orientation of the window.
  - Close the simulator to go back to the editor.
- ▶ Our window is now created.

## 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 the selected product.

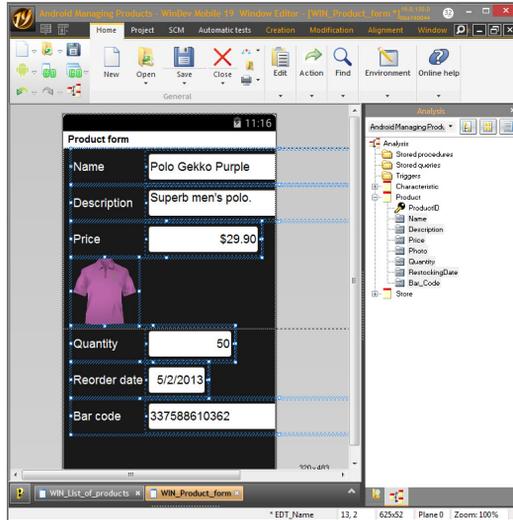
### Creating the window

- ▶ To create the form window:
  1. Create a new blank window.
    - Click  among the quick access buttons.
    - Click "Window" in the wheel that is displayed.
    - Choose "Blank" and validate the wizard.
  2. Save the window. Specify the title of the window: "Product form". Its name is automatically proposed: "WIN\_Product\_form". Validate.

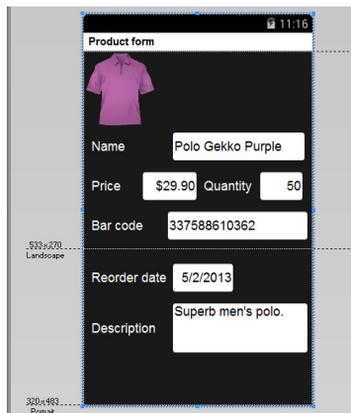
### Creating the controls

- ▶ To create the different edit controls for displaying 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 "Android Managing Products" analysis appear in the pane.

2. With the mouse, select the items of the "Product" 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", "BarCode", "Reorder Date" and "Description") so that they are visible in the window.
5. Reorganize the controls in the window. Respect the following order: "Photo", "Name", "Price", "Quantity", "BarCode", "Reorder Date", "Description".



6. We are going to view the navigation order in the window:
  - Press [F5]. The numbers that are displayed represent the navigation order in the window. Press [F5] again to make the numbers disappear.
  - To correct the navigation order, on the "Window" pane, in the "Order" group, expand "Navigation" and select "Define automatically".
7. Save the window.

- ▶ Run the test of the window (  among the quick access buttons). The window is displayed with empty controls.
- ▶ To display the product data:
  1. Display the processes associated with the window:
    - Perform a right mouse click in the area beside the window
    - Select "Code" from the popup menu.
    - The code editor appears.
  2. Enter the following code in the "End of initialization of WIN\_Product\_form" process:

```
FileToScreen ()
```

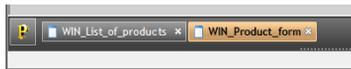
**FileToScreen** is used to display in the controls the data found in the data file, for the current record.

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

### Displaying the form from the list of products

Let's see how to display the form of the 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 found in the button bar:



2. Right-click the Looper control and select "Code" from the popup menu.
3. In the code window that is displayed, enter the following code in the "Selecting a row of..." process:

```
OpenMobileWindow(WIN_Product_form)
```



Notes

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 the existing windows found in the project. All you have to do is select the window with the keyboard or with the mouse.

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

4. Save the modifications by clicking  among the quick access buttons.
  5. Close the code window (click the cross at the top right of the 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.
    - The detailed window of the product is displayed.
  - ▶ Close the simulator.

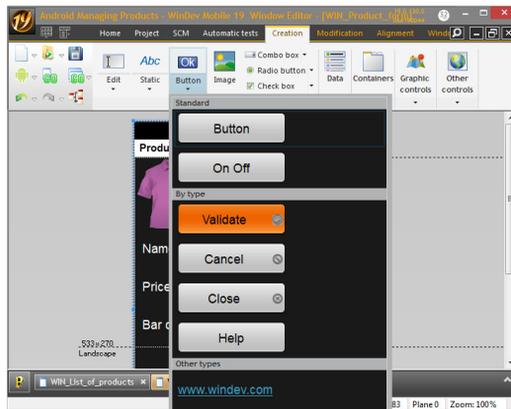
## Managing the creation and the modification of a product

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

### Modifying the product form

We are going to add 2 buttons into the "WIN\_Product\_form" window:

- a "Validate" button to manage the validation of modifications
  - a "Cancel" button to manage the return to the list of products.
- ▶ Display (if necessary) the "WIN\_Product\_form" window in the editor: click the corresponding button in the button bar.
  - ▶ To create the "Cancel" button:
    1. On the "Creation" pane, in the "Usual controls" group, expand "Button" (click the arrow found below ).
    2. The list of preset buttons is displayed.

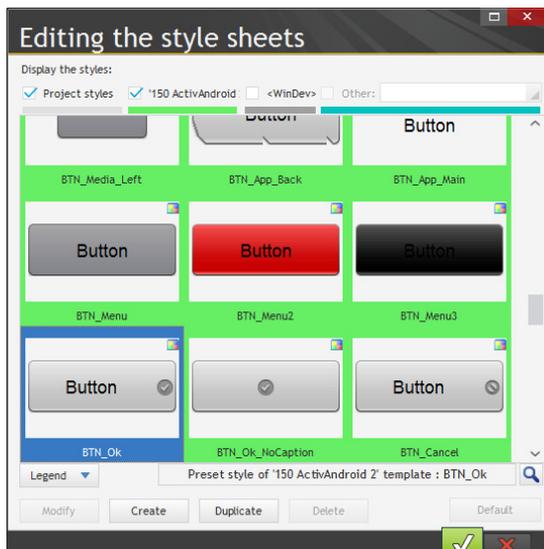


Notes

For some types of controls, WinDev Mobile proposes a list of preset controls. These controls are advanced controls, configured to perform a specific action. For example, the "Cancel" button proposed by the list of preset controls contains the code required by its execution.

3. Click "Cancel": the shape of the button appears under the mouse. Then, click at the bottom right of the window to create the button.

- ▶ We now want to add the "Validate" button and customize the style of this button.
  1. On the "Creation" pane, in the "Usual controls" group, expand "Button" (click the arrow found below ) .
  2. The list of preset buttons is displayed.
  3. Click "Button": the shape of the button appears under the mouse. Then, click at the bottom of the window to create the button (on the left of the "Cancel" button).
  4. Select the control and press the "Enter" key on the keyboard. The caption of the button switches to edit. Type "Validate" and press the "Enter" key on the keyboard.
  5. Resize the button if necessary (with the handles) in order for the caption to be entirely displayed in the button.
- ▶ To customize the style of this button:
  1. Select the "Validate" button: all you have to do is click it.
  2. Display the popup menu (right mouse click) and select "Choose a style".
  3. Each skin template proposes styles for the standard buttons: validation, cancelation, creation, ... Choose the "BTN\_Ok" style.



4. Validate. This style is automatically applied to the button.
- ▶ We are now going to enter the code of the "Validate" button.
  1. Right-click the button and select "Code" from the popup menu.
  2. In the "Click" process, enter the following code:

```
ScreenToFile ()
HModify (Product)
Close ()
```

Let's take a look at this code:

- **ScreenToFile** is used to initialize the items with the values of the linked controls, for the current record.
  - **HModify** is used to update the file data for the current record.
3. Save the modifications by clicking  among the quick access buttons.

4. Close the code window (click the cross at the top right of the 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 "Closing a child window" process of the "WIN\_List\_of\_products" window.
  1. Display the "WIN\_List\_of\_products" window in the window editor.
  2. Right-click the window background and select "Code" from the popup menu. Enter the following code in the "Closing a child window" process:

```
LooperDisplay(LOOP_Product)
```

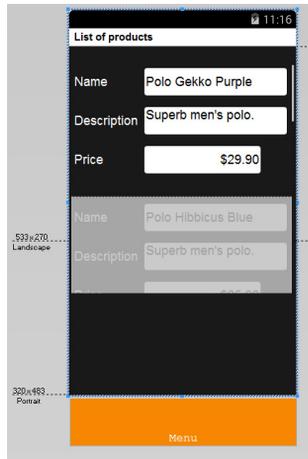
Let's take a look at this code:

- The "Closing a child window" process 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 the "WIN\_List\_of\_products" window.
3. Save the modifications by clicking  among the quick access buttons.
  4. Close the code window (click the cross at the top right of the code editor).
- ▶ Run the test of the "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 whose price is 25.90 Dollars.
    - The detailed window of the product is displayed. Modify the price of 25.90 Dollars and enter 19.90 Dollars and click the "Validate" button.
    - When going back to the list of products, you will notice that the price was updated for this article.
  - ▶ Close the simulator. The editor of WinDev Mobile 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 menu option ("New product") that will be used to open the "Product form" window.
  - Then, we will modify the code of the "Product form" window to manage the addition into the Product data file.
- ▶ To add a menu into the list of products:
    1. Display (if necessary) the "WIN\_List\_of\_products" window in the editor.
    2. On the "Window" pane, in the "Bars and menus" group, expand "Main menu" and select "Add the main menu".
    3. A menu with an option is added at the bottom of the window. This option is named "Menu".



4. We are going to modify this option:

- Right-click the option to display the popup menu.
- Select "Option description". The description window is displayed.
- Modify the name of the option ("OPT\_NewProduct") and its caption ("New Product"). Validate the description window.

5. The code of this option is used to open the "Product form" window and to reset its controls.

To enter this code:

- Right-click the option.
- Select "Code" from the popup menu.
- In the "Selecting the menu" process, enter the following code:

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

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

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

► We are now going to modify the window of the product form to manage the addition of a new record.

1. Display the "WIN\_Product\_form" window in the editor.
2. We are going to modify the code of the validation button:
  - Right-click the "Validate" button and select "Code" from the popup menu.
  - In the "Click" process, replace the existing code by the following code:

```
ScreenToFile ()
IF Product..NewRecord THEN
  HAdd (Product)
ELSE
  HModify (Product)
END
Close ()
```

Let's take a look at this code:

- **..NewRecord** is used to find out whether the current record must be created.
- If **HReset** was called beforehand, the property returns True (case of the 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**.

3. Save the modifications by clicking  among the quick access buttons.

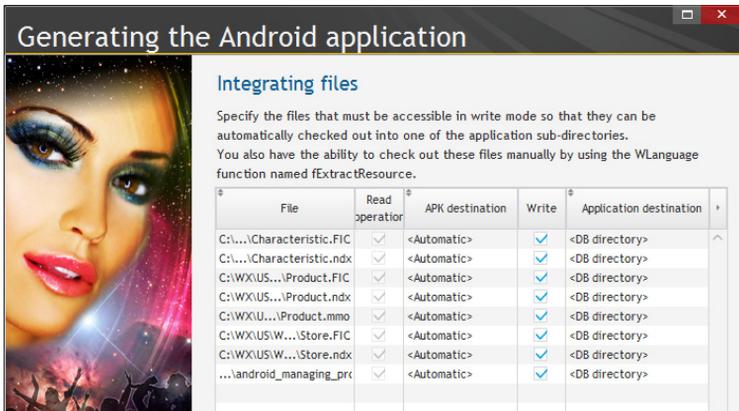
4. Close the code window (click the cross at the top right of the code editor).

- ▶ Display the "WIN\_List\_of\_products" window in the window editor and run its test in the simulator ( among the quick access buttons).
  - Click the "Menu" button and select "New Product".
  - Enter a new product.
  - Validate. The new product appears in the list of products.
  - Close the simulator.

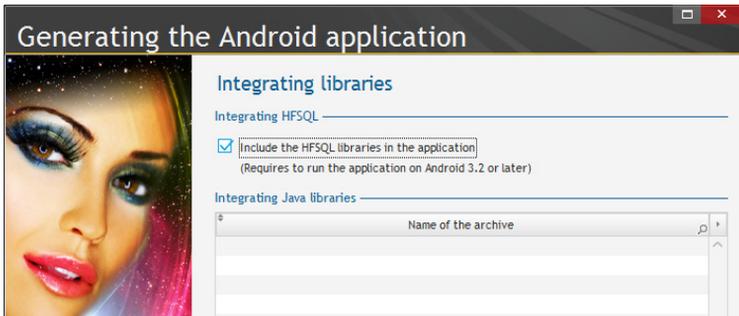
## Deploying the application

If you own an Android device, the operating mode of the application can be checked on your mobile. To do so, generate the application and install it on the mobile device connected to the PC. Before generating the application, we are going to define the first window that will be started by the application.

- ▶ To define the first project window:
  1. Select the "WIN\_List\_of\_products" window in the project explorer.
  2. Display the popup menu.
  3. Select "First project window". A specific icon (with a small 1) is displayed in front of the name of the window, in the project explorer.
- ▶ 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. The wizard for generating an Android application starts. Keep the default options. Via the arrow buttons found at the bottom of the screen, go to the "Integrating files" screen while keeping the default options. Several HFSQL data files have been filled beforehand for our example. They will be supplied with the application. In this case, they must be specified in the "Integrating files" screen.
  3. Click the "Add" button at the bottom of the screen. In the "EXE" sub-directory of the project, select:
    - the HFSQL data files (\*.fic, \*.ndx, \*.mmo).
    - the SQLite database (android\_managing\_products.db file). This database will be used only if the Android version installed on the device is earlier than version 3.2 (minimum version for HFSQL).
  4. Check the "Write" box for each file (required to be able to modify the data from the application).



5. The "Integrating libraries" screen is used to include specific libraries (HFSQL and Java).



If you own a device that uses Android 3.2 (or later), you can check "Include the HFSQL library in the application".

6. Validate the other screens until you reach the "Configuration" screen. This screen is used to configure the options of Android SDK. If you have chosen to include the HFSQL library, check whether the minimum version to run the application is "Android 3.2" (or later). Otherwise, a warning message is displayed in the wizard.



Notes

A database in SQLite format is supplied in our example. This database will be used by default if the Android version is earlier than version 3.2.

7. The last wizard screen is used to specify whether the application must be copied and run on the device connected to the computer or on an emulator. If this option is checked, a new screen allows you to select the runtime device. Select the device connected to the PC.

8. The test of the application can be run on your Android device.

## 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 setup of the application on the device because it uses hardware resources that are not accessible in Simulation mode.

### Overview

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

To do so, we are going to:

- create a button to start the camera. The photo will be retrieved in the format of an image in memory and displayed in the Image control of the product.
- create a button used to select a photo in the album found on the mobile.

### Creating the button for taking photos

► To create the button for taking photos:

1. Display the "WIN\_Product\_form" window in the editor.

2. Add a button into the window:

- On the "Creation" pane, in the "Usual controls" group, click  : the shape of the button appears under the mouse.
- Then, click at the top right of the product image to create the button.

► We are going to modify the button in order to associate it with an image representing a camera:

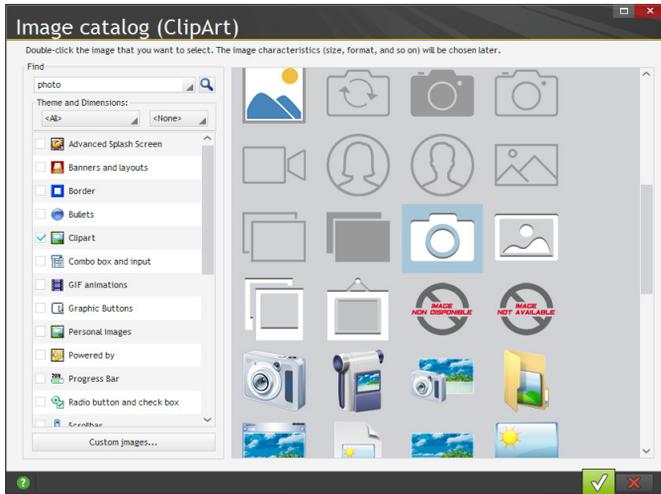
1. Select the button and display its popup menu (right mouse click).

2. Select "Description". The description window of the control is displayed.

3. In the "General" tab, position on the "Image" edit zone and click the "Catalog" button.

4. The image catalog of WinDev Mobile is opened. This catalog contains hundreds of images in different fields, formats and sizes.

5. In the "Find" area (top left), type "photo" and validate. Several images are displayed:



6. Select iOS 5 theme, and double-click the first image to select it.
7. In the next screen, you have the ability to choose the size of the image, the color, the orientation, its format and its name.
8. Keep all the default options and specify the name of the image ("Camera").
9. Validate the window.
10. The path of the image is displayed in the button description.
11. Give a name to the button: "BTN\_Camera".
12. Clear the caption of the button.
13. Validate the description window.
14. In the editor, reduce the size of the button.

## Taking photos

We are going to enter the code used to take a photo and to display it in the Image control of the product form.

► To manage the taking of photos:

1. Select "Code" from the popup menu of the button (right mouse click).
2. Enter the following code in the "Click" process:

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

In this code, **VideoStartApp** is used to start the native camera application of the device in order to save a video or to take a photo.

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

## Selecting a photo in the photo album

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

- ▶ To create the button for selecting the photo, we are going to "Copy - Paste" the button for taking photos that was just created:
  1. Select the "BTN\_Camera" button that was just created.
  2. Press [CTRL] + [C]: the button is copied into the clipboard.
  3. Press [CTRL] + [V]: the mouse cursor changes and the shadow of the button appears under the cursor.
  4. Click in the window beside the "BTN\_Camera" button: the new button is automatically created.
  5. Display the description window of the button (double-click the control):
    - Give a name to the button: "BTN\_Photo\_Album".
    - Select an image in the image catalog.
  6. Validate the description window.
- ▶ The code of this button is used to open the photo album of the device and to select an image in order to display it in the Image control of the product form.
  1. Select "Code" from the popup menu of the button (right mouse click).
  2. Replace the code of the "Click" process by the following code:

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

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

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

## Deploying the application

If you own an Android device, the operating mode of the camera can be checked on your mobile. To do so, generate the application and install it on the mobile device connected to the PC.



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 setup of the application on the device because it uses hardware resources that are not accessible in Simulation mode.

**The deployment is not mandatory:** the application can be deployed at the end of the tutorial to see all the features on the mobile device directly.

- To deploy the Android application directly:
1. Among the quick access buttons, expand  and select "Run the project test (mobile device)".
  2. WinDev Mobile automatically generates the Android archive corresponding to the application with the options selected during the last generation.
  3. At the end of generation, the application is automatically copied onto the device connected to the development computer.
  4. The test of the application can be run on your Android device.

## Managing the bar code of the product



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 setup of the application on the device because it uses 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 camera of the device and by a specific WLanguage function.

### Implementation

- To create the button for managing the bar codes:
1. Display (if necessary) the "WIN\_Product\_form" window in the editor.
  2. Add a button into the window:
    - on the "Creation" pane, in the "Usual controls" group, click .
    - the shape of the button appears under the mouse.
    - create the button beside the "Bar Code" control (resize the edit control if necessary).
  3. Display the description window of the button (double-click the control):
    - Give a name to the button: "BTN\_Bar\_Code".
    - Clear the caption of the button.
    - 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 size of the button.

- ▶ The code of this button is used to scan the bar code.
  1. Select "Code" from the popup menu of the button (right mouse click).
  2. Enter the following code in the "Click" process:

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

In this code, **BCCapture** is used to decode the information stored in a bar code by using the camera of the device.

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

## 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 the 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.

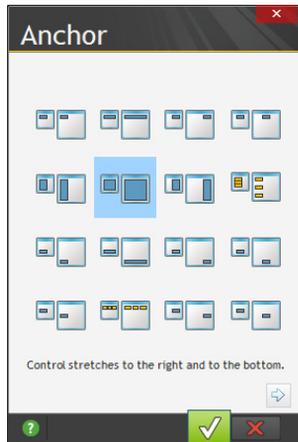
- ▶ To create the window:
  1. Create a new blank window. Click  among the quick access buttons. Click "Window" in the wheel that is displayed. Choose "Blank" and validate the wizard.
  2. Save the window. Specify the title of the window: "Map of stores". Its name is automatically proposed: "WIN\_Map\_of\_stores". Validate.

### Creating the Map control

- ▶ To create the Map control:
  1. On the "Creation" pane, in the "Graphic controls" group, click "Map". The shape of the control 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. Display the description window of the "Map" control (double-click the control for example).
  6. In the description window of the control, specify the name of the control ("MAP\_Store") and validate.
  7. In the editor, resize the control (with the handles) so that it occupies the entire window.

In order for this window to be usable both in portrait mode and in landscape mode, we are going to define anchors on the Map control: the size of the control will automatically adapt to the orientation of the window.

- ▶ To define anchors on the Map control:
  1. Select the Map control (click below).
  2. Display the popup menu (right mouse click) and select "Anchor".



3. Select "the control stretches to the right and to the bottom".
4. Validate the anchoring window. Two red arrows appear in the editor.
5. Save the modifications by clicking  among the quick access buttons.

## Displaying the stores on the map

### Principle

We are now going to add the code used to display on a map all the stores found in the "Store" file. To do so, the "Store" file will be browsed by 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 different information. We will be using the following information:

- Name,
- Latitude,
- Longitude.

