

CHAPTER 1.	RUN MOBILEFX STUDIO FOR FIRST TIME.....	2
CHAPTER 2.	INTRODUCTION TO MOBILEFX STUDIO	2
CHAPTER 3.	ON DEVICE PORTALS.....	3
CHAPTER 4.	MOBILEFX STUDIO WORKSPACE.....	3
CHAPTER 5.	MOBILE DEVICE EMULATORS	4
CHAPTER 6.	APPLICATION DEVELOPMENT LICENSING & R C L.....	4
CHAPTER 7.	PROJECT CONFIGURATIONS	4
CHAPTER 8.	DESIGNING YOUR FIRST ON DEVICE PORTAL.....	6
TASK 8.1.	MARKETING GOALS	6
TASK 8.2.	ADDING ODP PAGES	6
TASK 8.3.	IMPORTING GRAPHICS	7
TASK 8.4.	IMPORTING FILES FROM OTHER PROJECTS	8
TASK 8.5.	IMPORTING DATA USING PLUG-INS	8
TASK 8.6.	DEFINING THE ODP STYLE SHEET	8
TASK 8.7.	BREAKING VISUAL INHERITANCE	9
TASK 8.8.	DOUPLICATING PAGES.....	10
TASK 8.9.	SETTING INTUITIVE NAVIGATION AND EVENT PAGES	10
TASK 8.10.	ADDING COMMAND BARS.....	12
TASK 8.11.	FETCHING DATA FROM EXTERNAL DATA SOURCES	14
TASK 8.12.	ADDING DYNAMIC AND EXTERNAL PAGES	14
TASK 8.13.	ADDING WIDGETS	16
TASK 8.14.	ADDING SKIN IMAGES.....	17
TASK 8.15.	ADDING HYPERLINKS.....	18
TASK 8.16.	ADDING SOUNDS	18
TASK 8.17.	ADDING DATA CAPTURE FORMS	19
TASK 8.18.	ADDING BLUETOOTH DOWNLOADS	20
TASK 8.19.	ADDING A VIRAL MARKETING FORM.....	20
TASK 8.20.	ADDING LOCALISATION TOKENS.....	22
TASK 8.21.	ADDING RUN TIME SUBSTITUTION VARIABLES	23
TASK 8.22.	RE-INDEX YOUR APPLICATION	23
TASK 8.23.	EXECUTION FLOW DIAGRAM	23
CHAPTER 9.	COMPILE AND RUN.....	23

Chapter 1. RUN MOBILEFX STUDIO FOR FIRST TIME

Welcome to mobileFX Studio tutorial! This session will last approximately one hour, and it will attempt to give you some basic training on using mobileFX Studio for Mobile Marketing. In this tutorial you will be taught how to create an On Device Portal. To attend, you need a licensed version of mobileFX Studio, basic web development skills or basic understanding of HTML, but no programming skills are needed.

Starting mobileFX Studio for the first time, will pop up a Registration Form. Registering the product is required both for evaluation and commercial purposes, and you should fill in this form with valid information.

Note, that mobileFX Studio requires internet access at all times.

Please type your mobileFX user name and password from the registration e mail received from mobileFX web services, and click the hyperlink to validate. If you do not have a mobileFX user account, take a minute to fill in the Registration Form.

Click OK to continue.

If you are running mobileFX Studio under an evaluation license, the About window will pop up with the evaluation terms. If you own a Workstation License of mobileFX Studio, click Unlock to open the Licensing Agent, and enter your registration information.

In Licensing Agent window, you will need the Activation Request Key, the buyer Name, and buyer email. Type them in the appropriate fields, and click Request Unlock Code to continue.

The evaluation version of mobileFX Studio will run full featured for 60 days, but you will only be able to load, compile, and run, the sample Projects provided with the product. You will not be able to create your own applications.

Every time mobileFX Studio starts, it is checking mobileFX web site for latest versions of the product. By clicking the download hyperlink, the latest version of mobileFX Studio run times and Frameworks will be automatically downloaded on your computer.

Chapter 2. INTRODUCTION TO MOBILEFX STUDIO

mobileFX Studio, is an integrated development environment, for Visually designing mobile applications. Out-of-the box, mobileFX Studio will produce J2ME applications for MIDP - CLDC mobile phones. However, with additional plug in Compilers and Frameworks provided by mobileFX, it is possible to produce applications for RIM Black Berry devices, Google Androids, VODAFONE Betavine 3 60 widgets, and even iPhone applications. In other words, mobileFX Studio is aiming to become a unified development platform, that anyone will be able to produce breath-taking mobile applications, with cutting-edge technology and features, for any mobile phone. The big idea behind mobileFX Studio is to design you application visually, only once, and get it to work on any screen and any mobile phone.

Compared to other I D E software, such as Netbeans and Eclipse, mobileFX Studio is a complete solution, for rapid prototyping and development, of on-device portals, mobile games, and adver-games. However, the big difference between mobileFX Studio and other similar products, is that mobileFX Studio, is a marketing

oriented Application Designer, emphasizing on the mobile marketing features, that you can easily add, in any of you mobile applications.

For that reason, apart from the classic IDE related features offered by other software, such as the code editor and user interface Designers, Studio offers developers complete Frameworks, including source code, for creating applications that are able to send SMS messages, run instant-win campaigns, play videos, fetch data from web services or Bluetooth Hotspots, read RSS feeds, display mobile advertising banners from ad servers, and many more that we will have the chance to explore.

Chapter 3. ON DEVICE PORTALS

An On Device Portal, also referred with its three letter abbreviation as ODP, is a mobile application organized as a small monolithic web site, consisting of Pages, text, hyperlinks, images, videos, and other resources, all compiled into a single self-contained file, that runs stand alone and off line, on a mobile phone. The advantages of On Device Portals versus Mobile Internet Portals, is the chance to deliver value added services to consumers, that otherwise they would have to pay network charges to get. However, the undisputable advantage of On Device Portals, is brand awareness. With a branded On Device Portal installed on the menu of a consumer's mobile phone, a brand possesses a personalized full-duplex communication channel with its customer base, decoupled and independent from network operators.

