

# Tutorial: BlackBerry Object API Application Development Sybase Unwired Platform 2.1 ESD #3

#### DOCUMENT ID: DC01214-01-0213-01

LAST REVISED: May 2012

Copyright © 2012 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at *http://www.sybase.com/detail?id=1011207*. Sybase and the marks listed are trademarks of Sybase, Inc. <sup>®</sup> indicates registration in the United States of America.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

# Contents

Sybase Unwired Platform Tutorials	1
Getting Started with Unwired Platform	3
Installing Sybase Unwired Platform	3
Starting Sybase Unwired Platform Services	3
Starting Sybase Unwired WorkSpace	4
Connecting to Sybase Control Center	4
Learning Unwired WorkSpace Basics	5
Developing a BlackBerry Application	9
Installing the BlackBerry Java Plug-in for Eclipse	9
Setting System and Path Variables	9
Generating Java Object API Code	10
Creating the BlackBerry Project	12
Configuring BlackBerry Application Properties	14
Copying Unwired Platform Files to the Sample	
Project	14
Creating the User Interface	15
Creating a Launch Configuration for the Project	18
Testing the Device Application on the BlackBerry	
Simulator	20
Learn More About Sybase Unwired Platform	25
Index	.27

#### Contents

# **Sybase Unwired Platform Tutorials**

The Sybase<sup>®</sup> Unwired Platform tutorials demonstrate how to develop, deploy, and test mobile business objects, device applications, and mobile workflow packages. You can also use the tutorials to demonstrate system functionality and train users.

**Tip:** If you want to see the final outcome of a tutorial without performing the steps, the associated example project is available on SAP® Community Network: *http://scn.sap.com/docs/DOC-8803*.

- Learn mobile business object (MBO) basics, and use this tutorial as a foundation for the Object API application development tutorials:
  - Tutorial: Mobile Business Object Development
- Create native Object API mobile device applications:
  - Tutorial: Android Object API Application Development
  - Tutorial: BlackBerry Object API Application Development
  - Tutorial: iOS Object API Application Development
  - Tutorial: Windows Mobile Object API Application Development
- Create a mobile business object, then develop a mobile workflow package that uses it:
  - Tutorial: Mobile Workflow Package Development

Sybase Unwired Platform Tutorials

# **Getting Started with Unwired Platform**

Install and learn about Sybase Unwired Platform and its associated components.

Complete the following tasks for all tutorials, but you need to perform them only once.

## **Installing Sybase Unwired Platform**

Install Sybase Mobile SDK and Sybase Unwired Platform Runtime.

Before starting this tutorial, install all the requisite Unwired Platform components. See the Sybase Unwired Platform documentation at *http://sybooks.sybase.com/sybooks/sybooks/sybooks.xhtml*.

- Release Bulletin for Sybase Mobile SDK
- Installation Guide for Sybase Mobile SDK
- Release Bulletin for Runtime
- Installation Guide for Runtime
- 1. Install these Unwired Platform Runtime components:
  - Data Tier (included with single-server installation)
  - Unwired Server
- 2. Install Mobile SDK, which includes:
  - Development support for native Object API applications, HTML5/JS Hybrid (Mobile Workflow) applications, and native OData SDK applications.
  - Sybase Unwired WorkSpace, the Eclipse-based development environment for MBOs and mobile workflows.

## **Starting Sybase Unwired Platform Services**

Start Unwired Server, Sybase Control Center, the sample database, the cache database (CDB), and other essential services.

The way in which you start Unwired Platform services depends on the options you selected during installation. You may need to manually start Unwired Platform services. Select **Start > Programs > Sybase > Unwired Platform > Start Unwired Platform Services**.

The Unwired Server services enable you to access the Unwired Platform runtime components and resources.

## Starting Sybase Unwired WorkSpace

Start the development environment, where you can create mobile business objects (MBOs), manage EIS data sources and Unwired Server connections, develop Mobile Workflow applications, and generate Object API code.

Select **Start > Programs > Sybase > Unwired Platform > Unwired WorkSpace**. The Sybase Unwired WorkSpace opens in the Mobile Development perspective. The Welcome page displays links to the product and information.

#### Next

To read more about Unwired WorkSpace concepts and tasks, select Help > Help Contents.

## **Connecting to Sybase Control Center**

Open the Sybase Control Center Administration Console to manage Unwired Server and its components.