**MapAddMarker** is used to add a marker onto the map. Then, all you have to do is define a sufficient zoom level to see all the markers on the map. If the zoom is not set properly, some markers may not be visible or they may overlap on the map.

## Implementation

► To enter the code used to display the stores:

1. Right-click outside the window. Select "Code" from the popup menu. The processes associated with the window are displayed.
2. Enter the following code in the "Global declarations of..." process.

```
// Global variables
gMarker is Marker

// Load the stores
FOR EACH Store
  // Coordinates of marker
  gMarker.Position.Latitude = Store.Latitude
  gMarker.Position.Longitude = Store.Longitude
  // Name of marker
  gMarker.Name = Store.Name
  // Add the marker
  MapAddMarker (MAP_Store, gMarker)
END
// Default zoom
MAP_Store..Zoom = 7
MapDisplayPosition (MAP_Store, gMarker.Position)
```

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

A test can be run in the simulator but the mapping functions can only be run on the device. Indeed, this feature requires functions specific to the device and to the Android SDK.

## Deploying the application

If you own an Android device, the operating mode of the Map control can be checked on your mobile. To do so, generate the application and install it on the mobile device connected to the PC.



Caution!

The Map control uses the Google Maps API v2 library.

Several steps are required to use the Google Maps API v2 library in an Android application. You must:

1. Install the Google Play Services SDK.
2. Get a key for using the library.
3. Copy the key into the wizard for generating the Android application.

- ▶ Define the first project window (the first window that will be started when running the application).
  1. Select the requested window in the project explorer: "WIN\_Map\_of\_stores" in our example.
  2. Display the popup menu.
  3. Select "First project window". A specific icon (with a small 1) is displayed in front of the name of the window, in the project explorer.
- ▶ 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. The wizard for generating an Android application starts. Keep the default options. Via the arrow buttons found at the bottom of the screen, go to the "Integrating files" screen while keeping the default options. Several HFSQL data files have been filled beforehand for our example. They will be supplied with the application. In this case, they must be specified in the "Integrating files" screen.
  3. Click the "Add" button at the bottom of the screen. In the "EXE" directory of the project, select:
    - the HFSQL data files (\*.fic, \*.ndx, \*.mmo).
    - the SQLite database (android\_managing\_products.db file). This database will be used only if the Android version installed on the device is earlier than version 3.2 (minimum version for HFSQL).
  4. Check the "Write" box for each file (required to be able to modify the data from the application).
  5. The "Integrating libraries" screen is used to include specific libraries (HFSQL and Java). If you own a device that uses Android 3.2 (or later), you can check "Include the HFSQL library in the application".

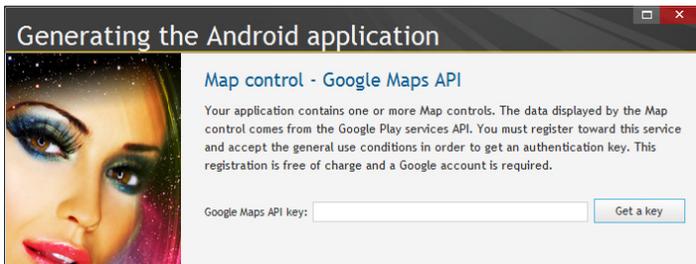
6. Validate the other screens until you reach the "Configuration" screen. This screen is used to configure the options of Android SDK. If you have chosen to include the HFSQL library, check whether the minimum version to run the application is "Android 3.2" (or later). Otherwise, a warning message is displayed in the wizard.



Notes

A database in SQLite format is supplied in our example. This database will be used by default if the Android version is earlier than version 3.2.

7. Go to the next screen. This screen is specific to the use of the Map control. This screen allows you to enter the key required to use the Map control. If you own a Google Maps API key, enter it. Otherwise, click "Get a key".



8. The last wizard screen is used to specify whether the application must be copied and run on the device connected to the computer or on an emulator. If this option is checked, a new screen allows you to select the runtime device. Select the device connected to the PC.

9. The test of the application can be run on your Android device.

## Using a Multiline Zone control

The "Multiline zone" control is often used on the mobile platforms.

This control is used to group several controls:

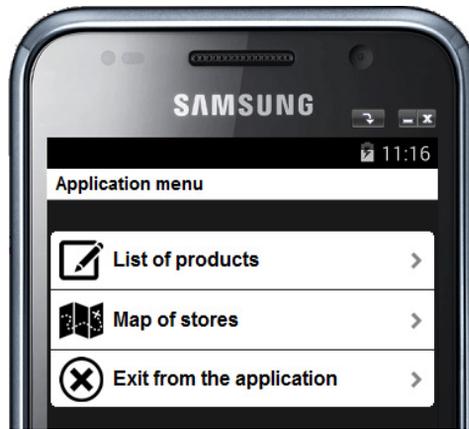
- options on a category,
- group of information about a contact, ...

This control can contain at the same time:

- lines defined in edit (static lines)
- lines defined at run time, by programming (dynamic lines).

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.
    - Click "Window" in the wheel that is displayed. Choose "Blank" and validate the wizard.
  2. Save the window. Specify the title of the window: "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 shape of the control appears under the mouse.
  2. Click inside the window to create the control.
  3. Display the description window of the control (double-click the control for example).
  4. In the description window, specify the name of the control ("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 includes 3 options:

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

## Modifying the Multiline Zone control

- To modify the Multiline Zone control:
1. Display the description window of the control (double-click the control for example).
  2. Click the "New line" button. A window is opened: this window contains all the preset line templates.
  3. Select the "Simple line with picto" template and validate. Repeat this operation twice. Now, the multiline zone contains:
    - a "blank" line,
    - 3 "simple lines with picto".



4. We are going to delete the blank line that is useless in our example:
  - Select the blank line (the first line) with the mouse.
  - Click the "Delete" button.
5. We are now going to modify the style of the multiline zone.
  - Display the "Style" tab.
  - Click the "Choose" button to choose a new style.
  - In the window that is displayed, choose the "MZ\_NoCaption\_Negativ" style.
  - Validate.
6. Validate the description window. Your menu is created.

Each line 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 line in order to represent the action of the menu.

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

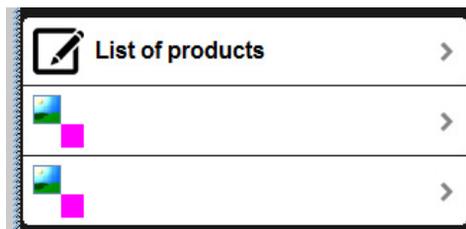
- ▶ To modify the image of the first line found in the Multiline Zone control:
  1. Click the first line and select the Image control.
  2. Display the description window of the image (double-click the control).
  3. In the description window:
    - Give a name to the image ("IMG\_ListOfProducts" for example).
    - Click the "Catalog" button in order to choose an image representing the action.
    - In the window of the image catalog, type "List" in the search control and press the "Enter" key.
    - Select an image via a double click.
    - In the window for configuring the generated image, select a size (80 for example), give a name to the image ("ListOfProducts" for example) and validate.



Notes

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 of the first line found in the Multiline Zone control:
    1. Click the first line and select the Static control.
    2. Display the description window (double-click the control).
    3. In the "General" tab of the description window:
      - Give a name to the control ("STC\_ListOfProducts" for example).
      - Change the caption ("List of products" for example).
    4. In the "Style" tab of the description window, change the color of the font: choose black.
    5. Validate the description window.



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

- ▶ To modify the image of the second line found in the Multiline Zone control:
  1. Click the second line and select the Image control.
  2. Display the description window of the image (double-click the control).
  3. In the description window:
    - Give a name to the image ("IMG\_MapOfStores" for example).
    - Click the "Catalog" button in order to choose an image representing the action.
    - In the window of the image catalog, type "Map" in the search control and press the "Enter" key.

- Select an image via a double click.
- In the window for configuring the generated image, select a size (80 for example), give a name to the image ("MapOfStores" for example) and validate.

4. Validate the description window.

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

1. Click the second line and select the Static control.
2. Display the description window (double-click the control).
3. In the "General" tab of the description window:
  - Give a name to the control ("STC\_MapOfStores" for example).
  - Change the caption ("Map of stores" for example).
4. In the "Style" tab of the description window, change the color of the font: choose black.
5. Validate the description window.

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

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

1. Click the third line and select the Image control.
2. Display the description window of the image (double-click the control).
3. In the description window:
  - Give a name to the image ("IMG\_Exit" for example).
  - Click the "Catalog" button in order to choose an image representing the action.
  - In the window of the image catalog, type "Close" in the search control and press the "Enter" key.
  - Select an image via a double click.
  - In the window for configuring the generated image, select a size (80 for example), give a name to the image ("Close" for example) and validate.
4. Validate the description window.

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

1. Click the third line and select the Static control.
2. Display the description window (double-click the control).
3. In the "General" tab of the description window:
  - Give a name to the control ("STC\_Exit" for example).
  - Change the caption ("Exit from the application" for example).
4. In the "Style" tab of the description window, change the color of the font: choose black.  
Validate the description window.

## Programming the menu

► We are now going to write the code used to perform each menu action:

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 included inside it.

2. In the code editor, in the "Selection (click) of a line in..." section, enter the following code:

```
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
    EndProgram()
END
```

3. Save the modifications by clicking  among the quick access buttons.

4. Close the code window (click the cross at the top right of the code editor).

## Running the test of the application

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 test of the project and define the first project window.

► To define the first project window:

1. Select the "WIN\_Menu" window in the project explorer.

2. Display the popup menu.

3. Select "First project window". A specific icon (with a small 1) is displayed in front of the name of the window, in the project explorer.

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

► To run the test of the project:

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 run its test. The procedure to follow is the same as the one presented in "Using the Map control", page 69.

## LESSON 2.5. DISTRIBUTING THE APPLICATION

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

---

- Generating the APK
- Available distribution modes



Estimated time: 20 min

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

## Deployment via Android Market/Google Play

---

Google Play (formerly Android Market) is an online service used to download applications (free of charge or not) on mobile 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.

Publishing applications on Google Play must comply with specific rules:

During the first publication, you must register toward 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 not mandatory. The validity period of the 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 the application" screen) 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.

**Note:** Google Play is the most common application but other ones 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.
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 the 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 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 (<http://developer.android.com/intl/fr/guide/developing/tools/adb.html>)

**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 the phone, in the "Applications" sub-menu.

## **Copying the application onto the 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 the application onto the device (external memory for example).
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.

**Note:** 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 the phone, in the "Applications" sub-menu.

## LESSON 2.6. QUESTIONS/ANSWERS

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

- Questions/Answers



Estimated time: 20 min

## Questions/Answers

### Question How do I use the messaging of the device?

You can send messages by using the native application for sending emails found on the device. To do so, you must:

- initialize the information of the Email structure.
- use *EmailRunApp*.

### Question How do I send an email directly?

You have the ability to send emails by using the email functions of WLanguage.

Example for sending emails:

```
MyMessage is Email

// Reset the variable to zero
EmailReset(MyMessage)

MyMessage..Sender = EDT_Sender
MyMessage..Subject = EDT_SUBJECT
MyMessage..Message = EDT_BODY
Add(MyMessage..Recipient, EDT_Recipient_Addr)

// Start a SMTP session
MySession is EmailSMTPSession
MySession..ServerAddress = sServer
IF EmailStartSession(MySession) = False THEN RETURN

// Send the message
EmailSendMessage(MySession, MyMessage)

// Close the session
EmailCloseSession(MySession)
```

### Question How do I establish the dialog with an FTP server?

The Android system allows you to connect to an FTP server to send or retrieve files. To do so, you must use the following functions:

- *FTPConnect* to connect to the FTP server
- *FTPDisconnect* to disconnect
- *FTPSend* to send a file from the device to the FTP server
- *FTPGet* to get a file from the server to the device.

Example for sending files onto an FTP server:

```
// Connection to an FTP server
nConnection is int
nConnection = FTPConnect("Address of FTP server")
ResSend = FTPSend(ConnectionNum, "/sdcard/photo.jpg", ...
                  "/Temp")

// Disconnection
FTPDisconnect(nConnection)
```

Caution: Using the FTP functions modifies the permissions required by the application (Permission required: INTERNET). This permission allows the applications to open the network sockets.

### Question

### How do I use the GPS of the device and the geo-localization functions?

The Android system includes functions for managing the GPS and for geo-localization. According to the device, the GPS functions will use the GPS chip of the device or a triangulation system by using the wireless networks.

To do so, you must use the following functions:

- **GPSInitParameter** to initialize the GPS (choice of localization system...)
- **GPSGetPosition** to retrieve the current position
- **geoDistance** to retrieve the distance between 2 positions
- **geoRunApp** to start the native mapping application of the device
- **geoGetAddress** to retrieve an address or a list of addresses in relation to a location
- etc.

Example for using the GPS functions:

```
// Find a localization provider
// that can give information about the speed
// while consuming an average amount of energy and
// while providing a high level of precision
GPSInitParameter(gpsAuto, gpsPrecisionHigh,
                gpsSpeed + gpsEnergyMedium)
MyPosition is geoPosition
MyPosition = GPSGetPosition()
Info("Latitude: " + MyPosition..Latitude)
Info("Longitude: " + MyPosition..Longitude)
```

Caution: Using the GPS functions modifies the permissions required by the application (Permission required: ACCESS\_FINE\_LOCATION).

**Question**    **How do I send an SMS from my Android application?**

The Android system allows you to send or read SMSs on a phone. To do so, you must use the following functions:

- initialize the information of the SMS structure
- **SMSSend** to send an SMS
- **SMSNbMessage** to find out the number of SMSs
- etc.

Example for sending an SMS:

```
// Initialize the SMS structure for a standard number
SMS.Number = "0600000000"
SMS.Message = "I am sending SMSs with WinDev Mobile!"

// Send the SMS
ResSend = SMSSend()

// Has an error occurred?
IF ResSend = False THEN
    Error(ErrorInfo(errMessage))
END
```

Caution: Using the SMS functions modifies the permissions required by the application (Permission required: SEND\_SMS).

**Question**    **How do I call native JAVA code from my Android application?**

When you develop an Android application WinDev Mobile, the native code is in JAVA. The WLanguage is translated into JAVA when compiling and generating the application. You have the ability, from the WinDev Mobile environment, to include native procedures in JAVA language. To do so, you must:

- create a set of global procedures
- create a global procedure
- click the gray bar of the procedure header (where the WL word is found).
- the header bar changes and it becomes purple. The WL word is transformed into Java.

Then, all you have to do is enter native JAVA code or copy JAVA code.

Caution: Make sure that the code entered or copied is written properly: WinDev Mobile does not check the completion. The errors will occur when compiling the project.





**PART 3**

iOS application

**10**

**Express**



**DEVELOP 10 TIMES FASTER**

**PCSOFT**



## 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 min

## Overview

---

To start working with WinDev Mobile, 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 with WinDev Mobile.

Before creating our first project for iOS, a configuration of the development computer is required.

## 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 19 is a 32-bit Windows application that can be used in Windows XP, Vista, 7, ..

The application will be created on the PC before it is compiled on a Mac (project, analysis, windows, ...).

This PC requires no setup of Mac/Apple tools.

### Why a Mac?

A Mac is required because the project generated on the PC must be compiled in a specific compiler to generate the iOS applications. The minimum version of the operating system must be version 10.6 (Snow leopard).

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 recommended version of Xcode is version 4.3.

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.



EXAMPLE

Answer

A corrected project is available. To open this project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My iOS project (Answer)".

- ▶ To create a project:
  1. Start WinDev Mobile 19 (if not already done). Close (if necessary) the current project to display the home window.
  2. In the home window, click the "Create a project" icon then "iPhone/iPad application".



3. The wizard for project creation starts. The different wizard screens help you create your project. The information specified in this wizard can be modified later.



Notes

Tip: To create a project, you can also:

1. Click  among the quick access buttons of the WinDev Mobile menu.
2. Click "Project" in the wheel that is displayed.

4. The first wizard screen is used to enter the name of the project, its location and its description.

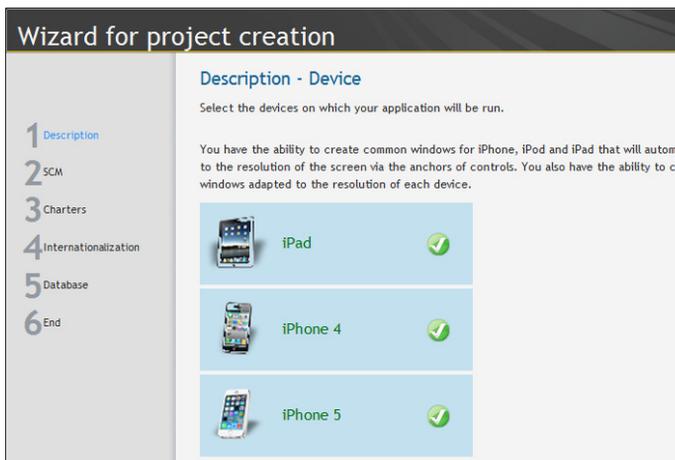


In our case, this project will be named "My\_iOS\_Project". WinDev Mobile proposes to create this project in the "\My Mobile projects\My\_iOS\_Project" directory. You can keep this location or modify it via the [...] button.

5. Go to the next screen via the arrows found at the bottom.

6. The wizard proposes to add documents. Keep the default options and go to the next screen.

7. The next screen is used to select the device for which the application is developed. In this example, we will keep "iPad", "iPhone4" and "iPhone5".



8. In the left section of the wizard, click "Charters". This step is used to define the programming charter. Don't modify the suggested options. Go to the next screen via the arrows found at the bottom.

9. This step is used to define the style book. Select "ActivPhone 5".

10. The other wizard steps not being important for our first project, click "End" in the left section of the wizard.

11. Click the validation button at the bottom of the wizard. The project is automatically created. WinDev Mobile displays the different possible actions.

12. Click "WinDev Mobile editor".

## 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, too simple, but we recommend that you 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. Click  among the quick access buttons of the WinDev Mobile menu:



2. A window shaped like a wheel is displayed. This window is used to create all the elements that can be associated with a project.
3. Click "Window". The wizard for window creation starts.
4. Select "Blank for iPhone" in the list of windows displayed on the left. In the list of skin templates found on the right, the "ActivPhone 5" skin template is selected by default. You can choose another skin template proposed in the list.

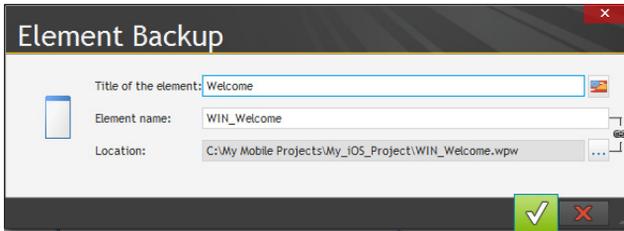


Note

The skin templates allow you to quickly create outstanding interfaces. A skin template defines the style of the window but also the style of all the controls that will be used in this window. No ugly interface anymore.

5. Validate. The window is automatically created in the editor.

- ▶ Save the window by clicking  among the quick access buttons. During the first backup, a specific window is displayed. This window proposes to enter:



- the title of the element: enter "Welcome". In our case, this title will be displayed in the title bar of the window.
- the name of the element is the name of the window. This name will be used in programming. By default, this name includes "WIN\_" that corresponds to the programming charter and "Welcome" that corresponds to the title of the window.



Note

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 uses a programming charter.

The programming charter 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 don't want to use this charter, all you have to do is disable it: on the "Project" pane, in the "Other actions" group, expand "Charter" and uncheck "Use the charter".

- the location that corresponds to the file name created for the window. The window is a file whose extension is "WPW", saved in the project directory.
- ▶ Click the green button to validate the backup of the window.

## Displaying a message

You are now going to create a button used to display a message.

- ▶ To create the "Display" button:
  1. On the "Creation" pane, in the "Usual controls" group, click . The button appears in creation under the mouse.
  2. Move the mouse toward the position where the control must be created in the window (at the top of the window for example). To drop the control in the window, all you have to do is perform a new left mouse click.
  3. Perform a right mouse click on the control that was just created. The popup menu of the control is displayed. Select "Description" from this popup menu. The description window of the button is displayed.

- ▶ Modify the characteristics of the control by entering the following information:



1. Name of this control: "BTN\_Display".
2. Caption of this control: "Display"

- ▶ Validate the description window of the control (green 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**.



Notes

WLanguage is the programming language supplied with WinDev Mobile. It is a 5th generation language (5GL) that includes highly sophisticated commands.

1. Select the "Display" button with the mouse: all you have to do is click it.
2. Display the popup menu of the control (right mouse click).
3. Select "Code". This option opens the code editor of WinDev Mobile, in which all the WLanguage statements can be entered.
4. Enter the following code in the "Click BTN\_Display" process:

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

**Note about the assisted input:** As soon as the first two characters are typed, WinDev Mobile proposes all the words of the WLanguage vocabulary containing these characters. The assisted development is a very powerful feature. No more mistake when typing 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.



Notes

When typing this code in the code editor, you have noticed that different colors are used by the different elements. 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.