Chapter 4. MOBILEFX STUDIO WORKSPACE

First, lets get familiar with mobileFX Studio workspace.

Every time you start mobileFX Studio, the New mobileFX Project window appears. From this window, you can choose to create a new template-based Project, load an existing Project from the file system or from V.S.S., or, open any the Studio sample applications. From the New mobileFX Project window, you can also clone any existing Project into a new Project. This is a very useful productivity feature, that allows you to clone and re-brand your Projects within minutes. For example, if you have a Tetris game for Coca-Cola, you can clone and re-brand this game for Fanta, just by changing its graphics and logos.

For this tutorial we will develop an On Device Portal for commercial purposes. From the New tab, click on New On Device Portal Project template; type in KFC; and click OK. Please note, that Project names must be suitable for file names too. As a general rule, Project names must consist only of letters and numbers, without any spaces.

mobileFX Studio, will generate the Project files under the Project folder, and reveal the Project management windows.

Like any other modern I.D.E., Studio is using Explorers for organizing context-specific elements of your Project. Some of the most frequently used explorers are, Project Explorer, for managing Project resources, Properties Explorer, for editing Project and element properties, Java Modules Explorer, for managing Java source files, and Packages Explorer, for accessing Java packages and referenced API libraries.

mobileFX Studio explorers are inside dockable panes, for organizing and maximizing its workspace according to developer preferences.

In Project Explorer, the resources of a Project are organized in a hierarchical view. The top level node is the Project node, under which, is the On Device Portal node, containing the On Device Portal Pages, the

localization node for entering the multilingual tokens of your Projects, and resource nodes for organizing sounds, images, videos, fonts and other resources. In game Projects, additional nodes are used for organizing sprites, weapons, maps and tiled layers.

mobileFX Studio toolbar, give you immediate access to most common commands, such as saving and importing files, compiling and running your Projects, showing Studio Explorers, saving and restoring your workspace layout, accessing V.S.S. commands, publishing your Project, and selecting a mobile phone emulator.

Chapter 5. MOBILE DEVICE EMULATORS

Mobile Phone Emulators are used for testing and debugging your application from within the I D E.

mobileFX Studio, is fully compatible, with Sun Microsystems Wireless Tool-kit Emulators, Sony Ericsson S D K for Java, and NOKIA S40 and S60 emulators. Special beta versions of mobileFX Studio, are compatible with RIM Black Berry, and Google Android emulator, while mobileFX Research and Development is working on an iPhone emulator for windows. Also, for Sony Ericsson JP7 and JP8 devices, mobileFX Studio can be used for On Device Debug. That is debugging mobile applications, while they run on a real mobile phone.

Chapter 6. APPLICATION DEVELOPMENT LICENSING & R C L

By clicking the Project node in Project Explorer, you can edit the Project Properties. In Properties Explorer, we change vendor to mobileFX. A very important step is licensing your mobile application for commercial use. Click on the commercial license ellipsis to obtain a commercial license R C L key from mobileFX web site. It is illegal to produce mobile applications using mobileFX Studio, without a valid commercial license.

Chapter 7. PROJECT CONFIGURATIONS

To edit advanced Project properties, click on the Compiler menu, and select Configurations. This will pop up the Project Configuration Settings window, where advanced Project properties are organized in tabs.

Through out those tabs, there is a Variants list. Variants are Project configuration snapshots. A single Variant, is a collection of Project properties, including Project references to JSR libraries and mobileFX Frameworks, Compiler switches, preCompiler variables, security permissions, and code signing certificates.

When a new Project is created, mobileFX Studio creates the default Variant of your Project, targeting MIDP two and CLDC one devices. If you wish to make your Project compatible with more mobile phones, then you can add several Variants, with settings targeting explicit mobile phone devices and models. As a general rule, for J2ME compatible mobile phones, the default Variant is pre-configured to compile applications that run on most mobile phones.

When compiling a mobile application with mobileFX Studio, you can either compile the selected Variant, or all Project Variants with a single click.

The first thing you need to define on every new Project, are the JSR libraries, and mobileFX Frameworks, that you will use. Libraries and Frameworks, enable marketing features on your mobile applications.

To add, or remove references, click on References tab. There is a list of currently referenced JSR libraries and mobileFX Frameworks. Every mobileFX J2ME application, references mobileFX common wireless Framework, and mobileFX ODP Framework Engine.

Click Edit button, to open the References window. In References Window, run time references are organized in two lists, the Wireless API List, and the mobileFX Frameworks list.

Click on mobileFX Frameworks tab first.

mobileFX Studio, encapsulates reusable, domain specific source code and features, into Framework Packages. Each Framework Package, when referenced by a Project, automatically enables certain Project wide features. For example, there are Frameworks for sending SMS messages, and interconnecting mobileFX Applications, using Bluetooth or HTTP.

Certain Framework Packages, classify the entire wireless application. Those Frameworks are called Engines. Such Engines are mobileFX ODP Engine, 2 D Scrolling Platform Game Engine, and others.

To enable pulling added value content from Bluetooth Hotspots, check mobileFX Bluetooth Network Layer Framework, and mobileFX Bluetooth OBEX Framework.

To enable pulling added value content from web servers, check mobileFX HTTP Framework.

To enable SMS sending feature, check the mobileFX SMS Framework.

Note, that each of those Frameworks require certain JSR libraries. Click on Wireless API Libraries tab to add those references too.

Check JSR 82 Bluetooth API, and JSR 120 Wireless Messaging API.

Also, for playing sounds and background music in your applications, check the JSR 135 Mobile Media API.

Now click OK to import the references in your Project. mobileFX Studio will take a minute to repair the source code of your Project, enabling the newly added features as described by the Frameworks and libraries, in design time editors as well.

Certain features require certain permissions to be set in your Project's manifest. Click on permissions; available permissions are categorized per API. In JSR 120 API, enable the Java x.microedition.io.connector.sms, and Java x.wireless.messaging.sms.send permissions.