From Sybase Control Center, you can:

- View servers and their status
- Start and stop a server
- View server logs
- Deploy a mobile application package
- Register application connections
- Set role mappings

For information on configuring, managing, and monitoring Unwired Server, click **Help** > **Online Documentation**.

1. Select Start > Programs > Sybase > Sybase Control Center.

**Note:** If the Sybase Control Center does not launch, make sure that the Sybase Control Center service is started in the Windows Services dialog.

2. Log in by entering the credentials set during installation.

Sybase Control Center gives you access to the Unwired Platform administration features that you are authorized to use.

## Learning Unwired WorkSpace Basics

Sybase Unwired WorkSpace features are well integrated in the Eclipse IDE. If you are unfamiliar with Eclipse, you can quickly learn the basic layout of Unwired WorkSpace and the location of online help.

- To access the online help, select **Help > Help Contents**. Some documents are for Sybase Unwired Platform, while others are for the Eclipse development environment.
- The Welcome page provides links to useful information to get you started.
  - Reopen the Welcome page by selecting **Help > Welcome**.
  - To close the Welcome page, click **X**.
  - To learn about tasks you must perform, select the **Development Process** icon.
- In Unwired WorkSpace, look at the area (window or view) that you will use to access, create, define, and update mobile business objects (MBOs).

Window	Description
WorkSpace Navigator view	Use this view to create Mobile Application projects, and review and modify MBO-related properties.
	This view displays mobile application project fold- ers, each of which contains all project-related re- sources in subfolders, including MBOs, datasource references to which the MBOs are bound, personal- ization keys, and so on.
Enterprise Explorer view	A view that provides functionality to connect to var- ious enterprise information systems (EIS), such as database servers, SAP <sup>®</sup> back ends, and Unwired Server.

Window	Description
Mobile Application Diagram	The Mobile Application Diagram is a graphical ed- itor where you create and define mobile business objects.
	<ul> <li>Use the Mobile Application Diagram to create MBOs (including attributes and operations), then define relationships with other MBOs. You can:</li> <li>Create MBOs in the Mobile Application Dia- gram using Palette icons and menu selections – either bind or defer binding to a datasource, when creating an MBO. For example, you may want to model your MBOs before creating the datasources to which they bind. This MBO de- velopment method is sometimes referred to as the top-down approach.</li> <li>Drag and drop items from Enterprise Explorer to the Mobile Application Diagram to create the MBO – quickly creates the operations and at- tributes automatically based on the datasource artifact being dropped on the Mobile Applica- tion Diagram.</li> </ul>
	Each new mobile application project generates an associated mobile application diagram.
Palette	The Palette is accessed from the Mobile Application Diagram and provides controls, such as the ability to create MBOs, add attributes and operations, and de- fine relationships, by dragging and dropping the corresponding icon onto the Mobile Application Di- agram or existing MBO.
Properties view	Select an object in the Mobile Application Diagram to display and edit its properties in the Properties view. While you cannot create an MBO from the Properties view, most development and configura- tion is performed here.
Outline view	Displays an outline of the active file and lists struc- tural elements. The contents are editor-specific.

Window	Description
Problems view	Displays validation errors or warnings that you may encounter in addition to errors in the Diagram editor and Properties view. Follow warning and error mes- sages to adjust MBO properties and configurations to avoid problems, and use as a valuable source for collecting troubleshooting information when report- ing issues to Customer Service and Support.
Error Log view	Displays error log information. This is a valuable source for collecting troubleshooting information.

Getting Started with Unwired Platform

# **Developing a BlackBerry Application**

Generate code for the BlackBerry platform, develop a BlackBerry device application with that code and sample files, and test the application's functionality on a simulator.

#### Prerequisites

- Install Sybase Unwired Platform Mobile SDK and Runtime as indicated in *Getting Started with Unwired Platform* on page 3.
- Create the mobile business objects (MBOs) that you deploy to Unwired Server using one of these methods:
  - Complete *Tutorial: Mobile Business Object Development*, which provides the foundation tasks for this tutorial.
  - Download and import the completed example project if you want to bypass performing the MBO tutorial. The associated example project is available on SAP® SDN: *http://scn.sap.com/docs/DOC-8803*
- Review the Supported Hardware and Software requirements for this release.

#### Task

Create a device application that communicates with the database mobile business objects that are deployed to Unwired Server.

## Installing the BlackBerry Java Plug-in for Eclipse

The BlackBerry Java Plug-in for Eclipse enables you to finish developing the BlackBerry application using BlackBerry smartphone-specific development, debugging, and simulation tools.