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

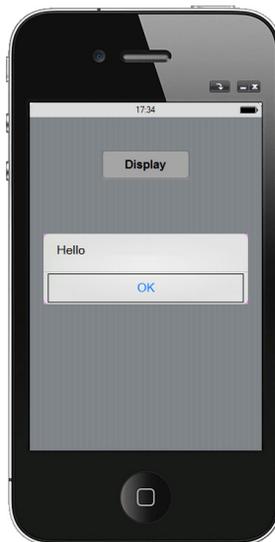
- ▶ Save the modifications by clicking  among the quick access buttons (on the left of ribbon) or by pressing [Ctrl]+[S].
- ▶ Close the code editor (cross at the top right of the code editor). The window re-appears.

## First test

---

For an iOS application, WinDev Mobile allows you to run the test of the application on the development computer via the simulation mode. This test simulates an iOS device on the development computer. This test is useful when the developer has no Mac device to compile the application. However, this test does not allow you to use the hardware components of the device (GPS, SMS, camera, ...).

- ▶ Let's now run the test of the window 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.



- ▶ Any developer knows that running a program test can be a long and tiresome job. In WinDev, a SINGLE CLICK allows you to run the test of the 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 editor of WinDev Mobile is redisplayed.

## 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 the Mac in order to compile it.
- Connect your device to your Mac and compile the Xcode project in order to generate the program for the connected device.
- Then, the program will be installed on the device. 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 also have the ability to click **iOS** among the quick access buttons).
2. WinDev Mobile proposes to select the first project window for the iPhone and iPad platforms. In our example, select the "WIN\_Welcome" window and validate (green button at the bottom of the screen).
3. The wizard for generating an iPhone/iPad application starts.



4. The first wizard screen is used to:

- define the name of the application, the company and the copyright.
- enter the bundle identifier.



#### Notes

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 standards of Apple. This identifier can be modified.

5. Go to the next screen. Specify the path of the different icons found in the application. Several icons must be provided:

- icons for iPad (iPad, iPad mini or iPad with the Retina screen)
- icons for iPhone (the iPhone 4 has a different resolution with the Retina screen).

6. Go to the next screen. Specify the path of the different images used when starting the application for iPad and for iPhone (images in PNG format). Default images are automatically created for your application.

7. Go to the next screen. Specify the version number of the generated application.

8. Go to the next screen. This screen 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 screen. This screen is used to specify:

- the minimum iOS version required to run the application.
- 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.

10. Validate the wizard. The generation is performed in the EXE folder of the project directory. The directory containing the source codes 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 the Mac. We won't present these operations here. See the online help for more details.

## LESSON 3.2. INTERFACE (GUI)

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

- Choosing the resolution according to the device
- Orientation of the window
- Management of touchscreen



Estimated time: 20 min

## 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 do not know the target devices:** 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).
  - develop 2 sets of windows, one for iPhone, one for iPad.
  - use the layouts to create different interfaces according to the runtime platform and to its resolution.

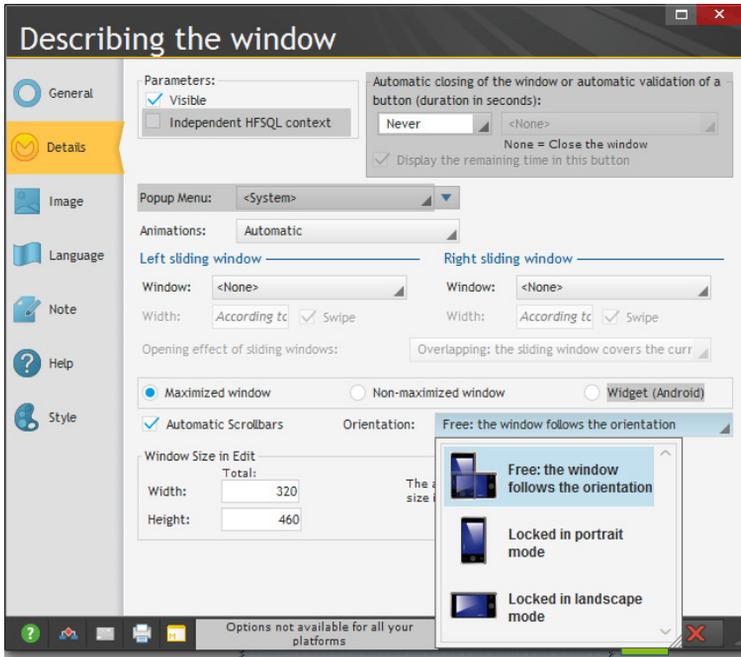
## Orientation of the window

---

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

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

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



No specific operation must be performed in the two first cases.

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.



EXAMPLE

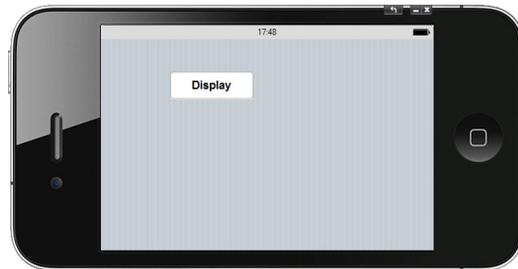
Answer

A corrected project is available. This project contains the different windows created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My iOS project (Answer)".

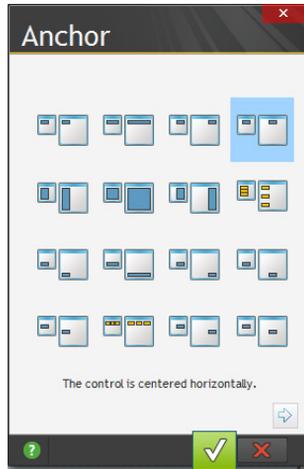
In our example, the project was created for a phone and its test was run in portrait mode in the simulator.

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

- ▶ Run the test of the project (📱 among the quick access buttons).
  1. The window is displayed in portrait mode.
  2. In the simulator, click the arrow in the shell (👉).
  3. The orientation of the window changes on the screen.
  4. In our example, the location of the button does not change: it does not adapt to the orientation of the screen.



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



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

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

## Management of touchscreen

---

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

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

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

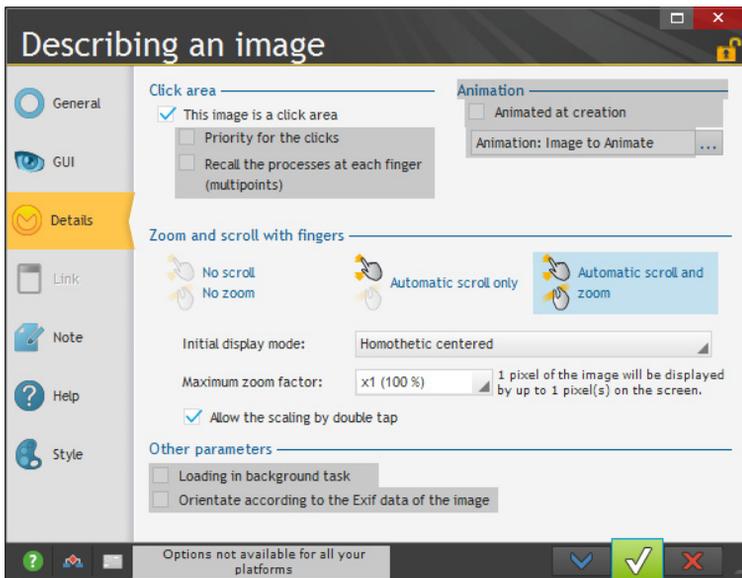
- ▶ Open (if necessary) the "My\_iOS\_Project" project that was created in the previous lesson.



Answer

A corrected project is available. To open this project, on the "Home" pane, in the "Online help" group, expand "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 must be created. To drop the control in the window, all you have to do is perform a new left mouse click.
  4. Double-click the Image control: the description window of the control is displayed.
  5. In the "General" tab, select an image found on your disk in the "Image" edit control via the file picker.
  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 managing the multi-touch are displayed:



8. Select "Automatic scroll and zoom".
9. Validate the description window of the control.
10. Save the window (click  among the quick access buttons).
11. A GUI error appears in the error pane: the automatic scrollbars of the window are in conflict with the scroll features of the Image controls.

**12.** To avoid this GUI error, disable the scrollbars of the window:

- Display the description window of the window ("Description" from the popup menu).
- In the "Details" tab, uncheck "Automatic scrollbars".
- Validate the description window.

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

**14.** Close the project.

## The different types of controls

---

WinDev Mobile proposes several controls to communicate with the user. 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 zones to create GUI similar to the native iOS windows.
- the Map control to view a position on a map or an itinerary.

Some of these controls are presented in the "Advanced programming" lesson.

## LESSON 3.3. DATABASES

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

---

- Available databases
- Synchronization



Estimated time: 10 min

## The available databases

---

### Overview

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

### HFSQL database

#### HFSQL Classic

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

In this case, the application is stand-alone. No Wi-Fi or 3G connection is required.

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

In the "Advanced programming" lesson, we will develop an application that uses a HFSQL Classic database.

#### 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 the 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.

## The synchronization

---

The synchronization mechanism is used to "synchronize" the data stored on a mobile device with the data stored on a server. The synchronization uses the mechanism of "universal replication".

This technique is available in WinDev, WebDev and WinDev Mobile.

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

## **Accessing data via a Webservice**

---

Another method can also be used to access the data on a server from a mobile device: the call to a Webservice.

In this case, the mobile device must be equipped with a Wi-Fi or 3G connection to connect to the Webservice.

The mobile application does not directly access the database. The application calls the functions of the Webservice. These functions return the data.

It is the Webservice that accesses the database.

This technique is used to have a business layer (the Webservice) common to several types of applications and interfaces (WinDev, WebDev or WinDev Mobile) and different types of operating systems (Windows, Android, iOS, ...).

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

## LESSON 3.4. ADVANCED PROGRAMMING

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

---

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



Estimated time: 1 h 30 min

## Overview

In this lesson, we are going to develop an iOS application that uses a HFSQL Classic database. This application will allow us to present some specific features of the iOS programming.

### Opening the project

- ▶ Start WinDev Mobile 19 (if not already done). Close (if necessary) the current project to display the home window.
- ▶ Open the "iOS Managing Products" project.  
To do so, in the home window, click "Tutorial" and select the "iOS Managing Products (Exercise)" project.  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "iOS Managing Products (Exercise)".



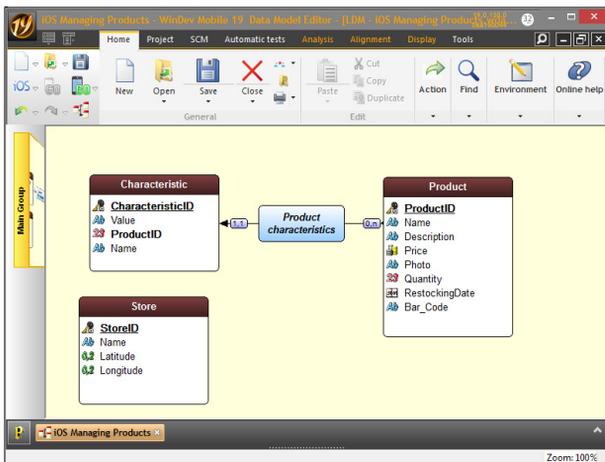
Answer

A corrected project is available. This project contains the different windows created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "iOS Managing Products (Answer)".

### Describing the project

Let's take a look at our source 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 the WinDev Mobile menu.
  2. The data model editor is displayed.



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

We are now going to develop our application.



Caution!

In this section, the different tests will be run in simulation mode in WinDev Mobile. Indeed, deploying the application on iPhone or iPad is quite long and a Mac is required.

The different steps for generating the Xcode project will be presented in details at the end of this lesson. Then, this project can be compiled on a Mac.

## 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.

### Creating the window

- ▶ To create a new window:
  1. Click  among the quick access buttons. Click "Window" in the wheel that is displayed. Choose "Blank for iPhone" and validate the wizard.
  2. Save the window by clicking  among the quick access buttons.
  3. In the window that is displayed, enter the title of the window: "List of products". The name and location of the window are automatically filled. Keep the default values and validate (green button).
  4. The window is added to the project.

### Creating the Looper control

A "Looper" control will be used to display the list of products. The main information about the products will be displayed in this control.

- ▶ To create the "Looper" control:
  1. On the "Creation" pane, in the "Data" group, expand "Looper" and select "Looper (vertical)". The control appears under the mouse.
  2. Click the position where the control must be created in the window (at the top for example). The wizard for creating a Looper control starts.
  3. In the wizard, select "Display the data coming from a file or from a query". Go to the next screen.
  4. Select the "Product" file. Go to the next screen.

5. Select the items to display:



6. Keep the "Name", "Description" and "Price" items (a checkmark must be found in front of these items). Go to the next screen.

7. Keep the proposed sort item (ProductID). The products will be sorted in the Looper control according to this item. Go to the next screen.

8. In the "Additional parameters" screen, keep the default options. Go to the next screen.

9. Keep the default name ("LOOP\_Product") and validate.

10. A message regarding the management of scrollbars in the window is displayed.

Indeed, both the window and the Looper 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 Looper control.

The vertical scroll will be performed in the Looper control and not in the window. Therefore, controls can be positioned above or below the Looper control without scrolling.

11. Click "Disable the automatic scrollbar". The Looper control appears in the window editor.

12. Increase the Looper control if necessary.

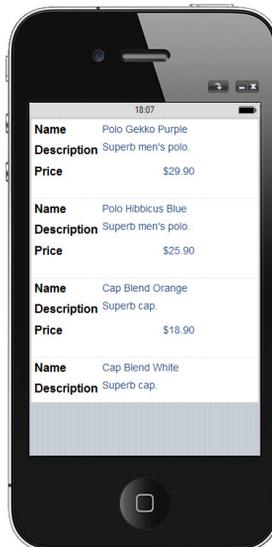


Notes

The data automatically appears in the window displayed in the editor. This concept is called "Live Data": you see the data found in your 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 view the result. Click  among the quick access buttons (or press [F9]).



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

## Improving the window

As you have noticed during the test, some controls do not display the entire data. We are going to resize the controls.

- ▶ To resize the Description control:
  1. Select the "Description" control.
  2. In the editor, enlarge the "Description" control.

The importance of anchors was presented in the previous lesson. We are now going to apply an anchor to our Looper control so that it is automatically resized according to the orientation of the screen.

- ▶ To anchor the Looper control:
  1. Select the Looper control.
  2. Display the popup menu of the control (right mouse click) and select "Anchor".
  3. In the window that is displayed, select the option .
  4. Validate. The red arrows indicating the anchoring options appear in the window.
- ▶ Run the window test to see the result.
  - Click  among the quick access buttons.
  - Modify the orientation of the window.
  - Close the simulator to go back to the editor.
- ▶ Our window is now created.

## Creating the form window

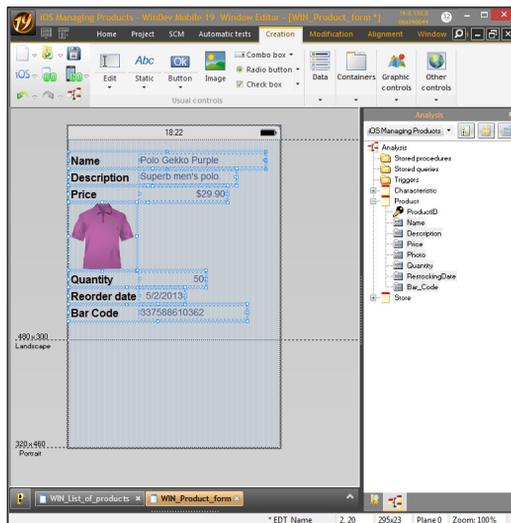
We are now going to create a new window used to display the product form. Then, this window will be started from the list of products to display the details of the selected product.

### Creating the window

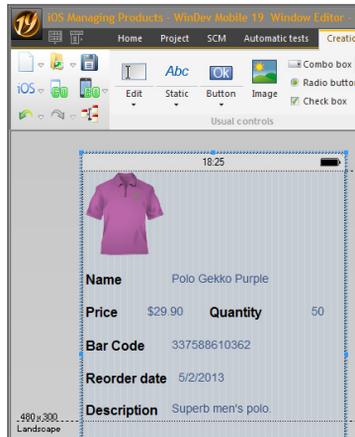
- ▶ To create the form window:
  1. Create a new blank window. Click  among the quick access buttons. Click "Window" in the wheel that is displayed. Choose "Blank for iPhone" and validate the wizard.
  2. Save the window. Specify the title of the window: "Product form". Its name is automatically proposed: "WIN\_Product\_form". Validate.

### Creating the controls

- ▶ To create an edit control:
  1. Display the "Analysis" pane if necessary (on the "Home" pane, in the "Environment" group, expand "Panes" and select "Analysis"). The different data files described in the "iOS Managing Products" analysis appear in the pane.
  2. With the mouse, select the items of the "Product" 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" and "Description" so that they are visible in the window.
5. Reorganize the controls in the window. Respect the following order: "Photo", "Name", "Price", "Quantity", "BarCode", "Reorder Date", "Description".



6. We are going to view the navigation order in the window: press the [F5] key. The numbers that are displayed represent the navigation order in the window. Press [F5] again to make the numbers disappear. To correct the navigation order, on the "Window" pane, in the "Order" group, expand "Navigation" and select "Define automatically".

7. Save the window.

- ▶ Run the test of the window (  among the quick access buttons). The window is displayed with empty controls.
- ▶ To display the product data:
  1. Display the processes associated with the window:
    - Perform a right mouse click in the area beside the window
    - Select "Code" from the popup menu.
    - The code editor appears.
  2. Enter the following code in the "End of initialization of WIN\_Product\_form" process:

```
FileToScreen ()
```

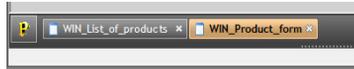
**FileToScreen** is used to display in the controls the data found in the data file, for the current record.

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

## Displaying the form from the list of products

Let's see how to display the form of the 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 found in the button bar:



2. Right-click the Looper control and select "Code" from the popup menu.
3. In the code window that is displayed, enter the following code in the "Selecting a row of..." process:

```
OpenMobileWindow(WIN_Product_form)
```



### Notes

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 the existing windows found in the project. All you have to do is select the window with the keyboard or with the mouse.

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

4. Save the modifications by clicking  among the quick access buttons.
  5. Close the code window (click the cross at the top right of the 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.
    - The detailed window of the product is displayed.
  - ▶ Close the simulator.

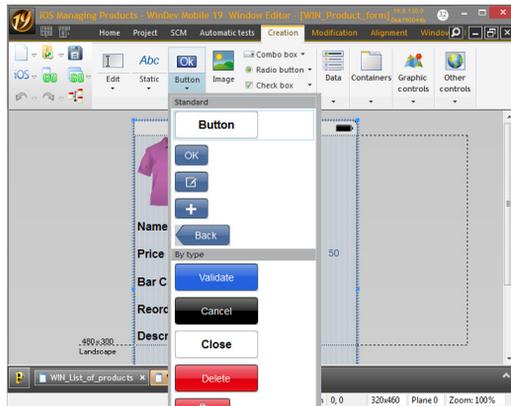
## Managing the creation and the modification of a product

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

### Modifying the product form

We are going to add 2 buttons into the "WIN\_Product\_form" window:

- a "Validate" button to manage the validation of modifications
  - a "Cancel" button to manage the return to the list of products.
- ▶ Display (if necessary) the "WIN\_Product\_form" window in the editor: click the corresponding button in the button bar.
- ▶ To create the "Cancel" button:
1. On the "Creation" pane, in the "Usual controls" group, expand "Button" (click the arrow found below ).
  2. The list of preset buttons is displayed.

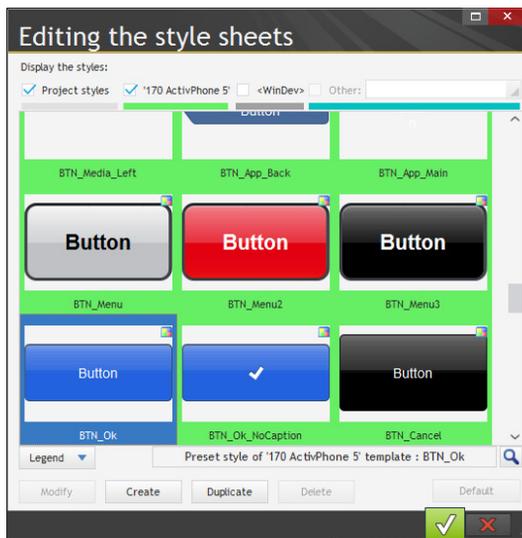


#### Notes

For some types of controls, WinDev Mobile proposes a list of preset controls. These controls are advanced controls, configured to perform a specific action. For example, the "Cancel" button proposed by the list of preset controls contains the code required by its execution.

3. Click "Cancel": the shape of the button appears under the mouse. Then, click at the bottom right of the window to create the button.

- ▶ We now want to add the "Validate" button and customize the style of this button.
  1. On the "Creation" pane, in the "Usual controls" group, expand "Button" (click the arrow found below ).
  2. The list of preset buttons is displayed.
  3. Click "Button": the shape of the button appears under the mouse. Then, click at the bottom of the window to create the button (on the left of the "Cancel" button).
  4. Select the control and press the "Enter" key on the keyboard. The caption of the button switches to edit. Type "Validate" and press the "Enter" key on the keyboard.
  5. Resize the button if necessary (with the handles) in order for the caption to be entirely displayed in the button.
- ▶ To customize the style of this button:
  1. Select the "Validate" button: all you have to do is click it.
  2. Display the popup menu (right mouse click) and select "Choose a style".
  3. Each skin template proposes styles for the standard buttons: validation, cancelation, creation, ... Choose the "BTN\_Ok" style.