In JSR 82 API, enable the Java x.microedition.io.connector.obex.client, Java x.microedition.connector.obex.bluetooth.client, Java x.obex.client session, and Java x.obex.operation permissions.

Then, click OK to close the Configuration Settings Window, and wait while mobileFX Studio repairs the Project according to your new configuration settings.

Chapter 8. DESIGNING YOUR FIRST ON DEVICE PORTAL

Task 8.1. MARKETING GOALS

From a Marketing Point of view, we want to design an On Device Portal, that will encapsulate certain Mobile Marketing concepts and Goals.

The first concept is Mobile Advertising, and subsequently, our goal will be to advertise the product. To make the product more desirable and attract customers in the point of sales, we will give consumers different discount offers, every few months.

Next, we will use Proximity Marketing to strengthen brand awareness. We will offer consumers Added Value Content, such as Wallpapers, Ringtones, and Adver Games, for free. In order to attract consumers in restaurants, we will only allow content download, from Bluetooth hotspots, installed on the restaurant's windows.

In order to establish Customer Loyalty, and prepare future Direct Marketing campaigns, we will use Fidelity Forms and Discount Vouchers. By offering consumers free discount vouchers, we will collect consumer's name, surname, age, mobile phone number, and their mobile phone's Bluetooth address. With the collected data we will build a Customer Loyalty database, that will be used later on, for Direct Marketing. More importantly, the collected data will be projected to product sales, in order to prove to the customer that the overall mobile marketing campaign, was a success.

Also, we will use Mobile Internet Marketing and forward consumers to the official mobile web site, for finding additional product information, on line games, and the hall of fame, of the branded mobile adver games.

Finally, we will use Viral Marketing, in order to get consumers to advertise the brand and the product to their friends and family. To do so, we will add an invitation form that consumers will use, for inviting their friends to visit the restaurants, download branded games, and find out about discounts and offers.

Task 8.2. ADDING ODP PAGES

In Project Explorer, right-click on the ODP node, and select New ODP Page. Type, Home, and click OK.

mobileFX Studio will open ODP Page Designer, and show added Pages under the Static Pages folder, in Project Explorer.

The new dockable panes on the right, are the Widgets Explorer, with all ODP controls that you can add in a Page, ODP Theme Composer for creating the shared Visual Style of your application, and Floating Elements Explorer for easily selecting floating controls within a Page.

Repeat the last action to add the following Pages as well.

About.

Downloads.

OBEX _ Cancelled.

OBEX _ Fail.

OBEX _ OK.

OBEX _ Requesting.

OBEX _ Searching.

Offers.

PEPSI.

Please _ Wait.

Provider _ Error.

SMS _ Fail.

SMS _ OK.

Title.

Update _ Confirm.

Vouchers.

Web Site.

Offers _ Update.

Click Save All button, to automatically name the new Pages and save them in Project Folder.

Task 8.3. IMPORTING GRAPHICS

To import graphics files from other programs, simply drag and drop your files on mobileFX Studio title bar. Graphics will be copied in your Project folder. mobileFX Studio can import any known graphics file format, including PSD, JPEG, PNG, GIF, BMP, TIFF, and others.

By default, when you drag and drop an image to mobileFX Studio, the image pixels are locked and you can not edit it. This is done in order to preserve the original pixels of the image that might have alpha channel information, like PNG 24, or to preserve the JPEG optimized compression.

Locked images have the Preserve Original property set to true, which also appears checked in the image pop up context menu, in Project Explorer. Finally, locked images, when opened in Pixel Editor, they show as Read Only. To edit the pixels of an image, simply reset the Preserve Original property to false, but you will automatically lose alpha channel information. In mobileFX Studio Pixel Editor, image transparency is mask based.

This image locking mechanism, applies because mobileFX Studio Pixel Editor, is optimized for pixel level editing, of 2D animated sprites by pixel artists, and not for photo editing by graphics artists. For photo editing, we recommend using other more appropriate software, such as Adobe Photoshop, or Corel Draw.

However, mobileFX Studio is compatible with most, if not all, known graphics formats, and you can replace previously imported graphics, simply by re drag and dropping them in Studio, again.

All graphics, appear under the Images node in Project Explorer, or under the Images tab, as thumb nails, in Widgets Explorer.

Now that graphics are imported, click on the root node in Project Explorer, and set Icon property, to k f c _ icon.

Task 8.4. IMPORTING FILES FROM OTHER PROJECTS

In mobileFX Studio, you can import other files too, such as sounds, music tracks, midi files, video files, etc.

You can also import files from other Projects, including entire Pages, sprites, tiled layers and maps.

When importing native Studio files, those are being copied into your Project Folder.

It is also possible to import uncategorized resources, such as binary files and XML files as Embedded Resources, or as Related Resources. Embedded Resources are included in the application file, in the /gfx folder. Related Resources are used only for keeping references to resources used by your application, but they are not added in the application file.

Task 8.5. IMPORTING DATA USING PLUG-INS

Using one of mobileFX Studio plug-ins, you can convert the dataset of a database query into a binary data-stream file, and add it in Embedded Resources. Then, you can easily view the data in a data-grid added in a Page.

MobileFX Studio is loaded with plug-ins that trans-code data from various data sources into data-streams, that can easily be used by widgets. You can also find plug-ins that convert videos to 3GP, edit XML RSS Feeds and convert them to binary data-streams, XML to data-stream converters, Barcode and Data-matrix generators, instant win code generators, and many more.

Task 8.6. DEFINING THE ODP STYLE SHEET

ODP Pages can share the same Visual Style, like good old HTML CSS files. A Style consists of Visual settings, colors, fonts and theme images. We call this Style, the master ODP Style, and to edit its properties, simply click on the ODP node in Project Explorer. This will load the ODP Style in Properties Explorer.

Set the following.

Set Start Up Page, to Title.

Set First Time Start Up Page, to Title.

Set Margin Top, to 66 pixels.

Set Transition, to Push Existing To Left.

Set Transition Steps, to 6.

Set Caption Y, to 34.

Set Caption Font Color, to Yellow.

Set Caption Font Style, to Bold.