- 1. Download the BlackBerry Java Plug-in for Eclipse (full installer) from *http://us.blackberry.com/developers/javaappdev/*.
- 2. Run the downloaded installer and follow the instructions to install the BlackBerry Java Plug-in for Eclipse. Accept the default installation folder: do not specify the Unwired WorkSpace Eclipse directory.

## **Setting System and Path Variables**

Set the Java system variable and add Java to the system path.

**1.** Set the JAVA\_HOME system variable:

- a) Right-click the computer icon for the system running the BlackBerry Java Plug-in, and select **Properties**.
- b) In System Properties, select Advanced, then click Environment Variables.
- c) In System variables, select JAVA\_HOME and click Edit.
- d) In the Edit System Variable dialog, add a JDK directory path.

Check the JDK system requirements for the BlackBerry environment you are using: *http://us.blackberry.com/developers/javaappdev/*.

If you have Sybase installed, you can use the existing JDK to set the JAVA\_HOME variable: JAVA\_HOME=C:\Sybase\UnwiredPlatform\JDK1.6.0\_31.

Note: For the Sybase 64-bit environment, you would still use the 32-bit JDK.

- 2. Add %JAVA\_HOME%\bin to the system path:
  - a) In System variables, select Path and click Edit.
  - b) In the Edit System Variable dialog, add a JDK directory path.

If you have Sybase installed, you can use the existing JDK to set the variable: C: \Sybase\UnwiredPlatform\JDK1.6.0\_31\bin.

3. Restart the system for the environment variables to take effect.

## **Generating Java Object API Code**

Use the Generate Code wizard to generate object API code for the SUP101 mobile application project. The code generation creates the business logic, attributes, and operations for the mobile business objects in the project.

#### Prerequisites

In Enterprise Explorer, you must be connected to both My Sample Database and My Unwired Server. Code generation fails if the server-side (runtime) enterprise information system (EIS) datasources referenced by the MBOs in the project are not running and available to connect to when you generate object API code.

#### Task

1. In Unwired WorkSpace, open the SUP101 mobile application project.

In WorkSpace Navigator, right-click the **SUP101** folder and select **Open in Diagram Editor**.

2. (Optional) If you are performing other tutorials, add a new folder to the project to organize the generated code for each device platform. For example, in WorkSpace Navigator, expand SUP101 and under Generated Code add a RIM folder. The Generated Code directory was created during the MBO tutorial.

- **3.** Right-click anywhere in the SUP101 Mobile Application Diagram and select **Generate** Code.
- **4.** In the Generate Code wizard, click **Next** to select the default configuration and continue without any changes.
- 5. In the Select Mobile Business Objects page, select the Customer MBO, then click Next.
- 6. In the Configure Options page, specify these values and click Finish.

Option	Description	
Language	Java	
Platform	Java ME for BlackBerry	
Unwired server	My Unwired Server	
Server domain	Accept default value.	
Page size	Accept default value of 1024.	
Package	(Optional) A unique name for the Java package.	
Project path	Leave the default, \SUP101\Generated Code, or browse to another folder you created for the device platform in step 2.	
Third-party jar file	You do not need to choose one for this tutorial. You see a warning at the top of the page: The third-party class 'net.rim.device.api.sys- tem.ApplicationDescriptor' cannot be found. You can ignore this warning; it does not affect code generation.	
Generate JavaDoc	Unselect.	

You can ignore the error icons that appear in WorkSpace Navigator under \Generated Code\src\SUP101, and the error in the Problems view: net cannot be resolved.

The error occurs because the package Java file called by SUP101DB. jar is not yet in the project build path. The errors resolve later, when you build the application in the BlackBerry Java Plug-in for Eclipse.

 In the Success dialog, click OK. In the Generated Code directory, you see a \src\SUP101 folder.

## **Creating the BlackBerry Project**

Create a new BlackBerry SUP101Sample project in the BlackBerry Java Plug-in for Eclipse. Add library resources to the project.

#### Prerequisites

Install the BlackBerry Java Plug-in for Eclipse. Review the *Supported Hardware and Software* requirements to ensure you have the correct version.

#### Task

1. Start the BlackBerry Java Plug-in for Eclipse.

**Note:** When the Eclipse Workspace Launcher prompts you to select a workspace, do not choose one that you are using with Unwired WorkSpace. If necessary, select or create a new workspace.