4. Validate. This style is automatically applied to the button.
- ▶ We are now going to enter the code of the "Validate" button.
    1. Right-click the button and select "Code" from the popup menu.
    2. In the "Click" process, enter the following code:

```
ScreenToFile ()
HModify (Product)
Close ()
```

Let's take a look at this code:

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

3. Save the modifications by clicking  among the quick access buttons.
4. Close the code window (click the cross at the top right of the 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 "Closing a child window" process of the "WIN\_List\_of\_products" window.

1. Display the "WIN\_List\_of\_products" window in the window editor.
2. Right-click the window background and select "Code" from the popup menu. Enter the following code in the "Closing a child window" process:

```
LooperDisplay (LOOP_Product)
```

Let's take a look at this code:

- The "Closing a child window" process 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 the "WIN\_List\_of\_products" window.
3. Save the modifications by clicking  among the quick access buttons.
  4. Close the code window (click the cross at the top right of the code editor).

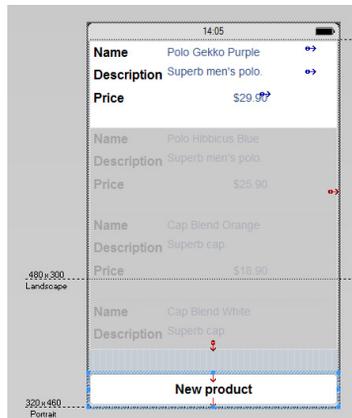
- Run the test of the "WIN\_List\_of\_products" window in the simulator ( among the quick access buttons).
- In the list of products, click one of the products with the mouse: for example, the "Polo Hibiscus Blue" product whose price is 25.90 Dollars.
  - The detailed window of the product is displayed. Modify the price of 25.90 Dollars and enter 19.90 Dollars and click the "Validate" button.
  - When going back to the list of products, you will notice that the price was updated for this article.
- Close the simulator. The editor of WinDev Mobile 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 the "Product form" window to manage the addition into the Product data file.
- To add a button into the list of products:
1. Display the "WIN\_List\_of\_products" window in the editor.
  2. On the "Creation" pane, in the "Usual controls" group, click : the shape of the button appears under the mouse. Then, click at the bottom of the window to create the button.

3. Select the control and press the "Enter" key on the keyboard. The caption of the button switches to edit. Type "New product" and press the "Enter" key on the keyboard.



4. The code of this option is used to open the "Product form" window and to reset its controls. To enter this code:

- Right-click the control.
- Select "Code" from the popup menu.
- In the "Click" process, enter the following code:

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

**HReset** initializes the variables of the items found in the Product file with the default values to manage a new record.

5. Save the modifications by clicking  among the quick access buttons.
6. Close the code window (click the cross at the top right of the code editor).

► We are now going to modify the window of the product form to manage the addition of a new record.

1. Display the "WIN\_Product\_form" window in the editor.
2. We are going to modify the code of the validation button:
  - Right-click the button and select "Code" from the popup menu.
  - In the "Click" process, replace the existing code by the following code:

```
ScreenToFile ()
IF Product..NewRecord THEN
  HAdd (Product)
ELSE
  HModify (Product)
END
Close ()
```

Let's take a look at this code:

- **..NewRecord** is used to find out whether the current record must be created.
  - If **HReset** was called beforehand, the property returns True (case of the 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**.
3. Save the modifications by clicking  among the quick access buttons.
  4. Close the code window (click the cross at the top right of the 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.
  - Enter a new product.
  - Validate. The new product appears in the list of products.
  - Close the simulator.

## Using the camera and displaying an image



Caution!

In this section, the different tests will be run in simulation mode in WinDev Mobile. Indeed, deploying the application on iPhone or iPad is quite long and a Mac is required.

The different steps for generating the Xcode project will be presented in details at the end of this lesson. Then, this project can be compiled on a Mac.

### Overview

We are going to manage the photo of the product. We will be using the camera of the device.

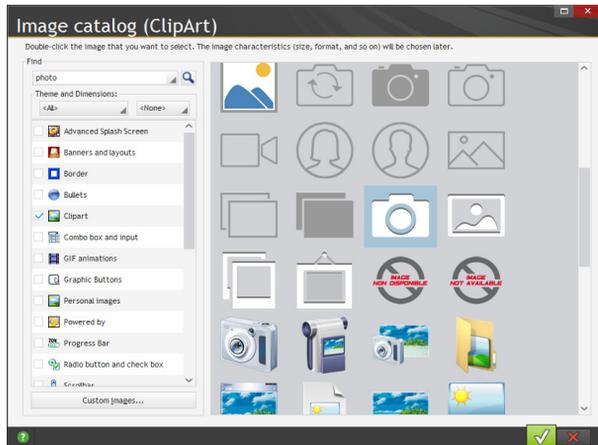
We are going to create a button to start the camera. The photo will be retrieved in the format of an image in memory and displayed in the image control of the product.

We are also going to create a button used to select a photo in the album of the mobile device.

### Creating the button for taking photos

- To create the button for taking photos:
1. Display the "WIN\_Product\_form" window in the editor.
  2. Add a button into the window:
    - On the "Creation" pane, in the "Usual controls" group, click : the shape of the button appears under the mouse.
    - Then, click at the top right of the product image to create the button.

- We are going to modify the button in order to associate it with an image representing a camera:
1. Select the button and display its popup menu (right mouse click).
  2. Select "Description". The description window of the control is displayed.
  3. In the "General" tab, position on the "Image" edit zone and click the "Catalog" button.
  4. The image catalog of WinDev Mobile is opened. This catalog contains hundreds of images in different fields, formats and sizes.
  5. In the "Search" area (top left), type "photo" and validate. Several images are displayed:



6. Double-click the image to select it.
7. In the next screen, you have the ability to choose the size of the image, the color, the orientation, its format and its name.
  - Reduce the lightness to -100%.
  - Specify the name of the image ("Camera").
8. Validate the window.
9. The path of the image is displayed in the button description.
10. Give a name to the button: "BTN\_Camera".
11. Clear the caption of the button.
12. Validate the description window.
13. In the editor, reduce the size of the button.

## Taking photos

We are going to enter the code used to take a photo and to display it in the Image control of the product form.

► To manage the taking of photos:

1. Select "Code" from the popup menu of the button (right mouse click).
2. Enter the following code in the "Click" process:

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

In this code **VideoStartApp** is used to start the native camera application of the device in order to save a video or to take a photo.

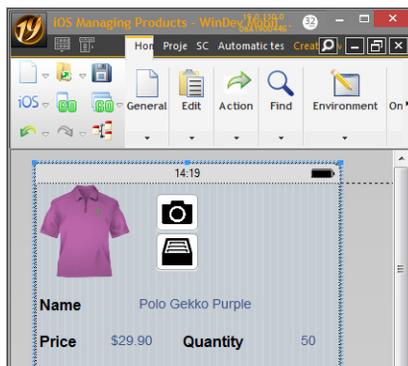
3. Save the modifications by clicking  among the quick access buttons.
4. Close the code window (click the cross at the top right of the code editor).

## Selecting a photo in the photo album

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

► To create the button for selecting the photo:

1. Add a button into the "WIN\_Product\_form" window:
  - on the "Creation" pane, in the "Usual controls" group, click .
  - the shape of the button appears under the mouse.
  - create the button below the previous button.
2. Display the description window of the button (double-click the control):
  - Give a name to the button: "BTN\_PhotoAlbum".
  - Clear the caption of the button.
  - Select an image in the image catalog.
3. Validate the description window.
4. In the editor, reduce the size of the button.



► The code of this button is used to open the photo album of the device and to select an image in order to display it in the Image control of the product form.

1. Select "Code" from the popup menu of the button (right mouse click).
2. Enter the following code in the "Click" process:

```
// Local variable
sPhoto is string
// Start the selection
// The SelectedImage procedure will be run
// when an image is selected in the album
sPhoto = AlbumPicker(SelectedImage)
```

In this code, **AlbumPicker** is used to retrieve the photo selected in the photo album. This function calls the "SelectedImage" procedure. This procedure is used to assign the Image control with the selected image.

3. To create the procedure:
  - Select the "SelectedImage" word.
  - Display the popup menu (right mouse click).
  - Select "New .. New local procedure".
  - Validate the name of the procedure.
  - Enter the following code:

```
PROCEDURE SelectedImage(sImagePath)
// If an image is selected
IF sImagePath <> "" THEN
// Displays the image in the Image control
IMG_Photo = sImagePath
END
```

Let's take a look at this code: if an image is selected (which means if the path of the image does not correspond to an empty string), then the path is assigned to the Image control.

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

## Managing the bar code of the product



Caution!

In this section, the different tests will be run in simulation mode in WinDev Mobile. Indeed, deploying the application on iPhone or iPad is quite long and a Mac is required. This deployment will be proposed at the end of this lesson.

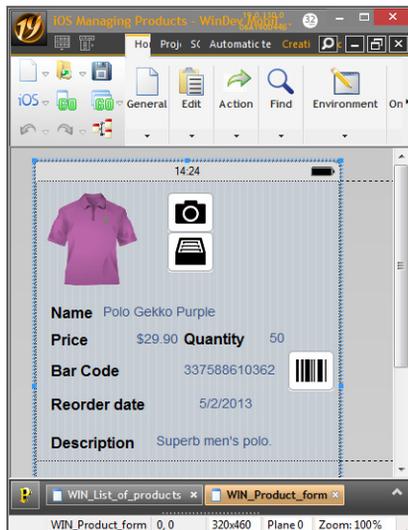
### Overview

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 camera of the device and by a specific WLanguage function.

## Implementation

- ▶ To create the button for managing the bar codes:
  1. Display (if necessary) the "WIN\_Product\_form" window in the editor.
  2. Add a button into the window:
    - on the "Creation" pane, in the "Usual controls" group, click .
    - the shape of the button appears under the mouse.
    - create the button beside the "Bar Code" control (resize the edit control if necessary).
  3. Display the description window of the button (double-click the control):
    - Give a name to the button: "BTN\_BarCode".
    - Clear the caption of the button.
    - 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 size of the button.



- ▶ The code of this button is used to scan the bar code.
  1. Select "Code" from the popup menu of the button (right mouse click).
  2. Enter the following code in the "Click" process:

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

In this code, **BCCapture** is used to decode the information stored in a bar code by using the camera of the device.

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

## 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 file contains the addresses of 2 stores that will be localized on a map via the mapping functions.

### Creating the window

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

► To create the window:

1. Click  among the quick access buttons. Click "Window" in the wheel that is displayed. Choose "Blank for iPhone" and validate the wizard.
2. Save the window. Specify the title of the window: "Map of stores". Its name is automatically proposed: "WIN\_Map\_of\_stores". Validate.

### Creating the Map control

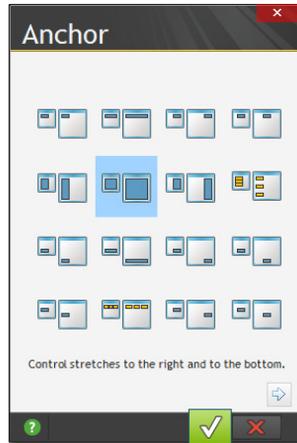
► To create the Map control:

1. On the "Creation" pane, in the "Graphic controls" group, click "Map". The shape of the button 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. Display the description window of the "Map" control (double-click the control for example).
6. In the description window of the control, specify the name of the control ("MAP\_Store") and validate.
7. In the editor, resize the control (with the handles) so that it occupies the entire window.

In order for this window to be usable both in portrait mode and in landscape mode, we are going to define anchors on the Map control: the size of the control will automatically adapt to the orientation of the window.

► To define anchors on the Map control:

1. Select the Map control (click below).
2. Display the popup menu (right mouse click) and select "Anchor".



3. Select "the control stretches to the right and to the bottom".
4. Validate the anchoring window. Two red arrows appear in the editor.
5. Save the modifications by clicking  among the quick access buttons.

## Displaying the stores on the map

### Principle

We are now going to add the code used to display on a map all the stores found in the "Store" file. To do so, the "Store" file will be browsed by 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 different information. We will be using the following information:

- Name
- Latitude
- Longitude

**MapAddMarker** is used to add a marker onto the map. Then, all you have to do is define a sufficient zoom level to see all the markers on the map. If the zoom is not set properly, some markers may not be visible or they may overlap on the map.

### Implementation

- ▶ To enter the code used to display the stores:
  1. Right-click outside the window. Select "Code" from the popup menu. The processes associated with the window are displayed.

2. Enter the following code in the "Global declarations of..." process.

```
// Global variables
gMarker is Marker

// Load the stores
FOR EACH Store
  // Coordinates of marker
  gMarker.Position.Latitude = Store.Latitude
  gMarker.Position.Longitude = Store.Longitude
  // Name of marker
  gMarker.Name = Store.Name
  // Add the marker
  MapAddMarker (MAP_Store, gMarker)
END
// Default zoom
MAP_Store..Zoom = 7
MapDisplayPosition (MAP_Store, gMarker.Position)
```

3. Save the modifications by clicking  among the quick access buttons.

4. Close the code window (click the cross at the top right of the code editor).

A test can be run in the simulator but the mapping functions can only be run on the device. Indeed, this feature requires functions specific to the device.

### Close the window

We are going to create a button to close this window. To do so, we will be using a preset "Close" button.

► To create the "Close" button:

1. Resize the Map control in order to create a button at the bottom of the window.
2. On the "Creation" pane, in the "Usual controls" group, expand "Button" (click the arrow found below .
3. In the list of preset buttons, click "Close": the shape of the button appears under the mouse. Then, click at the bottom of the window to create the button.
4. Resize the created button if necessary.
5. Like for the Map control, you must modify the anchor of this control: the "Width and bottom" anchor must be applied to it (.

## Using a Multiline Zone control

The "Multiline zone" control is often used in the iOS applications.

This control is used to group several controls:

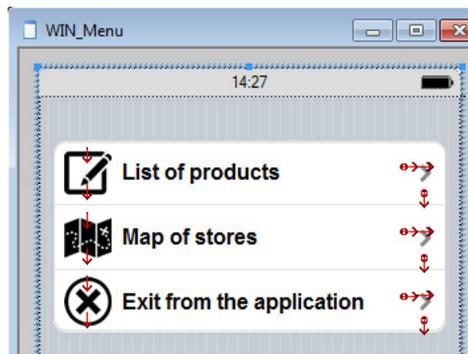
- options on a category,
- group of information about a contact, ...

This control can contain at the same time:

- lines defined in edit (static lines).
- lines defined at run time, by programming (dynamic lines).

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 blank 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. Click "Window" in the wheel that is displayed. Choose "Blank for iPhone" and validate the wizard.
2. Save the window. Specify the title of the window: "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 shape of the control appears under the mouse.
2. Click inside the window to create the control.
3. Display the description window of the control (double-click the control for example).
4. In the description window, specify the name of the control ("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 includes 3 options:

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

## Modifying the Multiline Zone control

► To modify the Multiline Zone control:

1. Display the description window of the control (double-click the control for example).
2. Click the "New line" button. A window is opened: this window contains all the preset line templates.
3. Select the "Simple line with picto" template and validate. Repeat this operation twice. Now, the multiline zone contains:
  - a "blank" line.
  - 3 "simple lines with picto".



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

- Select the blank line (the first line) with the mouse.
- Click the "Delete" button.

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

Each line 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 line in order to represent the action of the menu.

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

- ▶ To modify the image of the first line found in the Multiline Zone control:
  1. Click the first line and select the Image control.
  2. Display the description window of the image (double-click the control).
  3. In the description window:
    - Give a name to the image ("IMG\_ListOfProducts" for example).
    - Click the "Catalog" button in order to choose an image representing the action.
    - In the window of the image catalog, type "List" in the search control and press the "Enter" key.
    - Select an image via a double click.
    - In the window for configuring the generated image:
      - Select a size (80 for example).



Notes

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.

- Modify the lightness of the image (-100% for example to get a black image).
  - Give a name to the image ("ListOfProducts" for example).
  - Validate.
4. Validate the description window.
- ▶ To modify the Static control of the first line found in the Multiline Zone control:
    1. Click the first line and select the Static control.
    2. Display the description window (double-click the control).
    3. In the description window:
      - 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 line: access to the map of stores

- ▶ To modify the image of the second line found in the Multiline Zone control:
  1. Click the second line and select the Image control.
  2. Display the description window of the image (double-click the control).
  3. In the description window:
    - Give a name to the image ("IMG\_MapOfStores" for example).
    - Click the "Catalog" button in order to choose an image representing the action.
    - In the window of the image catalog, type "Map" in the search control and press the "Enter" key.
    - Select an image via a double click.
    - In the window for configuring the generated image:
      - Select a size (80 for example).
      - Modify the lightness of the image (-100% for example to get a black image).
      - Give a name to the image ("MapOfStores" for example).
      - Validate.
  4. Validate the description window.
- ▶ To modify the Static control of the second line found in the Multiline Zone control:
  1. Click the second line and select the Static control.
  2. Display the description window (double-click the control).
  3. In the description window:
    - 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 3<sup>rd</sup> line: exit from the application

- ▶ To modify the image of the third line found in the Multiline Zone control:
  1. Click the third line and select the Image control.
  2. Display the description window of the image (double-click the control).
  3. In the description window:
    - Give a name to the image ("IMG\_Exit" for example).
    - Click the "Catalog" button in order to choose an image representing the action.
    - In the window of the image catalog, type "Close" in the search control and press the "Enter" key.
    - Select an image via a double click.
    - In the window for configuring the generated image:
      - Select a size (80 for example).
      - Modify the lightness of the image (-100% for example to get a black image).
      - Give a name to the image ("Close" for example).
      - Validate.
  4. Validate the description window.

- ▶ To modify the Static control of the third line found in the Multiline Zone control:
  1. Click the third line and select the Static control.
  2. Display the description window (double-click the control).
  3. In the description window:
    - 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

- ▶ We are now going to write the code used to perform each menu action:
  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 included inside it.