Set Caption Font Size, to Large.

Set Caption Font Effects, to Shadow.

Set Caption Font Effects Color, to dark red.

Set Font Color, to white.
Set Font Style, to Bold.
Set Font Size, to Small.
Set Font Effects, to Shadow.
Set Hyperlink Color, to white.
Set Hyperlink Sel Color, to white.
Set Hyperlink Sel Border Color, to white.

In Theme Composer, make sure that the drop-down box is set to ODP, and click the add button, to add an image. On Properties Explorer, set the Image property to flames_back. Now your Page has a background!

However, if you change the screen resolution in ODP Designer to 320 by 480 pixels, the background image, will appear unfit for this screen size. To fix this, in Properties Explorer, change the Scale Width and Scale Height properties to stretch screen. Now your background image is fit for any screen size.

Add another image in Theme Composer, and from Properties Explorer, set the Image property to kfc _ top _ logo, and the Anchor property to top-middle.

Repeat this by adding another image, set the Image property to PEPSI _ bottle, and the Anchor property to bottom-right. Then, set conditional display property, to height conditional display. This will hide this image from the theme, if the screen height is less than 320 pixels. To preview this condition, switch the ODP Designer to 176 by 208 pixels.

Image anchoring, is a design pattern, that allows you to fix images on relative positions on the screen and control when they should show.

Every Page shares the master ODP Style and theme using Visual Inheritance. However, it is possible to partially overwrite Visual inheritance for a Page. Simply select the Page in Project Explorer, and alter the properties needed. Inherited properties appear in Properties Explorer with a gray shade, and the text Inherited.

Task 8.7. BREAKING VISUAL INHERITANCE

In Project Explorer, click on the Title Page to load it. Then, right-click on it again, and clear the Visual Inheritance option. The Page will become black. This means that the Visual Style of the Title Page, is no longer linked with the master ODP Style.

Go to Properties Explorer.

Erase the Caption text.

Set the Timeout Millis property, to 2000 milli seconds.

Set the Next Page property, to Home.

Set the Background Color property, to white.

In Theme Composer, make sure that the drop down box shows [Title], and click the Add Image button. Select the K F C _ logo image, and anchor it in the center of the screen.

Task 8.8. DOUPLICATING PAGES

Right-click on the Title Page, and select Duplicate ODP Page. Name the new Page, Sponsor.

In Theme Composer, change the Image to logo _ PEPSI.

In Page Properties, set Transition to Fade in, and Transition Steps to 6.

Note that Title Page should have as next Page, the Sponsor Page.

Also, set the Home Page Transition to Fade in.

Task 8.9. SETTING INTUITIVE NAVIGATION AND EVENT PAGES

All mobileFX On Device Portal applications, have Intuitive user interface, which is ideal for mobile phones that have limited navigation abilities. In order to enable intuitive navigation logic, you need to define Page types and navigation handling.

For example, open Title Page, by clicking its node in Project Explorer. Go to Properties Explorer, locate the Page Type property, and set it to Title. Now, locate the Left Button property, and clear it. This property, controls how the application should respond, when pressing the left button on your mobile phone's joystick. Also, clear the Caption property.

Open Home Page.

Set Page Type, to Home.

Clear the Left Button Property.

Open OBEX Cancelled Page.

Set Caption to Cancelled.

Set Page Type, to fx Bluetooth Cancelled.

Set Time Out Millis, to 2000 milli seconds.

Set Left Button, to Home navigation command.

Set Next Page, to Home.

In Widgets Explorer, select the Images tab, and scroll down to find image b t fail. Double-click on the image to insert it in the Page. Images that are inserted as block elements in a Page, are called inline images. Inline images are not scaled at run time. Therefore you should create your inline images, taking into account that your application will run on different screen sizes. We recommend using inline images no bigger than 176 by 208 pixels.

Under the image, type The Operation was cancelled.

Open OBEX Fail Page.

Set Caption to Failure.

Set Page Type, to fx OBEX Fail.

Set Time Out Millis, to 5000 milli seconds.

Set Left Button, to Home navigation command.

Set Next Page, to Home.

Inset the b t fail image inline, and type, Failed to download free content from KFC Bluetooth Hotspots. Please make sure you are within proximity range and try again.

Open OBEX OK Page.

Set Caption to Success.
Set Page Type, to fx OBEX Success.
Set Time Out Millis, to 5000 milli seconds.
Set Left Button, to Home navigation command.
Set Next Page, to Home.
Inset the b t ok image inline, and type, You will receive mobile content within minutes. Please stay put.

Open OBEX Requesting Page.
Set Caption to Requesting.
Set Page Type, to fx OBEX Sending.
Clear Left Button property.
Inset the b t r q image inline, and type, Requesting mobile content. Please wait...

Open OBEX Searching Page.
Set Caption to Searching.
Set Page Type, to fx Bluetooth Search Start.
Clear Left Button property.
Inset the b t find image inline, and type, Searching for Bluetooth Hotspot. Please wait...

Open Please Wait Page.
Set Caption to Working.
Set Page Type, to Generic Please Wait.
Clear Left Button property.
Type, Please wait...

Open Provider Error Page.
Set Caption to Network Error.
Set Page Type, to fx ODP Provider Error.
Set Time Out Millis, to 5000 milli seconds.
Set Left Button, to Home navigation command.
Set Next Page, to Home.
Type, Failed to complete operation due to Network Error. Please try again.

Open SMS Fail Page.
Set Caption to Failure.
Set Page Type, to fx SMS Fail.
Set Time Out Millis, to 5000 milli seconds.
Set Left Button, to Home navigation command.
Set Next Page, to Home.
Type, Failed to send SMS message. Please try again.

Open SMS OK Page.
Set Caption to Success.
Set Page Type, to fx SMS Success.
Set Time Out Millis, to 5000 milli seconds.
Set Left Button, to Home navigation command.
Set Next Page, to Home.
Type, The SMS was successfully sent.

Open Update Confirm Page.
Set Caption to Update Needed.
Set Page Type, to fx ODP Provider Update Confirm.
Set Left Button, to Home navigation command.