- 2. Depending on the BlackBerry Plugin version:
  - Select File > New > BlackBerry Project, or
  - Select File > New > BlackBerry > BlackBerry Project.
- 3. In the New BlackBerry Project wizard, use these values and click Next.
  - Project name enter SUP101Sample.
  - Use a project specific JRE select the BlackBerry OS version that was automatically installed with the plugin.

🏶 New BlackBerry Project 📃 🗖 🔀
Create a BlackBerry Project
Create a BlackBerry project in the workspace or in an external location.
Project name: SUP101Sample
Contents
<ul> <li>Create new project in workspace</li> </ul>
Create project from existing source
Directory: C:\Eclipse\workspaces\BB_cust_tutorial\SUP101Sample Browse
JRE
Use a project specific JRE:     BlackBerry JRE 6.0.0
O Use default JRE (currently 'jre6') Configure JREs
Working sets
Add project to working sets
Working sets: Select
The default compiler compliance level for the current workspace is 1.6. The new project will use a project specific compiler compliance level of 1.3.
(?) < Back Next > Finish Cancel

- 4. In the Java Settings page, modify the build path to point to the correct location for the sup\_client2.jar and UltraLiteJ12.jar files for the project:
  - a) Click the **Libraries** tab.
  - b) Click Add External JARs.
  - c) Browse to <UnwiredPlatform\_InstallDir>\UnwiredPlatform \MobileSDK213\ObjectAPI\BB.
  - d) Select the two JAR files, then click **Open**.
  - e) Click Finish.

## **Configuring BlackBerry Application Properties**

Modify the BlackBerry application description to define the general BlackBerry properties used in an application.

- 1. In Package Explorer, expand the SUP101Sample project.
- 2. Open the BlackBerry\_App\_Descriptor.xml file.
- 3. Select the Application tab.
- 4. In the Title, enter SUP101Sample.
- **5.** (Optional) Enter a version, vendor, and description. You can review the other options in the Application Descriptor window, such as general attributes, resources, and build settings of the project. For this tutorial, leave all other settings unchanged.

General Information This section describes the application.	e general BlackBerry properties used in this	Locale Resources This section describes the resources used in this application. * Description is shown on Desktop Manager only.
Title: Version: Vendor: Description: Application type: Name of main MIDlet class	SUP101Sample 0.0.1 Vendor BlackBerry Application	Internationalized resource bundle available Resource bundle: Title ID: Description ID:
Application argument: Home screen position: Auto-run on startup Startup tier:	0 7 Violation icon on the BlackBerry home screen	Application Icons Specify icons to be used by this application. Rollover Icon File Add Add External Remove

6. Click File > Save.

## **Copying Unwired Platform Files to the Sample Project**

Copy Unwired Platform BlackBerry runtime files and generated object API code to the SUP101Sample project. You had created the code previously in the Generate Code wizard.

- 1. In Windows Explorer, browse to <UnwiredPlatform\_InstallDir> \UnwiredPlatform\MobileSDK213\ObjectAPI\BB.
- Copy these files: CommonClientLib.cod, MessagingClientLib.cod, MocaClientLib.cod, sup\_client2.cod, and UltraLiteJ12.cod.

- 3. Paste the copied COD files into the BlackBerry Java Plug-in simulator directory, located in <*BlackBerry\_Eclipse\_InstallDir>*\plugins \net.rim.ejde.componentpack
- 4. In Windows Explorer, go to the workspace folder you set for Sybase Unwired WorkSpace, and copy the generated code files. For example, go to C:\Documents and Settings\user1\SUP-wksp \SUP101\Generated Code\src and copy the SUP101 folder.
- 5. In the BlackBerry Application Development workspace, go to the SUP101Sample project and paste the folder into the src directory. For example, go to C:\Documents and Settings\user1\BB-wksp \SUP101Sample\src and paste the SUP101 folder. In Package Explorer, under the SUP101Sample/src folder, you see the SUP101 and SUP101.intrnl folders.

If the copied folders do not appear, click **Refresh** or press **F5**.

## **Creating the User Interface**

Add sample code from the ZIP archive to the SUP101Sample application. The files provide the functionality and layout of the user interface.

#### Prerequisites

Obtain text files from the SUP\_BB\_Custom\_Dev\_Tutorial\_code.zip file. The text files contain code snippets that you need to build the user interface. You will create these classes for the interface: CustomerSample, CustomerSampleScreen, CustomerList, CustomerDBCallback, and ObjectAPIUtil.