  2. In the code editor, enter the following code in the "Selection (click) of a line in..." section:

```
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 the cross at the top right of the code editor).

## Running the test of the application

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 (in iPhone and in iPad) and run a full test of the project.

- ▶ To define the first project window:
  1. Select the "WIN\_Menu" window in the project explorer.
  2. Display the popup menu.
  3. Select "First window for the iPhone platform". A specific icon (with a small 1) is displayed in front of the name of the window, in the project explorer.
  4. Select "First window for the iPad platform". A specific icon (with a small 1) is displayed in front of the name of the window, in the project explorer.

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

- ▶ To run the test of the project:
  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 run its test.

## Generating the application

- ▶ To generate the iOS application:
  1. On the "Project" pane, in the "Generation" group, click "Generate" (you also have the ability to click **iOS** among the quick access buttons).
  2. The wizard for generating an iOS application starts.
  3. The first wizard screen is used to:
    - define the name of the application, the company and the copyright.
    - enter the bundle identifier.

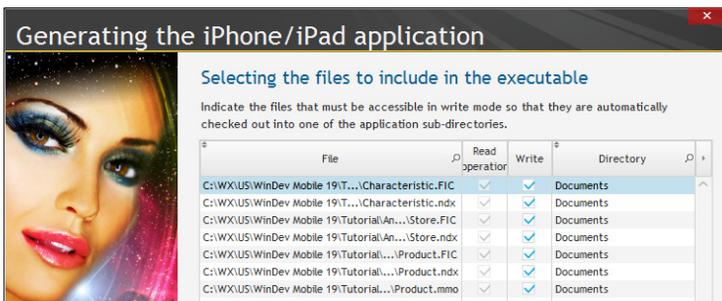


Notes

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 standards of Apple. This identifier can be modified.

4. Go to the next screen. Specify the path of the different icons found in the application. Three icons must be provided:
  - an icon for iPad
  - two icons for iPhone (the iPhone 4 has a different resolution with the Retina screen).
5. Go to the next screen. Specify the path of the different images used when starting the application for iPad and for iPhone (images in PNG format). Default images are automatically created for your application.
6. Go to the next screen. Specify the version number of the generated application.
7. Go to the next screen. This screen is used to include specific files (data files, images, ...). In our example, we are going to include the HFSQL data files.
  - Click the "Add" button.
  - Select the data files (.fic, .ndx and .mmo) found in the EXE directory of the project. The list of files is displayed.
  - Indicate that these data files must be in write mode: check the "Write" box.



8. Go to the next screen. This screen is used to specify:
  - the minimum iOS version required to run the application.
  - 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.
9. Validate the wizard. The generation is performed in the EXE folder of the project directory. The directory containing the source codes 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 the Mac. We won't present these operations here. See the online help for more details.

## LESSON 3.5. DISTRIBUTING THE APPLICATION

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

---

- Available distribution modes



Estimated time: 5 min

## 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 without any restrictions via the sales network of App Store. 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 a network containing up to 100 devices (iPhone or 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 test of the application and/or to deploy the application on a device (iPhone or iPad), you must register toward the iOS Developer Program. This registration is not free of charge. The list of registration programs is available from the following address:

<http://developer.apple.com/programs/which-program/>

Three types of registration are available:

- iOS Developer Program - Individual
- iOS Developer Program - Company
- 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.

## LESSON 3.6. QUESTIONS/ANSWERS

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

---

- Questions/Answers



Estimated time: 10 min

## Questions/Answers

### Question How do I use the messaging of the device?

You can send messages by using the native application for sending emails found on the device. To do so, you must:

- initialize the information of the Email structure.
- use *EmailRunApp*.

### Question How do I send an email directly?

You have the ability to send emails by using the email functions of WLanguage.

Example for sending emails:

```
MyMessage is Email

// Reset the variable to zero
EmailReset(MyMessage)

MyMessage..Sender = EDT_Sender
MyMessage..Subject = EDT_SUBJECT
MyMessage..Message = EDT_BODY
Add(MyMessage..Recipient, EDT_Recipient_Addr)

// Start a SMTP session
MySession is EmailSMTPSession
MySession..ServerAddress = sServer
IF EmailStartSession(MySession) = False THEN RETURN

// Send the message
EmailSendMessage(MySession, MyMessage)

// Close the session
EmailCloseSession(MySession)
```

### Question How do I establish the dialog with an FTP server?

The iOS system allows you to connect to an FTP server in order to send or retrieve files. To do so, you must use the following functions:

- *FTPConnect* to connect to the FTP server
- *FTPDisconnect* to disconnect
- *FTPSend* to send a file from the device to the FTP server
- *FTPGet* to get a file from the server to the device.

Example for sending files onto an FTP server:

```
// Connection to an FTP server
nConnection is int
nConnection = FTPConnect("Address of FTP server")
ResSend = FTPSend(ConnectionNum, "/sdcard/photo.jpg", ...
                  "/Temp")

// Disconnection
FTPDisconnect(nConnection)
```

### Question

### How do I use the GPS of the device and the geo-localization functions?

The iOS system includes functions for GPS management and for geo-localization. According to the device, the GPS functions will use the GPS chip of the device or a triangulation system by using the wireless networks.

To do so, you must use the following functions:

- **GPSInitParameter** to initialize the GPS (choice of localization system...)
- **GPSGetPosition** to retrieve the current position
- **geoDistance** to retrieve the distance between 2 positions
- **geoRunApp** to start the native mapping application of the device
- **geoGetAddress** to retrieve an address or a list of addresses in relation to a location
- etc.

Example for using the GPS functions:

```
// Find a localization provider
// that can give information about the speed
// while consuming an average amount of energy and
// while providing a high level of precision
GPSInitParameter(gpsAuto, gpsPrecisionHigh + ...
                gpsSpeed + gpsEnergyMedium)
MyPosition is geoPosition
MyPosition = GPSGetPosition()
Info(" Latitude: " + MyPosition..Latitude)
Info(" Longitude: " + MyPosition..Longitude)
```

### Caution:

During the first startup of a GPS function, the system asks the user for the authorization to perform a geo-localization. If the user refuses, all the GPS functions used in the rest of the application will fail.

To re-allow the use of GPS for this application, the system configuration of the application must be modified.

**Question**    **How do I display a chart in an iOS application?**

WinDev Mobile includes a Chart control. This control allows you to draw a pie chart, a bar chart, a line chart, ... in an iOS application.

The principle is as follows:

1. Calculate (via a query or procedures in WLanguage) the values to represent in the chart.
2. Create a Chart control in a window (on the "Creation" pane, in the "Graphic controls" group, click "Chart").
3. In the wizard for creating the Chart control, specify the data source of the chart (query, file, by programming...)

Then, the created chart can be modified by programming with the WLanguage functions. All the functions used to handle a Chart control start with the letters "gr":

- grDraw
- grLegend
- grTitle
- grAddData
- etc.

**Question**    **How do I call native Objective C code from my iOS application?**

When you develop an iOS application with WinDev Mobile, the native code is in Objective C. WLanguage is translated into Objective C during the compilation. You have the ability, from the WinDev Mobile environment, to include native procedures in Objective C language. To do so, you must:

- create a set of global procedures
- create a global procedure
- click the gray bar of the procedure header (where the WL word is found).

The header bar becomes light blue. The WL word is transformed into ObjC.

All you have to do is enter or copy native Objective C code.

Caution: Make sure that the code entered or copied is written properly: WinDev Mobile does not check the completion. The errors will occur when compiling the project in Xcode.

Note: You also have the ability to include custom libraries (.h and.a) or system libraries (MediaPlayer.framework for example).





**PART 4**

# Windows Store Apps application

# 10

**Express**



**DEVELOP 10 TIMES FASTER**

**PCSOFT**



# LESSON 4.1. MY FIRST WINDOWS STORE APPS PROJECT

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

---

- Required configuration
- Creating a Windows Store apps project
- My first window
- My first test
- First deployment



Estimated time: 40 min

WinDev Mobile 19 "Express" version:  
The development of applications in "Windows Store Apps" mode  
is not available in this trial version.

## Overview

---

To start working with WinDev Mobile, 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 with WinDev Mobile.

Before creating our first project for Windows Store Apps, a configuration of the development computer is required.

## Configuration required for the Windows Store apps applications

---

To develop a WinDev Mobile application for Windows Store apps, the following elements must be installed on the development computer:

- The Windows 8 SDK of Microsoft. The SDK for Windows 8 (Software Development Kit) is a set of files and applications distributed by Microsoft in order to allow the compilation of applications for the Windows 8 operating system.
- The distributable pack of Microsoft libraries. These libraries are required to generate a Windows Store apps application.
- A developer license for Windows 8.

Note: A project for Windows Store Apps can be developed on any system. However, the development computer must be running Windows 8 in order to compile and create the Windows Store Apps application.

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

## My first project

---

### Creating the project

We are going to create our first project in Windows Store apps mode.



EXAMPLE

Answer

A corrected project is available. To open this project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My Windows Store apps project (Answer)".

- ▶ To create a project:
  1. Start WinDev Mobile 19 (if not already done). Close (if necessary) the current project to display the home window.
  2. In the home window, click the "Create a project" icon then "Windows Store apps application".



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

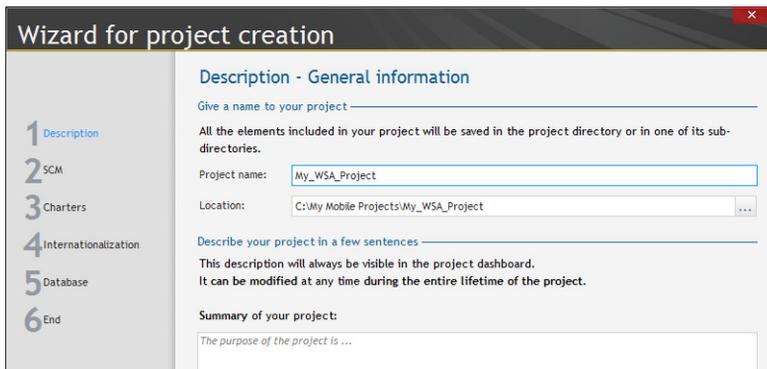


Notes

Tip: To create a project, you can also:

1. Click  among the quick access buttons of the WinDev Mobile menu.
2. Click "Project" in the wheel that is displayed.

3. The first wizard screen is used to enter the name of the project, its location and its description. In our case, this project will be named "My\_WSA\_Project". WinDev Mobile proposes to create this project in the "\\My Mobile projects\\My\_WSA\_Project" directory. You can keep this location or modify it via the [...] button.



4. Go to the next screen via the arrows found at the bottom.

5. The wizard proposes to add documents. Keep the default options and go to the next screen.
6. The following screen is used to configure the directory of Windows 8 SDK.



Note: The path of the SDK is stored for all the forthcoming projects.

7. In the left section of the wizard, click "Charters". This step is used to define the programming charter. Don't modify the suggested options. Go to the next screen via the arrows found at the bottom.
8. This step is used to define the style book. Select "Win 8 Blue".
9. The other wizard steps not being important for our first project, click "End" in the left section of the wizard.
10. Click the validation button at the bottom of the wizard. The project is automatically created. WinDev Mobile displays the different possible actions.
11. Click "WinDev Mobile editor".

## 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, too simple, but we recommend that you 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 Store apps application with WinDev Mobile.

### Creating the window

► To create the window:

1. Click  among the quick access buttons of the WinDev Mobile menu:



2. A window shaped like a wheel is displayed. This window is used to create all the elements that can be associated with a project.
3. Click "Window". The wizard for window creation starts.
4. Select "Blank" in the list of windows displayed on the left. In the list of skin templates found on the right, the "Win 8 Blue" skin template is selected by default. You can choose another skin template proposed in the list.



Note

The skin templates allow you to quickly create outstanding interfaces. A skin template defines the style of the window but also the style of all the controls that will be used in this window. No ugly interface anymore.

5. Validate. The window is automatically created in the editor.
- Save the window by clicking among the quick access buttons. During the first backup, a specific window is displayed. This window proposes to enter:
- the title of the element: enter "Welcome". In our case, this title will be displayed in the title bar of the window.
  - the name of the element is the name of the window. This name will be used in programming. By default, this name includes "WIN\_" that corresponds to the programming charter and "Welcome" that corresponds to the title of the window.



Note

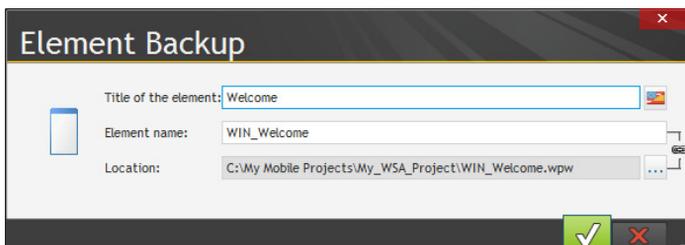
Let's take a look at the window name proposed by WinDev Mobile: this name starts with the letters "WIN\_". This prefix is automatically added because the project uses a programming charter.

The programming charter 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 don't want to use this charter, all you have to do is disable it: on the "Project" pane, in the "Other actions" group, expand "Charter" and uncheck "Use the charter".

- the location that corresponds to the file name created for the window. The window is a file whose extension is "WPW", saved in the project directory.



- Click the green button to validate.

## Displaying a message

You are now going to create a button used to display a message.

- ▶ To create the "Display" button:
  1. On the "Creation" pane, in the "Usual controls" group, click **Ok**. The button appears in creation under the mouse.
  2. Move the mouse toward the position where the control must be created in the window (at the top of the window for example). To drop the control in the window, all you have to do is perform a new left mouse click.
  3. Perform a right mouse click on the control that was just created. The popup menu of the control is displayed. Select "Description" from this popup menu. The description window of the button is displayed.
- ▶ Modify the characteristics of the control by entering the following information:



1. Name of this control: "BTN\_Display".
2. Caption of this control: "Display"

- ▶ Validate the description window of the control (green 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**.



Notes

WLanguage is the programming language supplied with WinDev Mobile. It is a 5th generation language (5GL) that includes highly sophisticated commands.

1. Select the "Display" button with the mouse: all you have to do is click it.
2. Display the popup menu of the control (right mouse click).
3. Select "Code". This option opens the code editor of WinDev Mobile, in which all the WLanguage statements can be entered.
4. Enter the following code in the "Click BTN\_Display" process:

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

**Note about the assisted input:** As soon as the first two characters are typed, WinDev Mobile proposes all the words of the WLanguage vocabulary containing these characters. The assisted development is a very powerful feature. No more mistake when typing 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.

**Notes**

When typing this code in the code editor, you have noticed that different colors are used by the different elements. 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.

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

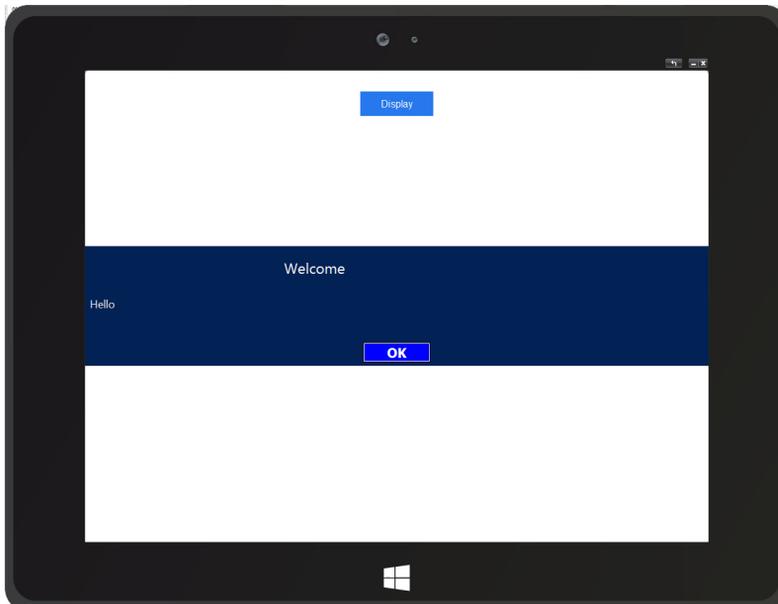
- ▶ Save the modifications by clicking  among the quick access buttons or by pressing [Ctrl]+[S].
- ▶ Close the code editor (cross at the top right of the code editor). The window re-appears.

## First test

---

For a Windows Store apps application, WinDev Mobile allows you to run the test of the application on the development computer via the simulation mode. This test performs a simulation of a Windows Store apps device on the development computer.

- ▶ Let's now run the test of the window 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 a tablet.
  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, a SINGLE CLICK allows you to run the test of the 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 editor of WinDev Mobile is redisplayed.

## First local deployment

---

### Principle

A Windows Store Apps application can be directly run on the development computer if the system used is Windows 8. To run the application in stand-alone mode on the development computer, you must:

- Generate the Windows Store Apps application in WinDev Mobile.
- Then, the program will be installed on the computer. It will run in stand-alone mode.

Let's take a look at these different steps.

## Implementation

► To generate the Windows Store Apps 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 the "WIN\_Welcome" window and validate (green button at the bottom of the screen).
3. The wizard for generating a Windows Store Apps application starts.
4. The first wizard screen is used to select the generation mode of the application. You can:
  - Generate and deploy the application on a local computer: we will be using this option.
  - Generate the application for Windows Store: this option is used to generate the application and to distribute it via the Windows store. See the online help for more details.



5. Go to the next screen.
6. Specify the general information about the application: the name of the application, the company and the copyright.
7. Go to the next screen. This screen is used to define the version number of the application.
8. Go to the next screen. Specify the path of the different images used when starting the application: image of the tile, splash screen, ... Default images are automatically created for your application. Specify the characteristics of the tile associated with the application.
9. Go to the next screen. Specify the custom error message displayed by the application.
10. Go to the next screen. This screen is used to include specific files (data files, images, ...). This possibility will not be used in our example. Keep the default options.
11. Go to the next screen. This screen is used to specify the features used by the application. In this example, don't check anything and go to the next screen.
12. The Windows Store Apps applications must be signed to operate. The wizard proposes to generate a certificate or to use an existing certificate. In this last case, all you have to do is select the requested certificate.



To generate a new certificate:

- Specify the name of the certificate to generate.
  - The next wizard plane is used to create the certificate.
  - Click the "Create the certificate" button and specify the different passwords.
  - Validate the setup of the certificate on the current computer.
  - Enter a password again.
  - The Windows explorer displays the created certificate.
  - In the explorer, display the popup menu of the file (right mouse click) and select "Install the certificate". The wizard for importing the certificate starts.
  - In the wizard, select the storage location: "Local computer". Click "Next".
  - Select "Place all certificates in the following store".
  - Click the "Browse" button and select the "Trust root certification authorities" store.
  - Click the "Next" button until the end of the wizard. A message indicates that the import was successfully performed.
  - In the wizard, check "The certificate was successfully installed" and go to the next screen.
- 13.** Validate the generation of the Windows 8 application. The application is generated and the corresponding tile is created on the home screen of Windows 8. The application can be started immediately.

Note: The generated application can be installed on a Windows 8 tablet. See the online help for more details.

## LESSON 4.2. INTERFACE (GUI)

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

---

- Orientation of the window
- Management of touchscreen



Estimated time: 20 min

WinDev Mobile 19 "Express" version:  
The development of applications in "Windows Store Apps" mode  
is not available in this trial version.

## Overview

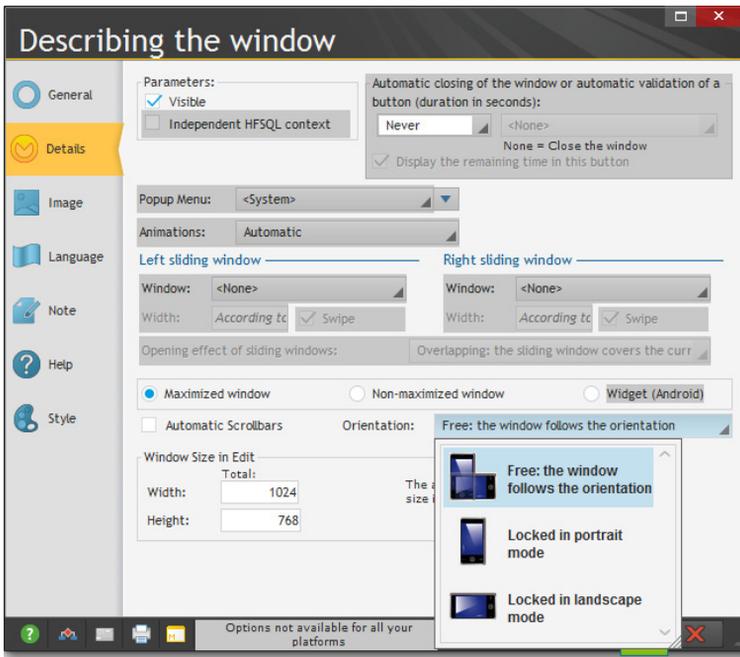
The Windows Store Apps applications are available both on a PC (in tile mode) and on tablets. WinDev Mobile allows you to easily create interfaces that adapt to the device used.

## Orientation of the window

On a tablet, a window can have one of the following orientations:

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

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



No specific operation must be performed in the two first cases.

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

## Practical example

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



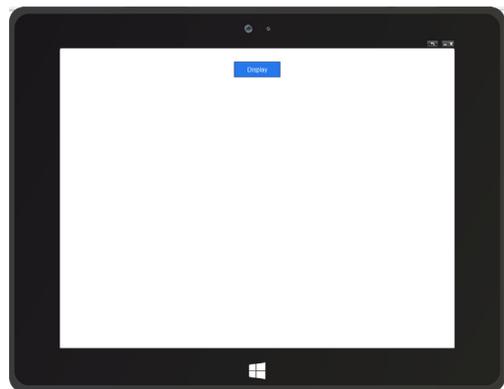
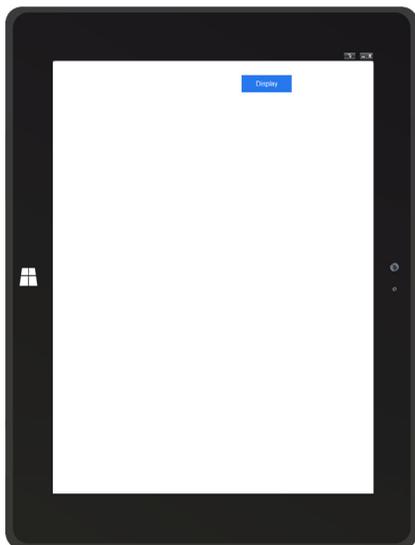
EXAMPLE

Answer

A corrected project is available. To open this project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My Windows Store Apps project (Answer)".

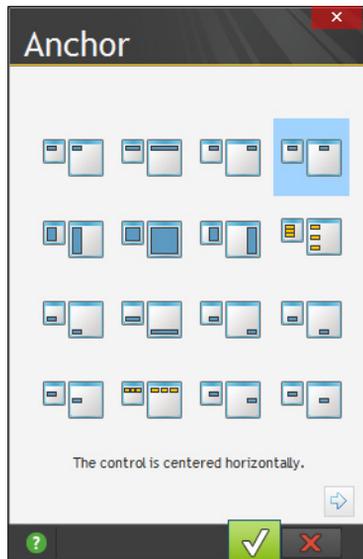
In our example, the test of our project was run in portrait mode in the simulator. We are now going to run its test in landscape mode in the simulator.

- ▶ Run the test of the project ( among the quick access buttons).
  1. The window is displayed in portrait mode.
  2. In the simulator, click the arrow in the shell ().
  3. The orientation of the window changes on the screen.
  4. In our example, the location of the button does not change: it does not adapt to the orientation of the screen.



- ▶ We are now going to modify our window in order for the "Display" button to be centered in the window and to remain centered regardless of the device orientation.
- ▶ Stop the test and go back to the editor.
- ▶ To center the button in the window:
  1. Select the button (click the button).
  2. On the "Alignment" pane, click "Center in the parent (horz)".

- In order for the button to remain centered in the window, we are going to use the control anchoring:
1. Select the button (click the button).
  2. Display the popup menu (right mouse click).
  3. Select "Anchor": the window for defining the anchors is displayed:



4. Select "Horizontally centered" and validate (green button).
- Run the test of the project (🏃 among the quick access buttons):
- The button is centered in portrait mode.
  - Change the orientation of the simulator.
  - The button remains centered in landscape mode.

## LESSON 4.3. DATABASES

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

---

- Available databases
- Synchronization



Estimated time: 10 min

WinDev Mobile 19 "Express" version:  
The development of applications in "Windows Store Apps" mode  
is not available in this trial version.

## The available databases

---

### Overview

In the Windows Store Apps applications, only the HFSQL database is accessible in native mode. Both the Classic mode and the Client/Server mode are available.

### 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 memory of the device. The maximum storage size depends on the amount of memory on the device.

In the "Advanced programming" lesson, we will develop an application that uses a HFSQL Classic database.

#### 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 the WiFi or Internet network and on the volume of requested data.

The access to the data will be performed by the Hxxx functions of WLanguage and/or by SQL queries.

## The synchronization

---

The synchronization mechanism is used to "synchronize" the data stored on a mobile device with the data stored on a server. The synchronization uses the mechanism of "universal replication".

This technique is available in WinDev, WebDev and WinDev Mobile.

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

## **Accessing data via a Webservice**

---

Another method can also be used to access the data on a server from a mobile device: the call to a Webservice.

In this case, the mobile device must be equipped with a Wi-Fi or 3G connection to connect to the Webservice.

The mobile application does not directly access the database. The application calls the functions of the Webservice. These functions return the data.

It is the Webservice that accesses the database.

This technique is used to have a business layer (the Webservice) common to several types of applications and interfaces (WinDev, WebDev or WinDev Mobile) and different types of operating systems (Windows, Android, iOS, ...).

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

## LESSON 4.4. ADVANCED PROGRAMMING

### This lesson will teach you the following concepts...

- Creating a management application
- Specific controls: Looper
- Handling the database
- Features specific to the device used (Photo, ...)



Estimated time: 1h 30min

WinDev Mobile 19 "Express" version:  
The development of applications in "Windows Store Apps" mode  
is not available in this trial version.

## Overview

In this lesson, we are going to develop a Windows Store Apps application that uses a HFSQL Classic database.

This application will allow us to present some specific features of the Windows Store Apps programming.

### Opening the project

- ▶ Start WinDev Mobile 19 (if not already done). Close (if necessary) the current project to display the home window.
- ▶ Open the "WSA Managing Products" project.  
To do so, in the home window, click "Tutorial" and select the "Windows Store apps Managing Products (Exercise)" project.  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Windows Store apps Managing Products (Exercise)".



EXAMPLE

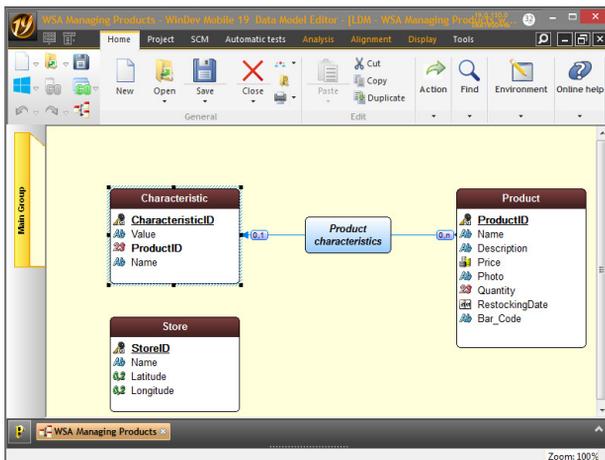
Answer

A corrected project is available. This project contains the different windows created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Windows Store apps Managing Products (Answer)".

### Describing the project

Let's take a look at our source 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 the WinDev Mobile menu.
  2. The data model editor is displayed.



3. This analysis includes 3 data files:

- A "Product" file, that contains the description of the product: name, price, quantity, ...
- A "Characteristic" file, that contains the different characteristics of the product. For example, if the product is a tee-shirt, its characteristics will correspond to the size, the color, ... Therefore, the "Characteristic" file is linked to the "Product" file.
- A "Store" file, that contains the GPS coordinates of each store.

4. Close the data model editor (click the cross at the top right of the editor).

We are now going to develop our application.



Caution!

In this section, the different tests will be run in simulation mode in WinDev Mobile. Indeed, deploying the Windows Store Apps application is quite long.

## 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.

### Creating the window

► To create a new window:

1. Click  among the quick access buttons. Click "Window" in the wheel that is displayed. Choose "Blank" and validate the wizard.
2. Save the window by clicking  among the quick access buttons.
3. In the window that is displayed, enter the title of the window: "List of products". The name and location of the window are automatically filled. Keep the default values and validate (green button at the bottom of the screen).
4. The window is added to the project.

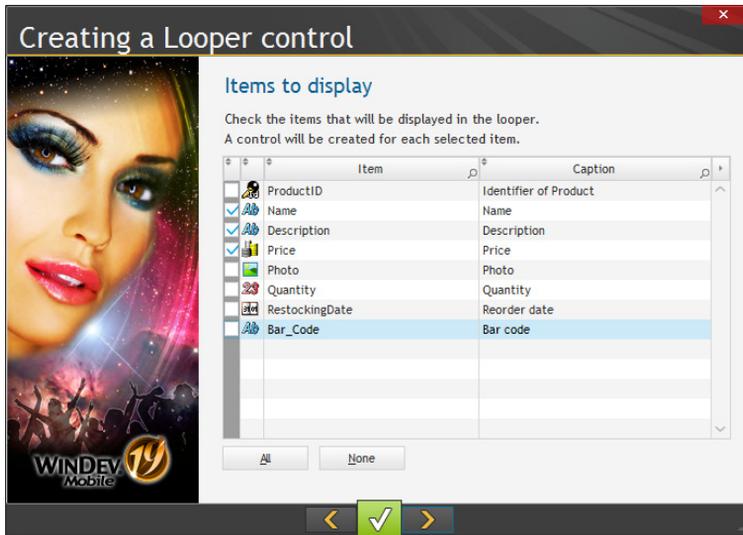
### Creating the Looper control

A "Looper" control will be used to display the list of products. The main information about the products will be displayed in this control.

► To create the "Looper" control:

1. On the "Creation" pane, in the "Data" group, expand "Looper" and select "Looper (vertical)". The control appears under the mouse.
2. Click the position where the control must be created in the window (at the top for example). The wizard for creating a Looper control starts.
3. In the wizard, select "Display the data coming from a file or from a query". Go to the next screen.
4. Select the "Product" file. Go to the next screen.

5. Select the items to display:



6. Keep the "Name", "Description" and "Price" items (a checkmark must be found in front of these items). Go to the next screen.
7. Keep the proposed sort item (ProductID). The products will be sorted in the Looper control according to this item. Go to the next screen.
8. In the "Additional parameters" screen, keep the default options. Go to the next screen.
9. Keep the default name ("LOOP\_Product") and validate.
10. Increase the Looper control if necessary.



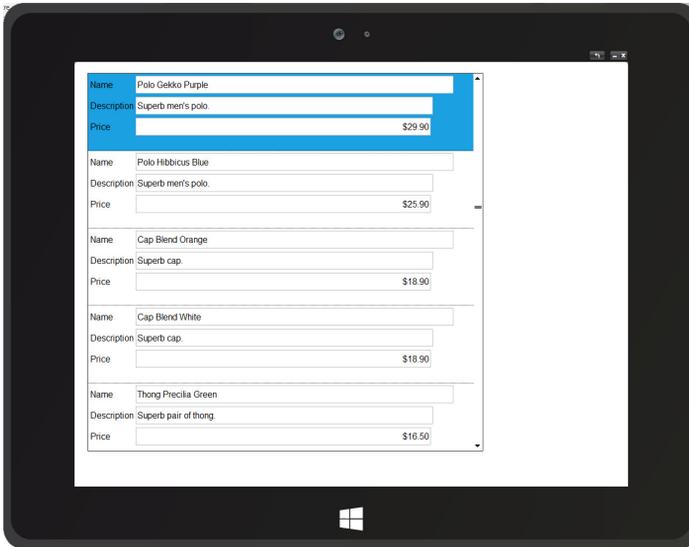
Notes

The data automatically appears in the window displayed in the editor. This concept is called "Live Data": you see the data found in 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 view the result. Click  among the quick access buttons (or press [F9]).



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

## Improving the window

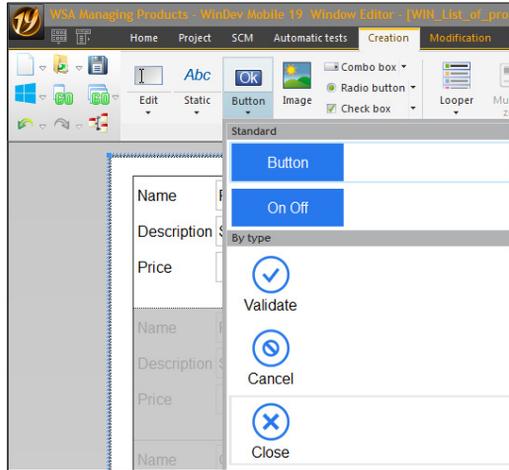
The importance of anchors was presented in the previous lesson. We are now going to apply an anchor to our Looper control so that it automatically resized according to the orientation of the tablet.

- ▶ To anchor the Looper control:
  1. Select the Looper control.
  2. Display the popup menu of the control (right mouse click) and select "Anchor".
  3. In the window that is displayed, select the option .
  4. Validate. The red arrows indicating the anchoring options appear in the window.

## Creating a button to close the window

We are now going to add a "Close" button: this button will allow the user to close the window (as well as the application in our case).

- ▶ To create the "Close" button:
  1. Resize the Looper control in order to create a button at the bottom of the window.
  2. On the "Creation" pane, in the "Usual controls" group, expand "Button" (click the arrow found below .
  3. The list of preset buttons is displayed.



Notes

For some types of controls, WinDev Mobile proposes a list of preset controls. These controls are advanced controls, configured to perform a specific action. For example, the "Close" button proposed by the list of preset controls contains the code required to its execution.

4. Click "Close": the shape of the button appears under the mouse. Then, click at the bottom left of the window to create the button.
  5. Resize the created button if necessary.
- Run the test of the window ( among the quick access buttons). Click the "Close" button. The window is closed and the editor is redisplayed.

## Creating the form window

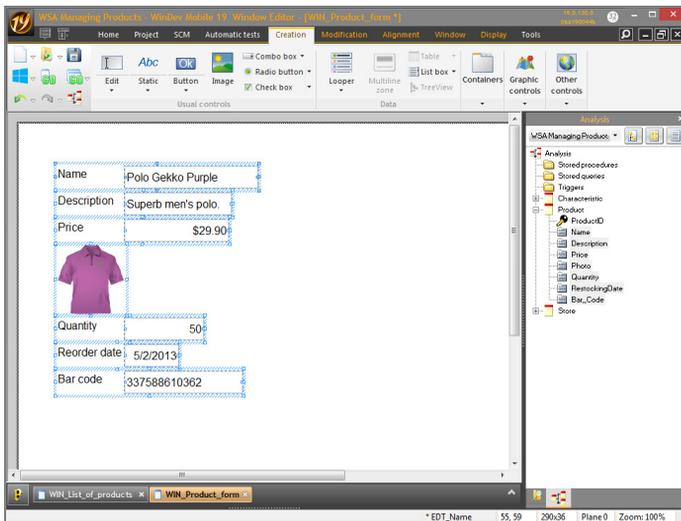
We are now going to create a new window used to display the product form. Then, this window will be started from the list of products to display the details of the selected product.

### Creating the window

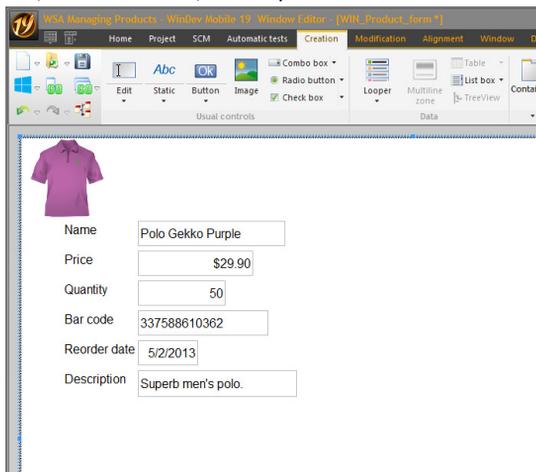
- To create the form window:
1. Create a new blank window.
    - Click among the quick access buttons.
    - Click "Window" in the wheel that is displayed. Choose "Blank" and validate the wizard.
  2. Save the window. Specify the title of the window: "Product form". Its name is automatically proposed: "WIN\_Product\_form". Validate.

### Creating the controls

- ▶ To create an edit control:
  1. Display the "Analysis" pane if necessary (on the "Home" pane, in the "Environment" group, expand "Panes" and select "Analysis"). The different data files described in the "WSA Managing Products" analysis appear in the pane.
  2. With the mouse, select the items of the "Product" 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 "Description" control so that it is visible in the window.
5. Reorganize the controls in the window. Respect the following order: "Photo", "Name", "Price", "Quantity", "BarCode", "Reorder Date", "Description".



6. We are going to view the navigation order in the window: press the [F5] key. The numbers that are displayed represent the navigation order in the window. Press [F5] again to make the numbers disappear. To correct the navigation order, on the "Window" pane, in the "Order" group, expand "Navigation" and select "Define automatically".

7. Save the window.

► Let's study the anchoring applied to the controls of this window. The "Name" control and the "Description" control will be anchored.

1. For the "Name" control:

- Select the "Name" control and perform a right mouse click.
- Select "Anchor" from the popup menu.
- The control will stretch to the right: select  and validate.

2. For the "Description" control:

- Select the "Description" control and perform a right mouse click.
- Select "Anchor" from the popup menu.
- The control will stretch to the right and to the bottom: select  and validate.

► Run the test of the window ( among the quick access buttons). The window is displayed with empty controls.

► To display the product data:

1. Display the processes associated with the window:

- Perform a right mouse click in the area beside the window
- Select "Code" from the popup menu.
- The code editor appears.

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