Don't type anything in this Page yet, we will deal with it later on.

Right-click in Project Explorer, and select Refresh Project Explorer command.

Note that the Home Page now appears with a home icon, and the other Pages you edited, with a gray arrow on their icon. The gray arrow indicates that this Page is an event Page. Event Pages are part of a messaging and notifications mechanism. They appear on the screen automatically and asynchronously, as a response to an event, triggered by a Framework action, or core sub-systems of the mobile application. Also, the Page type text appears inside parenthesis next to Page name.

For example, the SMS OK Page will appear if the application successfully sends an SMS message; if it fails, the SMS Fail Page will appear instead. During SMS transmission, the Generic Please Wait Page will be showing.

Even though event Pages are optional, you should always add the necessary event Pages in your applications, in order to allow progress interactivity, process cancellation, and for notifying your users about the background operations of your application.

MobileFX applications run in parallel threads, and often, multiple tasks are executed at the same time. The main application thread, has a tick interval, of 50 milli seconds. You can change this value by editing the Mills Per Tick Project property. However, 50 milli seconds per tick, means that this application will run with 20 frames per second, which is ideal for smooth games and multimedia applications. We strongly recommend, leaving this value, as is.

Task 8.10. ADDING COMMAND BARS

All modern mobile phones have at least two action buttons, just under their screen. Those buttons are used by intuitive navigation logic as shortcut buttons, and they are referred to as S1 and S2. In order to control S1 and S2, you need to add as needed, a Command Bar widget.

Open Home Page, add a Command Bar widget, and select it.

Go to Properties Explorer.

Set Skin Image, to command _ bar.

Set S1 Button Caption, to About.

Set S1 Button Hyperlink, to Page About.

Set S2 Button Caption, to Exit.

Set S2 Button Hyperlink, to navigation command, Exit.

Set Back Gradient Color From, to transparent.

Set Back Gradient Color To, to transparent.

Set Height, to 24 pixels.

Now, right click on the Command Bar, and select Copy. Open About Page, and Paste it. Select the command bar again.

Go to Properties Explorer.

Set S1 Button Caption, to Back.

Set S1 Button Hyperlink, to navigation command, Back.

Copy this Command Bar again, and paste it to the following Pages.

Downloads.

Offers.

PEPSI.

Update Confirm.

Vouchers.

Web Site.

Paste the Command Bar in OBEX Cancelled Page and select it.

Go to Properties Explorer.

Set S1 Button Hyperlink, to navigation command Home. Since event Pages appear automatically and asynchronously as a response to an event, the back navigation command is not logical, and should be replaced with home.

Copy this Command Bar again, and paste it to the following Pages.

OBEX Fail.

OBEX OK.

Provider Error.

SMS FAIL.

SMS OK.

Paste the Command Bar in OBEX Searching Page and select it.

Go to Properties Explorer.

Set S1 Button Caption, to Cancel.

Set S1 Button Hyperlink, to navigation command Bluetooth cancel. This will cancel Bluetooth Hotspot discovery, and cause Page OBEX Cancelled to show.

Paste the Command Bar in OBEX Requesting Page and select it.

Go to Properties Explorer.

Set S1 Button Visible, to False.

Copy this Command Bar again, and paste it to the Please Wait Page.

Now all your Pages, except the Title Page, should have a relevant Command Bar, handling S1 and S2 buttons.

Task 8.11. FETCHING DATA FROM EXTERNAL DATA SOURCES

In order to access from your application, data from external data sources, such as web sites and data bases, mobileFX ODP Framework offers you a very sophisticated service, the ODP Provider. Using a simple straight jacket methodology, ODP Provider, implements a transparent Data Provision Layer, which abstracts the complexity of networking your mobile application, and allows you to define how your application should exchange data, with back-office systems.

ODP Provider, is utilizing four high level concepts, in order to define a Data Provision Strategy. Those concepts are,

Data Provider, that defines HOW data should be fetched.

Data Source, that defines WHERE data should be fetched from.

Data Request Format, that defines an application request PROTOCOL between our application and the data source.

Data Storage, that defines where data should be saved, once fetched.

A Data Provider encapsulates the knowledge of fetching data in a particular way, and from a particular data source. Available Data Providers are, the JAR Provider for loading local resources, the HTTP Provider for downloading or posting data to web servers, the S P P Provider for downloading or posting data to servers over Bluetooth Hotspots, the File System Provider for loading or saving date to the mobile phone's flash memory, and many others.

External data, can be Pages stored on servers, or widget specific data, that are decoded and presented by a widget control. For example, the Image Viewer control, can use the ODP Provider system, to download and display images from a web server, or a city map from Google maps.

In the following section, we will focus on how to configure the ODP Provider system, for downloading Pages, from a server.

In this application, set the Default Provider, to HTTP POST.

Set Default Data Source, to `http, colon, //, www dot mobileFX dot com, / web, / campaigns, / kfc _ demo, / provider.asp`.

Set Default Request Data Format, to URL Encoded Form.

Expand Default Request Data Packet, and select `sys _ app name, sys _ serial number, and sys _ version`. Those are the default variables that will be submitted to the server, every time the ODP Provider, posts data.

Finally, set the Default Storage, to R M S.

This stands for Record Management Store. It is a persistent memory area allocated to J2ME applications.

Task 8.12. ADDING DYNAMIC AND EXTERNAL PAGES

The following information is extremely important.

Pages in an On Device Portal application can be, Static, Dynamic, or External.

Static Pages are permanently embedded inside the application file.

Dynamic Pages are included in the application file but they can be conditionally updated from a server. As a general rule, for every Dynamic Page, we need an External Page to update and replace it. Dynamic Pages, or their External Page replacements, are cached in the memory of the mobile phone, until they expire.

Even though On Device Portals are supposed to run off-line, and be monolithic, External Pages are NOT included in the application file. To view External Pages, the application must download them from a server over the air. Over the air means over GPRS, 3-G, WiFi, or Bluetooth. To link any kind of Page with an external Page, you do so as if it was a Static Page, using Hyperlinks. Therefore at design time, there is no difference between Static Pages and External Pages. In that sense, an External Page can contain hyperlinks to Static Pages, or links to other External Pages of the application – or even External Pages of other applications. This ability allows you to offer through On Device Portals, hybrid Mobile Internet Applications.