- If you are viewing this guide online from the Sybase Product Documention Web site, click *SUP\_BB\_Custom\_Dev\_Tutorial\_code.zip* to access the ZIP archive containing the text files.
- If you are viewing this guide as a PDF, go to the Sybase Product Documentation Web site at *http://sybooks.sybase.com/sybooks/sybooks.xhtml*. Select **Sybase Unwired Platform**. Select the appropriate version, navigate to this topic in the tutorial, then click the link for the ZIP file to save the files locally.

#### Task

- 1. In Package Explorer, right-click SUP101Sample, and select New > Package.
- 2. In the New Java Package wizard, use these values and click Finish.
  - Source folder verify SUP101Sample/src appears.
  - Name-enter com.sybase.sup.samples.objectapi.

- 3. In Package Explorer, you will create one new Java class for the application in the com.sybase.sup.samples.objectapi package, and then copy the remaining four classes in the ZIP file into the package. Here is the list of files in the ZIP:
  - CustomerDBCallback.java implements the CallbackHandler to demonstrate how to track changed entities in the onSynchronize callback method.
  - CustomerList.java populates the customer list.
  - CustomerSample.java creates the main customer application.
  - CustomerSampleScreen.java creates the customer screen.
  - ObjectAPIUtil.java-governs how the application initializes and synchronizes data in the background.
  - a) Right-click the **com.sybase.sup.samples.objectapi** package and select **New** > **Class**.
  - b) In the Java Class page, for the name, enter one of the Java class names, for example: CustomerSample, and click **Finish**.

📸 New Java Class			_ 🗆 🗙
Java Class Create a new Java	class.		C
Source folder:	SUP101Sample/src		Browse
Package:	com.sybase.sup.samples.objectapi		Browse
Enclosing type:			Browse
Name: Modifiers:	CustomerSample © public O default O private □ abstract □ final □ static	C protected	
Superclass:	java.lang.Object		Browse
Interfaces:			Add Remove
Which method stubs	would you like to create?  public static void main(String[] args)  Constructors from superclass  Inherited abstract methods		
Do you want to add	comments? (Configure templates and default v	alue <u>here</u> )	
?	[	Finish	Cancel

In the SUP101Sample, under the src

\com.sybase.sup.samples.objectapi package, you see the new Java
class: CustomerSample.java.

- c) Browse to the directory where you saved the ZIP file.
- d) Copy the content of the Java text file with the same name: CustomerSample.java.
- e) In Package Explorer, open the new Java class with the same name: CustomerSample.java.
- f) Paste the copied text code into the new Java class: CustomerSample.java.Copy over any existing content.

```
- -
SUP101Sample
               🕖 CustomerSample.java 🔀
   package com.sybase.sup.samples.objectapi;
                                                                  A 🗖
  * Created on Jul 20, 2010.
😼 🖲 import java.util.Vector;[]
  € /**
    * @author bdeng
    * Customer sample main application.
    * Note: For the sake of simplicity, this sample application m
    * resource bundles and resource strings. However, it is STRC
    * that application developers make use of the localization fe
    * within the BlackBerry development platform to ensure a seam
    * experience across a variety of languages and geographies.
    * on localizing your application, please refer to the BlackBe
    * Environment Development Guide.
    */
   public final class CustomerSample extends UiApplication
       private KeywordFilterField keywordFilterField;
                                  customerList;
       private CustomerList
       private Vector
                                   customers;
    •
```

g) Select File > Save.

The package contains errors until you create the other classes. Ignore these errors and continue.

- h) For the remaining four classes, copy the Java files from the ZIP archive to the SUP101Sample\src\com.sybase.sup.samples.objectapi package. The compilation errors in the Java files resolve when you finish creating the classes.
- 4. Modify the host name or IP address in the ObjectAPIUtil. java file to point to the Sybase Unwired Server.
  - a) In Package Explorer, expand the SUP101Sample project.

- b) Under the \src\com.sybase.sup.samples.objectapi package, doubleclick the ObjectAPIUtil.java file.
- c) Modify the host name or IP address, and **Save**.

## **Creating a Launch Configuration for the Project**

Create and define a new launch configuration for the SUP101Sample project. The configuration defines how the application launches and the target BlackBerry platform.

- 1. In Package Explorer, right-click the SUP101Sample project, and select Run As > Run Configurations.
- 2. Select BlackBerry Simulator and the New launch configuration icon.