```
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.

4. Save the window.

## Displaying the form from the list of products

Let's see how to display the form of the 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 found in the button bar:



2. Right-click the Looper control and select "Code" from the popup menu.

3. In the code window that is displayed, enter the following code in the "Selecting a row of..." process:

```
OpenMobileWindow(WIN_Product_form)
```



Notes

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 the existing windows found in the project. All you have to do is select the window with the keyboard or with the mouse.

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

4. Save the modifications by clicking  among the quick access buttons.
  5. Close the code window (click the cross at the top right of the 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.
    - The detailed window of the product is displayed.
  - ▶ Close the simulator.

## Managing the creation and the modification of a product

---

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

### Modifying the product form

We are going to add 2 buttons into the "WIN\_Product\_form" window:

- a "Validate" button to manage the validation of modifications.
  - a "Cancel" button to manage the return to the list of products.
- ▶ Display (if necessary) the "WIN\_Product\_form" window in the editor: click the corresponding button in the button bar.
  - ▶ To create the "Cancel" button:
    1. On the "Creation" pane, in the "Usual controls" group, expand "Button" (click the arrow found below .
    2. The list of preset buttons is displayed.
    3. Click "Cancel": the shape of the button appears under the mouse. Then, click at the bottom right of the window to create the button.
  - ▶ We are now going to add the "Validate" button according to the same principle.
    1. On the "Creation" pane, in the "Usual controls" group, expand "Button" (click the arrow found below .
    2. The list of preset buttons is displayed.
    3. Click "Validate": the shape of the button appears under the mouse. Then, click at the bottom of the window to create the button.

- ▶ We are now going to enter the code of the "Validate" button.
  1. Right-click the button and select "Code" from the popup menu.
  2. In the "Click" process, enter the following code:

```
ScreenToFile ()
HModify (Product)
Close ()
```

Let's take a look at this code:

- **ScreenToFile** is used to initialize the items with the values of the linked controls, for the current record.
  - **HModify** is used to update the file data for the current record.
3. Save the modifications by clicking  among the quick access buttons.
  4. Close the code window (click the cross at the top right of the 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 "Closing a child window" process of the "WIN\_List\_of\_products" window.
    1. Display the "WIN\_List\_of\_products" window in the window editor.
    2. Right-click the window background and select "Code" from the popup menu. Enter the following code in the "Closing a child window" process:

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

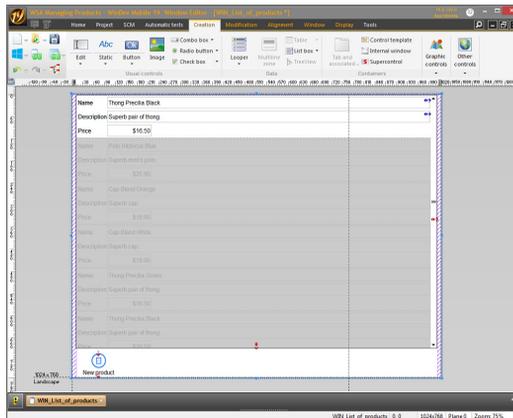
Let's take a look at this code:

- The "Closing a child window" process 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 the "WIN\_List\_of\_products" window. Via the **taCurrentSelection** constant, the update is performed from the current record in the data file.
3. Save the modifications by clicking  among the quick access buttons.
  4. Close the code window (click the cross at the top right of the code editor).
- ▶ Run the test of the "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 whose price is 25.90 Dollars.
    - The detailed window of the product is displayed. Modify the price of 25.90 Dollars and enter 19.90 Dollars and click the "Validate" button.
    - When going back to the list of products, you will notice that the price was updated for this article.
  - ▶ Close the simulator. The editor of WinDev Mobile 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 the "Product form" window to manage the addition into the Product data file.
- To add a button into the list of products:
1. Display the "WIN\_List\_of\_products" window in the editor.
  2. On the "Creation" pane, in the "Usual controls" group, expand "Button" (click the arrow found below )
  3. The list of preset buttons is displayed.
  4. Click "New": the shape of the button appears under the mouse. Then, click at the bottom of the window to create the button (on the left of the "Close" button).
  5. Select the control and press the "Enter" key on the keyboard. The caption of the button switches to edit. Type "New product" and press the "Enter" key on the keyboard.



6. The code of this option is used to open the "Product form" window and to reset its controls. To enter this code:
  - Right-click the control.
  - Select "Code" from the popup menu.
  - In the "Click" process, enter the following code:

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

**HReset** initializes the variables of the items found in the Product 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 the cross at the top right of the code editor).

► We are now going to modify the window of the product form to manage the addition of a new record.

1. Display the "WIN\_Product\_form" window in the editor.
2. We are going to modify the code of the validation button:
  - Right-click the button and select "Code" from the popup menu.
  - In the "Click" process, replace the existing code by the following code:

```
ScreenToFile ()
IF Product..NewRecord THEN
  HAdd(Product)
ELSE
  HModify(Product)
END
Close ()
```

Let's take a look at this code:

- **..NewRecord** is used to find out whether the current record must be created.
  - If **HReset** was called beforehand, the property returns True (case of the 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**.
3. Save the modifications by clicking  among the quick access buttons.
  4. Close the code window (click the cross at the top right of the 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.
  - Enter a new product.
  - Validate. The new product appears in the list of products.
  - Close the simulator.

## Using the camera and displaying an image

---

### Overview

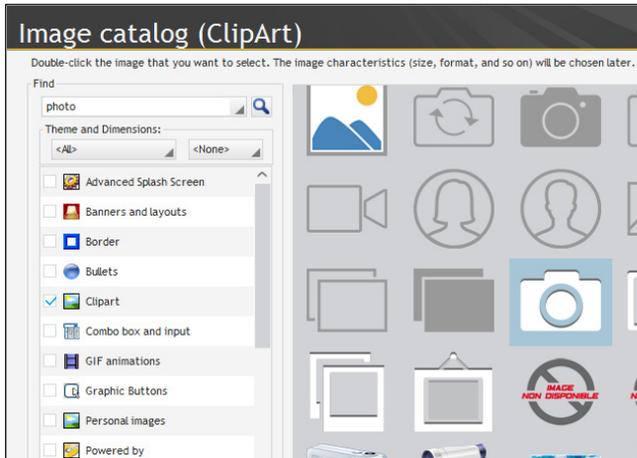
We are going to manage the photo of the product. We will be using the camera of the device.

We are going to create a button to start the camera. The photo will be retrieved in the format of an image in memory and displayed in the Image control of the product.

### Creating the button for taking photos

- To create the button for taking photos:
1. Display the "WIN\_Product\_form" window in the editor.
  2. Add a button into the window:
    - On the "Creation" pane, in the "Usual controls" group, click : the shape of the button appears under the mouse.
    - Then, click at the top right of the product image to create the button.

- ▶ We are going to modify the button in order to associate it with an image representing a camera:
  1. Select the button and display its popup menu (right mouse click).
  2. Select "Description". The description window of the control is displayed.
  3. In the "General" tab, position on the "Image" edit zone and click the "Catalog" button.
  4. The image catalog of WinDev Mobile is opened. This catalog contains hundreds of images in different fields, formats and sizes.
  5. In the "Search" area (top left), type "photo" and validate. Several images are displayed:



6. Double-click the first image to select it.
7. In the next screen, you have the ability to choose the size of the image, the color, the orientation, its format and its name.
8. Keep all the default options and specify the name of the image ("Camera").
9. Validate the window.
10. The path of the image is displayed in the button description.
11. Give a name to the button: "BTN\_Camera".
12. Clear the caption of the button.
13. Validate the description window.
14. In the editor, reduce the size of the button.