The most important reasons for adding External Pages in an On Device Portal, are to offer post-release extensibility to your applications, and avoiding big application files. Such size demanding applications could be product catalogues, mobile content catalogues, etc.

For example, if you need to develop a product catalogue application for 1000 products, you could have a different Page per product, and download those Pages in your application on demand. Having 1000 products, would mean having 1000 product images as well. If product images were stored inside the application file, this would greatly increase the file size, risking rendering the application obsolete, if any of those products was discontinued. In order to avoid all those problems, and offer a crystal-clear and straight-forward approach, you should use External Pages. mobileFX Studio Compiler is capable of detecting if an image is shared across Pages, or if it used only by a Page, and pack it in different ways.

What we need to achieve in this tutorial's ODP, is getting the Offers Page to expire, and automatically update it with new offers, at a pre defined date time. To do so, the Offers Page must be classified as Dynamic, and use an External Page for replacing it.

For that reason we will use 3 Pages: the Offers Page as the Dynamic Page, the Offers Update Page as the replacement External Page, and Update Confirm Page as the manual update confirmation Page.

As a design practice, please consider finalizing your design flow, prior to releasing your application. Once an application is released, you should not add any more Dynamic Pages; only update the External Pages as needed.

Select the Offers Page, and set the Page Flow property, to Dynamic.

In a real case scenario, we would set Page Expiration, to After Time stamp, and Page Expiry Date Time, to 2010, 09, 01, 00, 00. This stands for the 1st of September 2010, and it is the time stamp that this dynamic Page will expire and request an update.

However, in order to demonstrate the update feature without long waits, set Page Expiration to After View Count. Also, set Page Expiration Interval to 2. With those settings, the Page will expire straight after viewing it two times in a row, and the ODP will ask for updating it.

Set on expiration, to Show Page.

Set Update Confirm Page, to Update _ Confirm. This is the Page to display instead of the expired Page, asking for user confirmation before downloading an updated version of the Page.

In Widgets Explorer, select the Images tab and scroll down to find image offer _ 001. Double-click on the image to insert it in the Page. Under the image you can add some product descriptive text, explaining the offer to consumers.

Select the Update Confirm Page, and set the Caption, to Update Needed.

Type in the text, The offers have expired, please download the latest offers available to your location.

Type in the text, Update Now, select it, and click the Edit Hyperlink button. In ODP Commands tab, scroll down to ODP Provider Commands, and select the do, colon, //, provider _ update command. Click OK to set the Hyperlink.

Finally, we need to set the External Page with the new offers.

In Project Explorer, select the Offers Update Page.

Set its Page Flow property to External.

Set its Page Images property, to include theme images in JAR file, and the rest of the images in ODP Page. By setting this property, all shared images will be compiled into the application file. However, inline images and other resources used only by that Page, will be compiled into a different Page file. If your External Page had a different theme, then you can set this property to Embed All Images in ODP Page.

Set property: Page Is Update For, to Offers Page. This is how we link a Dynamic Page, with an External Page.

If you right-click on Project Explorer and refresh, External Pages will appear on a separate folder.

As a general rule and design pattern, common graphics of dynamic and external Pages, should be included in the application file, where inline images and other resources, should be excluded from the application file. At ODP level, you can set how the Compiler should pack Pages, by setting the Packing property, and the Page Images property for all Pages. Those properties are inheritable by all Pages of the ODP, and you can explicitly break inheritance if a Page requires otherwise.

Task 8.13. ADDING WIDGETS

Like HTML web Pages, in mobileFX ODP Pages, you can add widgets such as captions, edit boxes, check boxes, sliders, picture lists, buttons, tab strips, option menus, data grids, image viewers, command bars, and many more.

Several more sophisticated widgets, such as RSS feeds, require additional Frameworks, to be linked with your Project.

You can edit existing widgets, extend their properties, or even create your own widgets, from within mobileFX Studio. All mobileFX widgets are defined in XML, and their functionality is programmed in Java, based on a base class in the ODP Framework.

Let us add some buttons, in our Home Page. First, click on Home Page node in Project Explorer, to load it in Page Designer. From the Widgets Explorer, double-click on Skin Button. With your mouse, click on the black rectangle, on ODP Designer. This is a skin button.

In Properties Explorer,

Set the Overlay Image property, to button _ offers image.

Set Sel Glyph Image, to can _ selector.

Set Border Style, to None.

Set Sel Border Style, to None.

With your mouse, right-click on the skin button, to show the pop up context menu. Select Copy. Press Control D to de select the selected skin button. Now, right-click on the Page to show the pop up context menu again, and select Paste. Repeat paste three more times until you get five skin buttons in total.

Select the second skin button, and set Overlay Image property to button _ downloads.

Select the third skin button, and set Overlay Image property to button _ vouchers.

Select the fourth skin button, and set Overlay Image property to button _ web site.

Select the fifth skin button, and set Overlay Image property to button _ PEPSI.

Task 8.14. ADDING SKIN IMAGES

A very important design pattern, for creating applications that will fit on every screen, is Skin Images. Skin Images, are ordinary images, that use guides in order to tile, and cover the entire element they apply to.

In Project Explorer, locate the button _ shade image, and open it.

To convert any image to skin image, click on the Skin Cutter button.

You will then notice four green lines around the image. Those are your skin slicing guides. You can move the guides by dragging the conjunction points like this.

In this skin, the four corner slices will be preserved, and painted on the target canvas exactly as they are sliced. The five remaining areas, will be tiled in order to cover the canvas dimensions.

To fully understand this, go back to the ODP Designer, in Home Page, and select the first Skin Button.

In Properties Explorer, set the Skin Image property to button _ shade, and observe the Visual Designer. Try changing the preview to 320 by 480 pixels; observe that button width is automatically expanding to cover the entire screen, and the skin image is nicely tiling along it, preserving the corner slices.

Set the button height to 100 pixels. Notice that your skin is tiling nicely along the x axis, but not along the y axis.

To fix this, go back to Pixel Editor, select the Skin Cutter tool, and re-arrange the slicing guides like this.

This time we try to define tiling areas with the same color and no gradients, in order to tile smoothly.

Switch back to the ODP Designer, and click on Validate Page Design, to refresh the Page. Now the skin tiles nicely along both axes of the button.

Let us reset the button height to auto.

At run time, we would like the red shade to be used for marking the currently selected button. Reset Skin Image property to empty, and set Sel Skin Image to button _ shade.

To rapidly apply those settings to the rest of the buttons, right-click on the first button, and select the Apply Control Style command. A pop up window will ask if you wish to apply the selected control Style to same controls of this Page. Click yes to apply.

Switch preview to 240 by 320 pixels. This is the most common screen size for most consumer class mobile phones.

By applying the skin to your buttons, the button height has automatically changed. Since your Home Page has no caption, you can set the Margin Top property to 40 pixels, in order to shift the buttons a bit higher and avoid button scrolling.

Task 8.15. ADDING HYPERLINKS

The corner stone of ODP Execution Flow, are Hyperlinks. Hyperlinks, either in line with text, or as h ref on a widget, can be used to navigate from one Page to another, or to execute a command.

Open the Home Page, select the Hot Offers button, and locate the h ref property in Properties Explorer. Click on the ellipsis to open the ODP Hyperlink Window.

The ODP Hyperlink Window consists of four tabs:

The ODP Pages tab, containing a list of all ODP Pages in your application.

The ODP Commands tab, containing a list of all available ODP commands that your application can execute. ODP commands depend on the referenced Frameworks and JSR libraries.

The Mobile Content tab, that is out of our current training scope.

The Engine States tab, containing the states of State Machines and Automata of your application, also out of the current training scope.

Click on the ODP Pages tab, select the Offers Page, and click ok.

Repeat this for the other buttons.

Set h ref property of Downloads button, to Downloads Page.

Set h ref property of Vouchers button, to Vouchers Page.

Set h ref property of Web Site button, to Web Site Page.

Set h ref property of PEPSI button, to PEPSI Page.

Task 8.16. ADDING SOUNDS

Import the sample sound files in mobileFX Studio, by dragging them on the title bar.

You can Play a sound by clicking it in Project Explorer. Right-click on a sound and select Sound Editor. Sound Editor is a tool for editing sound waves, like trimming, clipping, and deleting samples. However, the actual purpose of this tool, is to help you re-sample your sounds and transcode them to formats appropriate for mobile phones, such as A-M-R, O-G-G VORBIS, MP3, and MP4.

Open Home Page and select the first button.

Set ClickSound, to Click Sound, and SelectSound, to Select Sound.

Right-click on the button, and select Apply Control Style, to copy the sound settings to the other buttons in the Page.

Likewise you can set background sounds in your Pages, either directly to the Page, or at the master ODP Style.

Task 8.17. ADDING DATA CAPTURE FORMS

Open the Vouchers Page in ODP Designer.

From Widgets Explorer, double-click on the Edit Box to add it in the Page.

Select the control and go to Properties Explorer.

Set Control Name, to u d _ name.

Set Font Color, to White.

Set Font Size, to Small.

Set Sel Font Color, to White.

Set Sel Font Size, to Small.

Set Margin Top, to 5 pixels.

Set Margin Bottom, to 10 pixels.

Copy the Edit Box, and press Control D, to de select it. Paste it two more times.

Select the second Edit Box, and set its Control Name, to u d _ surname.

Select the third Edit Box, and set its Control Name, to u d _ age.

Select the first Edit Box, press Left, to place the editing cursor, just before the Edit Box, and press Enter, to insert an empty line. In the empty line, align text to the left, and type, Name. Select the text and make it bold.

Repeat this for the other two Edit Boxes, typing Surname and Age. Select the third Edit Box, and set the Input Style to Numeric.

Under the third Edit Box, type Submit, select the text, and click the Edit Hyperlink button. In the Hyperlink field, type a P SMS command. P SMS stands for Premium SMS, and should be used for sending SMS messages, to premium numbers.

A P SMS command has the following signature.

Do, colon, //, p sms, open curly bracket, MSISDN, comma, keyword, comma, comma separated list of submission fields, close curly bracket.

In ODP commands, you can use LOOKUPS. A LOOKUP substitutes a control name, with its persisted value. For example, by typing, left bracket, u d _ name, right bracket, this will substitute to the value of the u d _ name Edit Box, that at run time, this could be John.

After pressing Submit, the Please Wait Page will appear while the ODP tries to send the SMS message. If the message is successfully delivered, then the SMS OK Page will automatically show. If the SMS message fails to submit, the SMS Fail Page will appear.

On the server side, once the server receives the SMS message, should validate it, and produce a reply SMS message, with a members discount coupon. This action, is extremely easy to implement, with mobileFX SMS Gate Server software. You can download the evaluation version of SMS Gate, from mobileFX web site.

Task 8.18. ADDING BLUETOOTH DOWNLOADS

Open the Home Page in ODP Designer. Copy the Offers button, and then open the Downloads Page. Paste the button three times.

Select the first button and set Overlay Image, to button _ ringtone.

Set h ref to, do, colon, //, b t _ obex _ get, open curly bracket, ringtone.mp3, close curly bracket.

Select the second button and set Overlay Image, to button _ wallpaper.

Set h ref to, do, colon, //, b t _ obex _ get, open curly bracket, wallpaper.jpg, close curly bracket.

Select the third button and set Overlay Image, to button _ game.

Set h ref to, do, colon, //, b t _ obex _ get, open curly bracket, game.JAR, close curly bracket.

For the next steps, you will need a Blue Giga Bluetooth Hotspot, and mobileFX Blue Gate software. You can download the evaluation version of Blue Gate, and order a Blue Giga Bluetooth Hotspot, from mobileFX web site.