## Taking photos

We are going to enter the code used to take a photo and to display it in the image control of the product form.

► To manage the taking of photos:

1. Select "Code" from the popup menu of the button (right mouse click).
2. Enter the following code in the "Click" process:

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

In this code **VideoStartApp** is used to start the native camera application of the device in order to save a video or to take a photo.

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

## Running the test of the application

The last step consists in specifying that the "WIN\_List\_of\_products" window is the first application window. To do so, we are going to define the first project window and run a full test of the project.

► To define the first project window:

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

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

► To run the test of the project:

1. Click  among the quick access buttons.
2. Your project starts with the window used to list the products.

That's it, our application is created, we must now compile it and install it on your computer in order to run its test.

## Generating the application

Before generating the application, make sure that your development computer is configured for generating Windows Store Apps applications. See “Configuration required for the Windows Store apps applications”, page 146 for more details.

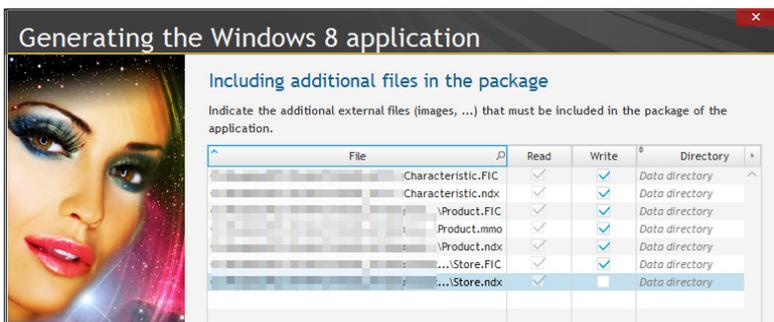
- ▶ To generate the Windows Store Apps application:
  1. On the "Project" pane, in the "Generation" group, click "Generate" (you also have the ability to click  among the quick access buttons).



Notes

A warning message regarding the certificate used for the generation may be displayed. This message informs you that the certificate used to generate the application is not found on the current computer. A certificate must be re-created when generating the application.

2. The wizard for generating a Windows Store Apps application starts.
3. The first wizard screen is used to select the generation mode of the application. Select "Generate and deploy the application on a local computer". Go to the next screen.
4. Specify the general information about the application: the name of the application, the company and the copyright.
5. Go to the next screen while keeping the default options until you reach the screen called "Including additional files in the package".
6. This screen is used to include specific files (data files, images, ...). In our example, we are going to add the data files used by the application:
  - Click the "Add" button.
  - Select the "Product.xxx", "Store.xxx" and "Characteristic.xxx" files found in the EXE sub-directory of the project.
  - To be able to enter information in these files, check "Write" for all these files.



7. Go to the next screen. This screen is used to specify the features used by the application. WinDev Mobile automatically selects the features used in the application (Microphone and Webcam). Keep these options and go to the next screen.

8. The Windows Store Apps applications must be signed to operate. The wizard proposes to generate a certificate or to use an existing certificate. If a certificate was created when generating the "My\_WSA\_Project" example or if you already own a certificate, select this certificate. To generate a new certificate, see "First local deployment", page 152.

9. Validate the generation of the Windows 8 application. The application is generated and the corresponding tile is created on the home screen of Windows 8. The application can be started immediately.

## LESSON 4.5. DISTRIBUTING THE APPLICATION

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

---

- Available distribution modes



Estimated time: 20 min

WinDev Mobile 19 "Express" version:  
The development of applications in "Windows Store Apps" mode  
is not available in this trial version.

## Overview

---

WinDev Mobile allows you to develop Windows Store Apps applications.

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

## Deployment

---

An application in tile mode can be deployed:

- On the local computer directly. The procedure was presented in the previous chapters.
- On a tablet directly. The different steps are presented in the online help (keywords: "Install, An application on a tablet"). If you own a tablet, we advise you to perform this setup.
- Via Windows Store. This method is the standard distribution mode of an application in tile mode. The wizard for generating a Windows Store Apps application allows you to generate the application for a deployment via Windows store.

The deployment via Windows Store requires several steps:

- **The local certification while respecting the certification criteria:** A Windows Store Apps application must be validated by Microsoft before it can be deployed on Windows Store. Microsoft proposes a tool for automatic application check, allowing you to validate your applications on the development computer. Your application must pass this local certification before it is published on the store.
- **The creation of a Microsoft developer account:** A Microsoft developer account is required to submit an application.
- **The submission for deployment on the store:** An application can be dropped on the store from the dashboard of your developer account. A wizard helps you enter the necessary information. To finalize the deployment, the application undergoes several manual tests used to get the certification of the application. The report status of certification will be emailed to you. If no problem occurs, the application becomes available on the store.

See the online help for more details about the deployment procedures.





**PART 5**

**Windows Mobile  
application**

**10**

**Express**



**DEVELOP 10 TIMES FASTER**



**PCSOFT**



# LESSON 5.1. 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: 1h

## Overview

To start working with WinDev Mobile, 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 with WinDev Mobile.

## 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 characteristics of the device will be automatically detected and proposed when creating the Windows Mobile project.



Answer

A corrected project is available. To open this project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "My Pocket project (Answer)".

► To create a project:

1. Start WinDev Mobile 19 (if not already done). Close (if necessary) the current project to display the home window.
2. In the home window, click the "Create a project" icon then "Windows Mobile application".



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



Notes

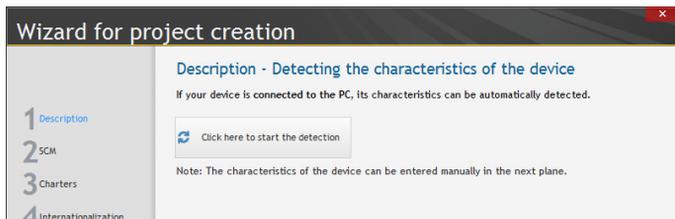
Tip: To create a project, you can also:

1. Click  among the quick access buttons of the WinDev Mobile menu.
2. Click "Project" in the wheel that is displayed.

3. The first wizard screen is used to enter the name of the 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.

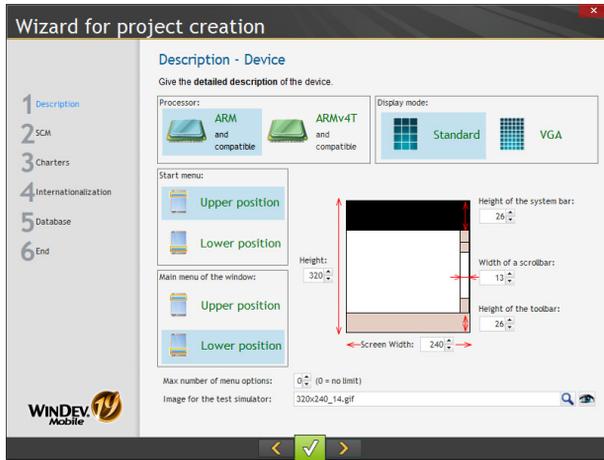


4. Go to the next screen via the arrows found at the bottom.
5. The wizard proposes to add documents. Keep the default options and go to the next screen.
6. The next screen is used to detect the parameters of the device connected to the PC.



- If your device is connected, click "Click here to start the detection".
  - If no device is connected, go to the next screen to manually define the characteristics of the device used.
7. If you have no device, the next screen allows you to choose the device that will be used to develop your application.
- Caution:** this screen is important because the size of your windows will depend on this choice. For our example, select "Windows Mobile 5/6". Go to the next screen.

8. The description window of your device is displayed.



Go to the next screen.

9. In the left section of the wizard, click "Charters". This step is used to define the programming charter. Don't modify the suggested options. Go to the next screen via the arrows found at the bottom.

10. This step is used to define the style book. Select "Elegant".

11. The other wizard steps not being important for our first project, click "End" in the left section of the wizard.

12. Click the validation button at the bottom of the wizard. The project is automatically created. WinDev Mobile displays the different possible actions.

13. Click "WinDev Mobile editor".

## 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, too simple, but we recommend that you 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. Click  among the quick access buttons of the WinDev Mobile menu:



2. A window shaped like a wheel is displayed. This window is used to create all the elements that can be associated with a project.

3. Click "Window". The wizard for window creation starts.

4. Select "Blank" in the list of windows displayed on the left. In the list of skin templates found on the right, the "Elegant" skin template is selected by default. You can choose another skin template proposed in the list.



Note

The skin templates allow you to quickly create outstanding interfaces. A skin template defines the style of the window but also the style of all the controls that will be used in this window. No ugly interface anymore.

5. Validate. The window is automatically created in the editor.

- Save the window by clicking  among the quick access buttons. During the first backup, a specific window is displayed. This window proposes to enter:
- the title of the element: enter "Welcome". In our case, this title will be displayed in the title bar of the window.
  - the name of the element is the name of the window. This name will be used in programming. By default, this name includes "WIN\_" that corresponds to the programming charter and "Welcome" that corresponds to the title of the window.



Note

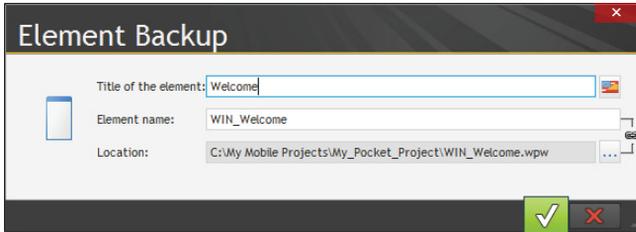
Let's take a look at the window name proposed by WinDev Mobile: this name starts with the letters "WIN\_". This prefix is automatically added because the project uses a programming charter.

The programming charter 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 don't want to use this charter, all you have to do is disable it: on the "Project" pane, in the "Other actions" group, expand "Charter" and uncheck "Use the charter".

- the location that corresponds to the file name created for the window. The window is a file whose extension is "WPW", saved in the project directory.



- ▶ Click the green button to validate.

### Displaying a message

You are now going to create a button used to display a message.

- ▶ To create the "Display" button:
  1. On the "Creation" pane, in the "Usual controls" group, click **OK**. The button appears in creation under the mouse.
  2. Move the mouse toward the position where the control must be created in the window (at the top of the window for example). To drop the control in the window, all you have to do is perform a new left mouse click.
  3. Perform a right mouse click on the control that was just created. The popup menu of the control is displayed. Select "Description" from this popup menu. The description window of the button is displayed.
- ▶ Modify the characteristics of the control by entering the following information:



1. Name of this control: "BTN\_Display".
  2. Caption of this control: "Display"
- ▶ Validate the description window of the control (green 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**.



## Notes

WLanguage is the programming language supplied with WinDev Mobile. It is a 5th generation language (5GL) that includes highly sophisticated commands.

1. Select the "Display" button with the mouse: all you have to do is click it.
2. Display the popup menu of the control (right mouse click).
3. Select "Code". This option opens the code editor of WinDev Mobile, in which all the WLanguage statements can be entered.
4. Enter the following code in the "Click BTN\_Display" process:

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

**Note about the assisted input:** As soon as the first two characters are typed, WinDev Mobile proposes all the words of the WLanguage vocabulary containing these characters. The assisted development is a very powerful feature. No more mistake when typing 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.



## Notes

When typing this code in the code editor, you have noticed that different colors are used by the different elements. 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.

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

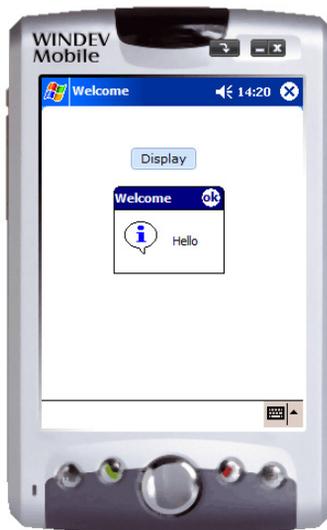
- To save the modifications, click  among the quick access buttons (on the left of ribbon) or by pressing [Ctrl]+[S].
- Close the code window (cross at the top right of the code editor). The window re-appears.

## First test

---

For a Windows Mobile application, WinDev Mobile allows you to run the test of the application on the development computer via the simulation mode. This test simulates a Windows Mobile device on the development computer. This test is useful when the developer is not equipped with a Windows Mobile device. However, this test does not allow you to use the hardware components of the device (SMS, ...).

- ▶ Let's now run the test of the window 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 the 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 editor of WinDev Mobile is redisplayed.

## 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 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 (green button at the bottom of the screen).
3. The wizard for creating a mobile executable starts.
4. The first wizard screen indicates whether automatic tests have been created on the project as well as their status. In our case, no test was run. Go to the next screen via the arrow buttons.
5. The next screen is used to define the name and icon of the application.



The icon of the executable can be chosen in the image catalog of WinDev Mobile ("Catalog" button).



#### Notes

As soon as an image can be used (in a control, in a window, in a report, ...), the "Catalog" button is displayed in the description window of the control. This button allows you to select an image among the images supplied in the image catalog of WinDev, WebDev and WinDev Mobile.

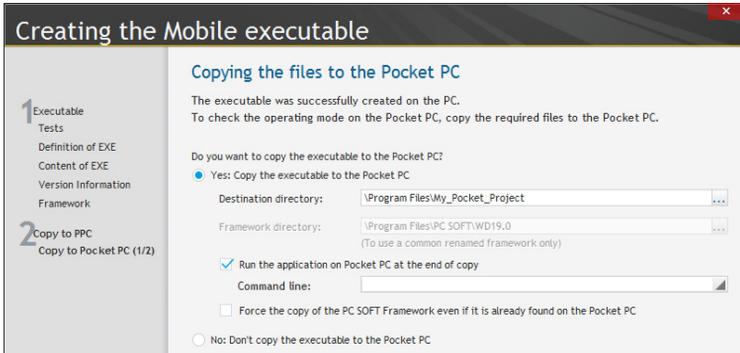
To perform a search in the image catalog:

- 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.

6. The other screens are not required by our application. Click the "2- Copy onto PPC" link found on the left of the wizard.

7. This screen is used to define the options for copying files onto the mobile:



8. The options depend on your configuration:

- If a mobile is connected, select "Yes: Copy the executable onto the Pocket PC" as well as "Start the application on the 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 connected, select "No: Don't copy the executable onto the Pocket PC". In this case, the application can be deployed on the mobile devices via a setup procedure.

9. Validate.

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

## LESSON 5.2. DATABASES

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

---

- Available databases
- Synchronization



Estimated time: 30 min

## Format of the databases

---

A Windows Mobile application can handle the 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 device (Pocket PC).
- **AS/400**, AS/400 database that can be used on mobile device (Pocket PC).
- ...

### HFSQL database

#### HFSQL Classic

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

In this case, the application is stand-alone. No Wi-Fi or 3G connection is required.

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

In the "Advanced programming" lesson, we will develop an application that uses a HFSQL Classic database.

#### 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 the WiFi or Internet network and on the volume of requested data.

The access to the data will be performed by the Hxxx functions of WLanguage and/or by SQL queries.



EXAMPLE

Example

The "Pocket Notes", "Pocket Attendance" and "Pocket Telephony" examples (supplied with WinDev Mobile) handle HFSQL data files.

These examples are accessible from the "Wizards, Examples and Components" pane.

GO

Test

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 Access databases (".MDB" files) previously exported from a PC.

Note: When an Access database (".MDB" file) is copied onto a mobile device (via the file explorer), this database is automatically transformed into a CEDB database (".CDB" file).

A CEDB database can be handled:

- from a WinDev Mobile application.
- from a standard WinDev application.

These operations are performed via the **cdbXXX functions** of WLanguage.



Notes

**Caution:** From Windows Mobile 5, the access to a standard database (tasks, contacts, appointments, ...) is no longer possible from a Windows application (and therefore from a standard WinDev application).

The CDB databases can be accessed from the PC, from the simulator, from the mobile device.

The standard databases can be accessed from the mobile device only.



Notes

**Caution:** The structure of the CEDB databases is not intended to process a large amount of data. Therefore, we recommend that you use HFSQL databases. Furthermore, HFSQL allows you to benefit from all the features available in WinDev Mobile (RAD, file link, ...).



Test

During the test (in simulation mode) of a WinDev Mobile application that handles a CEDB database, the database used is found on the **Pocket PC**.

## AS/400

This database format is accessible via a Native Access by the Windows Mobile applications. To use this Native Access, an additional 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:

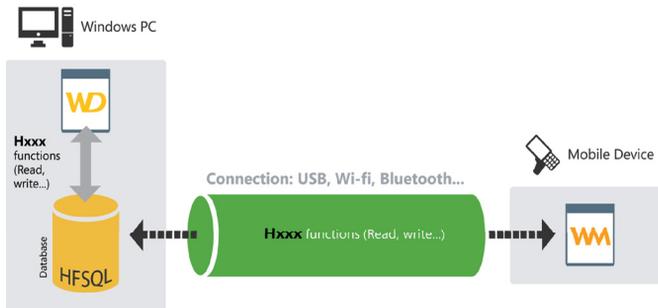
- a standard WinDev application used to handle the entire database.
- a WinDev Mobile application used to handle the entire database or part of this database.

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, ...

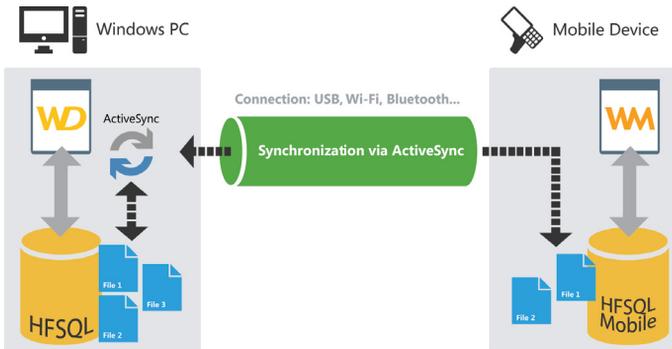
**HSubstDir** allows you to specify the data directory to use.



For example: application for taking orders in a restaurant. The new orders are automatically sent to the database found on the PC.

- **Copying the data files onto the mobile device:**

All the data files (or some of them) are copied onto each mobile device beforehand (Pocket PC for example). 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: application for opinion survey in the street. The answers will be available in the standard 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 access to the network (Ethernet card, Wi-Fi, ...).
- 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 and a project that can be used on PC. These two examples handle 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 entered on the mobile devices, all you have to do is synchronize the files.

If the data files used are in HFSQL format, all the 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 standard 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 purchase lists" 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 entered in the two projects.

## LESSON 5.3. ADVANCED PROGRAMMING

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

---

- Creating a management application
- Handling the database



Estimated time: 1h

## Overview

In this lesson, we are going to develop a Windows Mobile application that uses a HFSQL Classic database.

This application will allow us to present some specific features of the programming for Windows Mobile.

### Opening the project

- ▶ Start WinDev Mobile 19 (if not already done). Close (if necessary) the current project to display the home window.
- ▶ Open the "Pocket Managing Products" project.  
To do so, in the home window, click "Tutorial" and select the "Pocket Managing Products (Exercise)" project.  
Tip: if the home window is not displayed, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Pocket Managing Products (Exercise)".



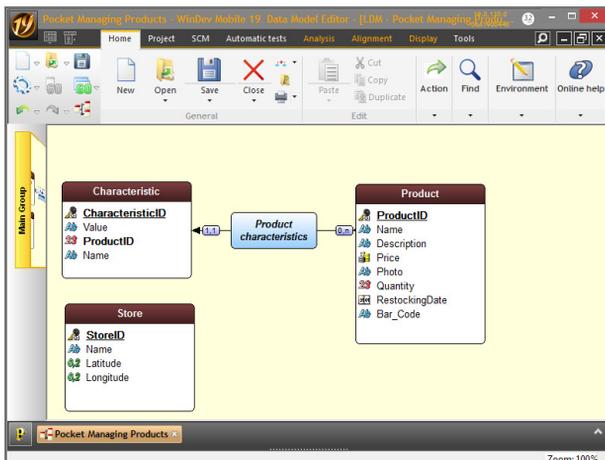
Answer

A corrected project is available. This project contains the different windows created in this lesson. To open the corrected project, on the "Home" pane, in the "Online help" group, expand "Tutorial" and select "Pocket Managing Products (Answer)".

### Describing the project

Let's take a look at our source 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 the WinDev Mobile menu.
  2. The data model editor is displayed.



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

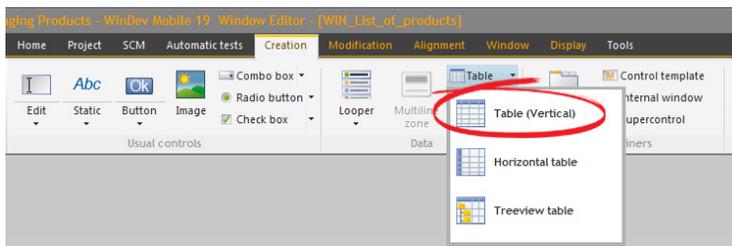
### Creating the window

- To create a new window:
  - Click  among the quick access buttons. Click "Window" in the wheel that is displayed.
  - Display the "Standard" tab, choose "Blank" and validate the wizard.
  - Save the window by clicking  among the quick access buttons.
  - In the window that is displayed, enter the title of the window: "List of products". The name and location of the window are automatically filled. Keep the default values and validate (green button at the bottom of the dialog window).
  - The window is added to the project.