In mobileFX Blue Gate, you should add a new Campaign, import in its PULL folder the files you want to offer, and set the reply identifiers, to the content identifiers, of the b t _ obex get commands. As long as the content identifiers match a reply rule in Bluetooth Hotspots, consumers will be able to download free content, on demand.

Finally, set the Page Caption to FREE Downloads. to emphasize that added value content is free to download.

Task 8.19. ADDING A VIRAL MARKETING FORM

Open the PEPSI Page, and break its Visual Inheritance.

Right-click on the ODP node and select Copy Style. Right-click on the PEPSI Page, and select Paste Style.

In Theme Composer, make sure the drop down box shows PEPSI, and remove all the images.

Add a theme image and set its image property to PEPSI _ bk. Set the Scale Width and Scale Height properties to Fit Screen, and the Anchor property to Top Middle.

Select the PEPSI node in Project Explorer, and scroll down in Properties Explorer, to find the Background Color property. Click the Color Capture button, and move the mouse picker to the blue background of the image.

Set Caption, to INVITE YOUR FRIENDS – WIN A FREE PEPSI

Set Margin Top, to 150 pixels.

Set Caption Y, to 128 pixels.

Set Caption Font Color, to a light cyan color. You can use the Color Capture tool again and pick one from the image.

Set Caption Font Size, to Medium.

Set Caption Font Effects, to None.

Add two Edit Boxes with control names u d _ msg and u d _ number respectively.

Set Font Color, to White.

Set Font Size, to Small.

Set Sel Font Color, to White.

Set Sel Font Size, to Small.

Set Border Color, to a dark blue color.

Set Sel Border Color, to a light cyan color.

Set Margin Top, to 5 pixels.

Set Margin Bottom, to 10 pixels.

On the first Edit Box, set Text to, Visit K F C to take advantage of the new discounts.

On the second Edit Box, set Input Style, to Numeric.

Type Message above the first edit box, and Telephone Number above the second.

Add a Hyperlink with the text Invite, under the second Edit Box.

Set the Hyperlink to do, colon, //, sms, open curly bracket, add a lookup field for u d _ msg, comma, add a lookup field for u d _ number, comma, zero, close curly bracket.

Task 8.20. ADDING LOCALISATION TOKENS

In Project Explorer, click on the Localization node, to open Localization Manager. With mobileFX Studio, you can create multilingual applications using Localization Tokens.

Each Localization Token, is a variable that depending on the selected Locale, at run time, it translates to a pre defined text.

To collect all the Localization Tokens used in the Pages, the widgets, and the Frameworks of your application, right-click on Localization Manager, and select the Scan For Tokens command.

This will produce a list of all Localization Tokens in your application. Tokens missing translation, appear with red shade for your convenience.

You can sort tokens by name. or by translation, by clicking the header.

Note the u d _ tokens. Those tokens are the control names of the Edit Boxes, used in Voucher Page. Set those tokens to Age, Name, and Surname respectively.

MobileFX Studio, is using three letter abbreviation prefixes, to name tokens. Tokens starting with C M D, are used in command bars and hyperlinks. Tokens starting with E R R, are reserved for error messages. Tokens starting with TITLE, are used for Page captions. Tokens starting with CAPTION, are used for widget labels. Finally, tokens starting with M S G, are used for plain text messages in the Pages. Tokens are identifiers and can not have spaces. Also, all tokens must be in Latin characters.

To add a new Locale, right-click on Localization Manager, and select the Add Locale command. Each locale must have a name.

To define the default Locale, click on Project node in Project Explorer, and change the Default Language property.

To change a Locale at run time, you need to add a Settings Page. For demonstration purposes, open the About Page, and type the text English in a new line. Select it, and click the Edit Hyperlink button. In ODP commands, find the Settings Commands. MobileFX Studio, is automatically generating the appropriate command for cycling through available locales, starting from the default locale. Click OK to apply.

Also, to include a message token in a Page, simply type its name, in brackets. This can be used both, for single word tokens, or paragraphs.

To restore the About Page, click the Restore Original Page Design button, or simply delete the Hyperlink.

To Remove a Locale, select a token cell under the Locale you want to remove, right-click, and select Remove Locale. You must confirm by clicking Yes in the pop up window.

To remove unused tokens, select them, and press Del.

MobileFX Studio has support for right to left writing, supporting languages such as Arabic and Hebrew. Right to Left is automatically enabled when you switch Locale in Windows Language Bar.

MobileFX Studio is using Uni-code at design time, and U T F 8 at run time. However, to successfully run mobileFX Studio on Windows operating systems, you must install the software on an English version of Windows, and then add the languages you need. MobileFX Studio saves its files in XML.

Therefore, the XML run time on your computer, must be English, despite the fact that you can create multilingual applications.

Task 8.21. ADDING RUN TIME SUBSTITUTION VARIABLES

In every ODP Page, you can inject in your text Substitution Variables. A Substitution Variable is an identifier that starts and ends with the % symbol. At run time, substitution variables are automatically replaced with a value.

Some common substitution variables are for heap memory usage, R M S usage, and getting the serial number of the application. Every mobileFX application has a unique serial number that is generated the first time the application runs. The serial number remains the same, as long as you do not delete the application, despite changing your SIM card.

Task 8.22. RE-INDEX YOUR APPLICATION

Before the release compilation of the application, it is recommended to re-index the Pages. Click on the ODP menu, and select re-index ODP Pages command. Optionally you should re-index your application every time you import External or Dynamic Pages from other applications. You shouldn't re-index your applications once they are released, or add additional dynamic Pages.

Task 8.23. EXECUTION FLOW DIAGRAM

In ODP Page Designer, click on Flow Design, to observe the navigation flow of your ODP, starting from the Title Page, showing the Home Page, with links to the other ODP Pages.

Chapter 9. COMPILE AND RUN

To compile the application you created so far, select Release Mode, from the drop down box on toolbar, and click on the Full Project Compilation button. mobileFX Studio Compiler, will compile the Project's resources and Java files into a single JAR file per Variant.

Once compilation is complete, click Run in console button, to run the application in the selected Mobile Phone Emulator.