### Creating the Table control

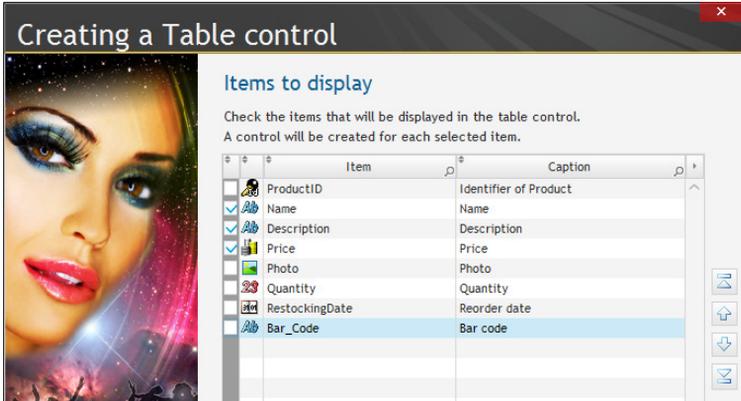
To display the list of products, we are going to use a "Table" control. The main information about the products will be displayed in this control.

- To create the "Table" control:
  - On the "Creation" pane, in the "Data" group, expand "Table" and select "Table (Vertical)".

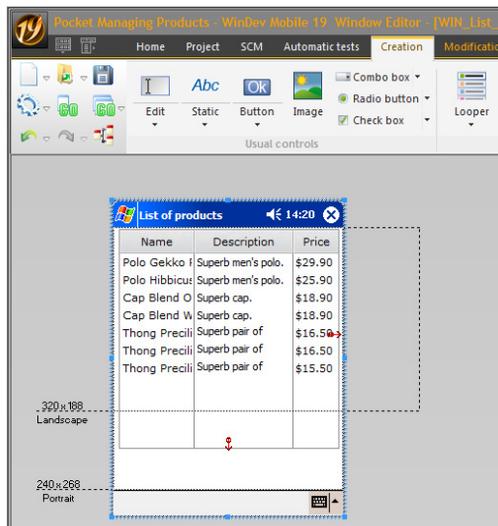


- The control appears under the mouse.
- Click the position where the control must be created in the window (at the top for example). The wizard for creating a Table control starts.

4. In the wizard, select "Display the data found in a file or an existing query". Go to the next screen.
5. Select the "Product" file. Go to the next screen.
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 screen.
8. Keep the proposed sort item (ProductID). The products will be sorted in the Table control according to this item. Go to the next screen.
9. In the "Additional parameters" screen, keep the default options. Go to the next screen.
10. Select the "Vertical" orientation and go to the next screen.
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.

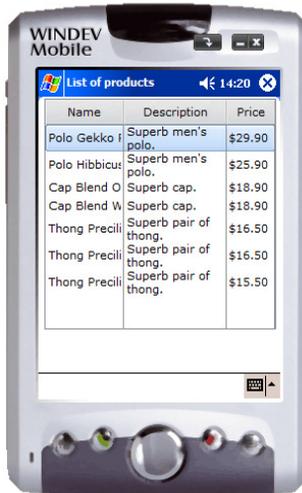




## Notes

The data automatically appears in the window displayed in the editor. This concept is called "Live Data": you see the data found in your 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 view 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 the selected product.

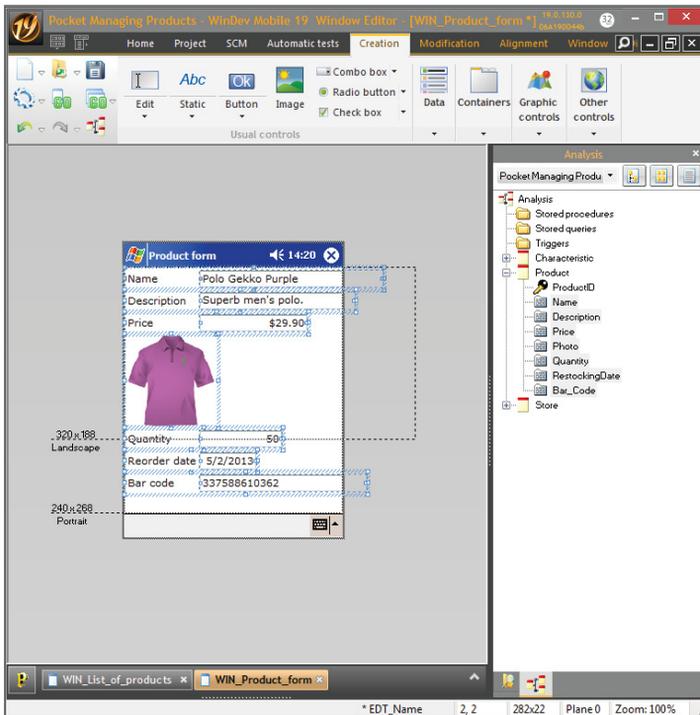
### Creating the window

- ▶ To create the form window:
  1. Create a new blank window. Click  among the quick access buttons. Click "Window" in the wheel that is displayed.
  2. In the wizard that is displayed:
    - Select the "Standard" tab.
    - Choose "Blank".
  3. Validate the wizard.
  4. Save the window. Specify the title of the window: "Product form". Its name is automatically proposed: "WIN\_Product\_form". Validate.

## Creating the controls

► To create an edit control:

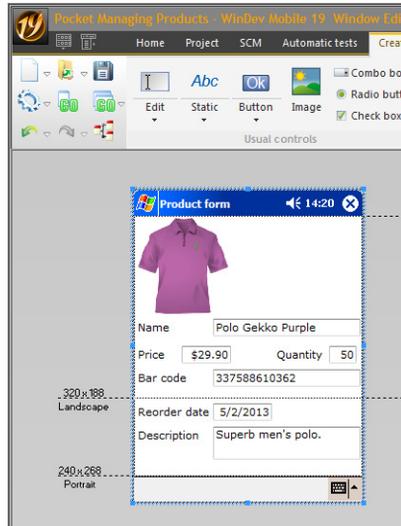
1. Display the "Analysis" pane if necessary (on the "Home" pane, in the "Environment" group, expand "Panels" and select "Analysis"). The different data files described in the "Pocket Managing Products" analysis appear in the pane.
2. With the mouse, select the items of the "Product" file displayed in 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", "BarCode", "Reorder Date" and "Description") so that they are visible in the window:

- Select the requested control.
- Use the sizing handles (black squares) to modify the size of the control.

## 5. Reorganize the controls in the window.



Respect the following order: "Photo", "Name", "Price", "Quantity", "BarCode", "Reorder Date", "Description".

6. We are going to view the navigation order in the window: press the [F5] key. The numbers that are displayed represent the navigation order in the window. Press [F5] again to make the numbers disappear. To correct the navigation order, on the "Window" pane, in the "Order" group, expand "Navigation" and select "Define automatically".

7. Save the window.

- ▶ Run the test of the window (  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 processes associated with the window:
    - Perform a right mouse click in the area beside the window
    - Select "Code" from the popup menu.
    - The code editor appears.
  2. Enter the following code in the "End of initialization of WIN\_Product\_form" process:

```
FileToScreen()
```

**FileToScreen** is used to display in the controls the data found in the data file, for the current record.

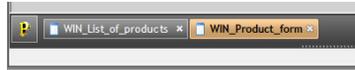
3. Close the code window (click the cross at the top right of the code editor).
4. Save the window.

## Displaying the form from the list of products

Let's see how to display the form of the 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 found in the button bar:



2. On the "Creation" pane, in the "Usual controls" group, click : the shape of the button appears under the mouse. Then, click at the bottom of the window to create the button.

3. Select the control and press the "Enter" key on the keyboard. The caption of the button switches to edit. Type "Modify the element" and press the "Enter" key on the keyboard.

4. Resize the button if necessary (with the handles) in order for the caption to be entirely displayed in the button.

5. Right-click the button and select "Code" from the popup menu.

6. In the code window that is displayed, enter the following code in the "Click" process:

```
Open(WIN_Product_form)
TableDisplay(TABLE_Product, taCurrentSelection)
```



Notes

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 the existing windows found in the project. All you have to do is select the window with the keyboard or with the mouse.

If the name of the window 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 the cross at the top right of the 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 window of the product is displayed.

► Close the simulator.

## Managing the creation and the modification of a product

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

### Modifying the product form

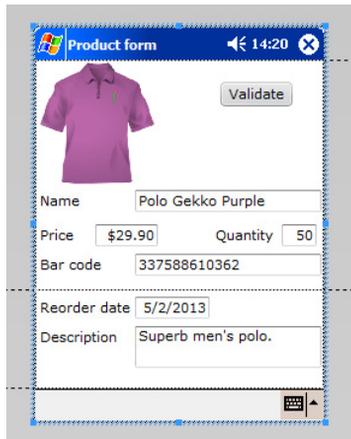
We are going to add a button into the "WIN\_Product\_Form" window: a "Validate" button to manage the validation of modifications.



Notes

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 in the "WIN\_Product\_form" window:
  1. Display (if necessary) the "WIN\_Product\_form" window in the editor: click the corresponding button in the button bar.
  2. On the "Creation" pane, in the "Usual controls" group, click : the shape of the button appears under the mouse. Then, click inside the window to create the button (beside the image for example).
  3. Select the control and press the "Enter" key on the keyboard. The caption of the button switches to edit. Type "Validate" and press the "Enter" key on the keyboard.



4. Resize the button if necessary (with the handles) in order for the caption to be entirely displayed in the button.
- ▶ We are now going to enter the code of the "Validate" button.
    1. Right-click the button and select "Code" from the popup menu.
    2. In the "Click" process, enter the following code:

```
ScreenToFile ()
HModify (Product)
Close ()
```

Let's take a look at this code:

- **ScreenToFile** is used to initialize the items with the values of the 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 the cross at the top right of the 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 Hibiscus Blue" product whose price is 25.90 Dollars.
    - Click "Modify the element".
    - The detailed window of the product is displayed. Modify the price of 25.90 Dollars and enter 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 article.
  - ▶ Close the simulator. The editor of WinDev Mobile 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 the "Product form" window to manage the addition into the Product data file.
- ▶ 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:
    1. On the "Creation" pane, in the "Usual controls" group, click : the shape of the button appears under the mouse. Then, click inside the window to create the button (beside the "Modify the element" button for example).
    2. Select the control and press the "Enter" key on the keyboard. The caption of the button switches to edit. Type "New product" and press the "Enter" key on the keyboard.



3. Resize the button if necessary (with the handles) in order for the caption to be entirely displayed in the button.

4. The code of this option is used to open the "Product form" window and to reset its controls. To enter this code:

- Right-click the "New product" button.
- Select "Code" from the popup menu.
- In the "Click" process, enter the following code:

```
HReset (Product)
Open (WIN_Product_form)
TableDisplay (TABLE_Product)
```

**HReset** initializes the variables of the items found in the Product 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 the 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 the cross at the top right of the code editor).

► We are now going to modify the window of the product form to manage the addition of a new record.

1. Display the "WIN\_Product\_form" window in the editor.

2. We are going to modify the code of the validation button:

- Right-click the button and select "Code" from the popup menu.

- In the "Click" process, replace the existing code by the following code:

```
ScreenToFile ()
IF Product..NewRecord THEN
  HAdd (Product)
ELSE
  HModify (Product)
END
Close ()
```

Let's take a look at this code:

- **..NewRecord** is used to find out whether the current record must be created.
  - If **HReset** was called beforehand, the property returns True (case of the 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**.
3. Save the modifications by clicking  among the quick access buttons.
  4. Close the code window (click the cross at the top right of the 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.
  - Enter a new product.
  - Validate. The new product appears in the list of products.
  - Close the simulator.

## Creating a menu window

---

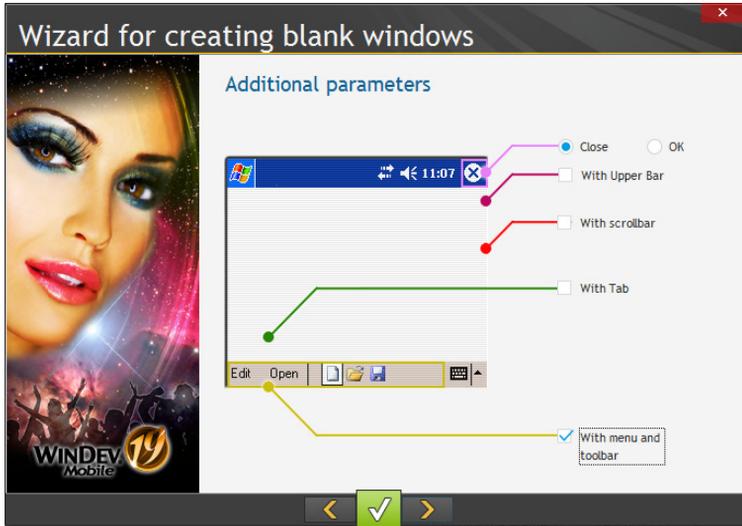
We are now going to create a window for implementing the menu of the application.

### Creating the window

First, 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. Click "Window" in the wheel that is displayed.
  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 screen.

6. In the screen of advanced parameters, select "With menu and toolbar".



Go to the next screen.

7. Specify the title of the window: "Menu".

8. Validate the wizard. The window is displayed in the editor.

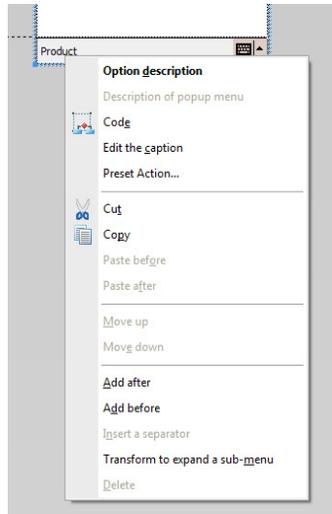
9. Save the window. The name entered in the wizard is automatically proposed in the backup window. Validate.

## 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 option to display the popup menu.
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 mouse click):



6. Select "Transform to expand a sub-menu".
7. In the input area, 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, type "Exit" and validate.

### Programming the menu

We are now going to enter the code of these two menu options.

- ▶ To define the WLanguage code of the "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 the "Exit" option (right mouse click) and select "Code".
  3. Enter 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 the "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. Enter the following code:

```
Open (WIN_List_of_products)
```

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

## Running the test of the application

---

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 the project.

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

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

- ▶ To run the test of the project:
  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.

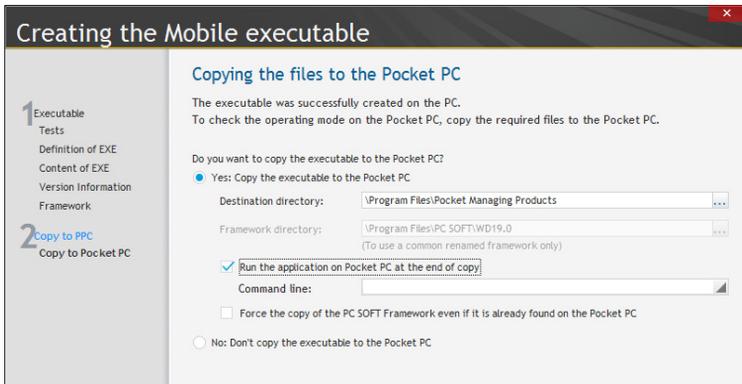
That's it, our application is created, we must now compile it and install it on the device in order to run its test.

## 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 screen indicates whether automatic tests have been created on the project as well as their status. In our case, no test was run. Go to the next screen via the arrow buttons.
  4. The next screen is used to define the name and icon of the application. The icon of the executable can be chosen in the image catalog of WinDev Mobile ("Catalog" button).
  5. The other screens are not required by our application. Click the "2- Copy onto PPC" link found on the left of the wizard.

6. This screen is used to define the options for copying files onto the mobile:

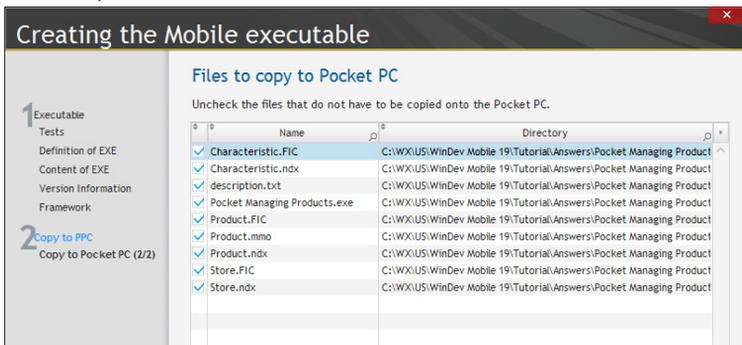


7. The options depend on your configuration:

- If a mobile is connected, select "Yes: Copy the executable to the 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 connected, select "No: Don't copy the executable on the Pocket PC". In this case, the application can be deployed on the mobile devices via a setup procedure.

8. Validate.

9. If "Yes: Copy the executable to the Pocket PC" was selected, the wizard proposes the files that must be copied onto the Pocket PC:



10. Validate the selection of suggested files.

That's it, your application is generated and run on the Windows Mobile device.

## LESSON 5.4. DISTRIBUTING THE APPLICATION

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

---

- Available distribution modes



Estimated time: 20 min

## Overview

WinDev Mobile allows you to develop applications for Windows Mobile.

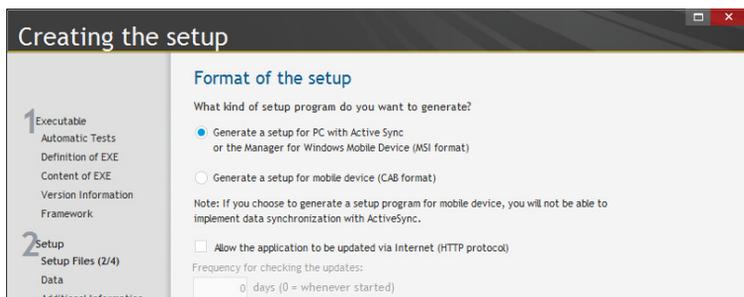
Several methods can be used to install a WinDev Mobile application on a device running Windows Mobile (Pocket PC for example):

- **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 the executable from the PC to the Pocket PC (as already seen in the previous lessons).

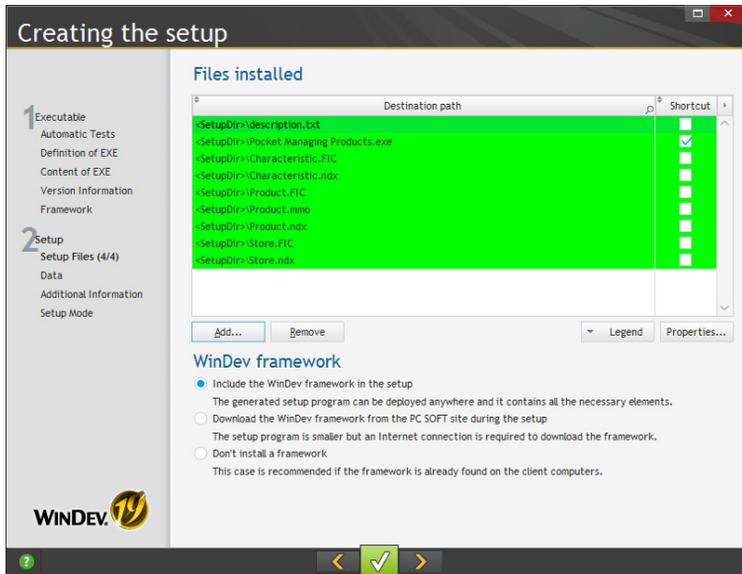
These different setup modes are available via the wizard for setup creation.

► 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 the application is displayed. Go to the next screen via the arrow buttons.
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 screen.
7. The wizard asks for the setup directory of the application. Keep the default option.
8. Specify the files to install. The wizard proposes the executable. You also have to install the data files of the 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 screen.

10. Uncheck the options for configuring the databases. Go to the next screen.

11. The wizard proposes to install additional modules. We won't select any. Go to the next screen.

12. The generation directory of the 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 CAB format

### Initial setup

This setup consists in:

- creating the executable of the application on the development computer via WinDev Mobile.
- generating the setup program of the 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 re-installing the application entirely. 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 executable of the application on the development computer via WinDev Mobile.
  - generating the setup program of the 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 MSI format

---

### Initial setup

This setup consists in:

- creating the executable of the application on the development computer via WinDev Mobile.
- generating the setup program of the 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 the PCs. The application will be automatically installed on the mobile device connected to the current computer, via ActiveSync.

Note: 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 re-installing the application entirely. In this case, the update is identical to the initial setup.
- update by Internet (HTTP protocol) in CAB format. This update consists in:
  - creating the executable of the application on the development computer via WinDev Mobile.
  - generating the setup program of the 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 has discussed a variety of subjects, but not all the features of WinDev Mobile, 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 one of the modules.

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 room to discuss all the 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 19!**