- 3. In the Projects tab, select the SUP101Sample project.
- 4. For the name, enter SUP101Sample.
- 5. In the JRE tab, select the Project JRE (BlackBerry JRE Version).
- 6. In the Simulator tab, select Launch Mobile Data System Connection Service (MDS-CS) with simulator. Then select the BlackBerry-SDK *Version* as the device.

Name: SUP101Sample
😰 Projects 🔜 JRE 🔲 Simulator 🧤 Source 🔲 Common
General Debugging Memory Network Ports View Advanced
Launch Mobile Data System Connection Service (MDS-CS) with simulator
Launch app or URL on startup:
Device:
BlackBerry-SDK - 9800
Automatically use default values for all prompts
Specify the number of seconds to wait before the automated response is selected:
0
PIN:
0x2100000A
ESN:
MEID:
Enable device security
System Locale:
Keypad Locale:
Apply Revert

If you are running the BlackBerry JDE 7.0 with the BlackBerry Java Plug-in v1.5, you need to launch the Mobile Data System (MDS) manually for the BlackBerry simulator to run.

MDS is a framework that offers security, wireless connectivity, manageability options, and development methods. You need it to run the simulator. It is a known issue for BlackBerry that the simulator cannot launch MDS. This has been fixed in BlackBerry JDE 7.1.

- a) From a command prompt, go to <*Eclipse\_InstallDir*>\plugins \net.rim.ejde.componentpack7.0.0\_7.0.0.33\components \MDS.
- b) Run MDS manually, enter: start run.bat.
- c) When the following message appears, ignore it and press any key to continue.

ERROR: Your java version does not meet the minimum requirement of 1.6.0\_20. Press any key to continue... MDS starts and you can launch the BlackBerry simulator. Leave the command window open.

- 7. In the **Network** tab, verify that **Disable Registration** is checked, or the sample application cannot get data from Unwired Server.
- 8. Click Apply, then Close.

## Testing the Device Application on the BlackBerry Simulator

Run the SUP101Sample application on the BlackBerry simulator, and change customer information to update the interface.

Depending on the simulator you are using, the screens may vary.

1. In Package Explorer, right-click the SUP101Sample project and select Run As > BlackBerry Simulator.

It may take several minutes for the BlackBerry simulator's Setup screen to appear. If this is the first time running the simulator, cancel the Setup screen.



**2.** On the main window, click **All** to access the applications screen, then scroll until you see the SUP101Sample application.



3. Click **SUP101Sample** to launch the application.

The application registers and synchronizes data from the server in the background.

In the initialization process, the system enables the operation to target change notifications using:

```
SynchronizationGroup
sg=SUP101DB.getSynchronizationGroup("default");
sg.setEnableSIS(true);
```

```
sg.save();
```

After the application initializes, the device application shows the SUP101Sample application with a list of customer data. You can scroll through the customer list to see more data, search, and make changes. The data loads from the database on demand.

When the application queries the customer list, it uses a named query findAll. If the customer list is long, it uses a SUP101DB.executeQuery() API to get only columns that are needed, such as (fname, lname...), instead of the entire customer object; this results in better performance.



- 4. Focus on the customer list and in the Search bar, enter dal. The customer list is filtered and only customers with a first or last name beginning with "dal" are shown, in this case, Daljit Sinnot.
- **5.** Select the customer, **Daljit Sinnot**, and click the trackball. The Update Customer screen for Daljit Sinnot appears.



6. Change the first name of the customer to abc and click submit.

The **submit** button maps to the update operation of the customer mobile business object. The synchronize operation uses SUP101DB.beginSynchronize and occurs in the background so the user interface is not affected. When the application synchronizes, any pending operations are uploaded to Unwired Server.

Any changes in the back end initiate notifications from the server. The device application uses a ChangeLog API, specifically ObjectList changeLogs = SUP101DB.getChangeLogs(query);, to query those managed items and use them to update the user interface if needed.

The customer list appears with the name change you made.

7. Close the simulator to stop the SUP101Sample application.

## Learn More About Sybase Unwired Platform

Once you have finished, try some of the other samples or tutorials, or refer to other development documents in the Sybase Unwired Platform documentation set.

Check the Sybase Product Documentation Web site regularly for updates: *http://sybooks.sybase.com/sybooks/sybooks.xhtml*, then navigate to the most current version.

#### Tutorials

Try out some of the other getting started tutorials available on the Product Documentation Web site to get a broad view of the development tools available to you.

#### Example Projects

An example project is the end results of a finished tutorial without going through the steps. Download example projects from the SAP<sup>®</sup> Community Network (SCN) at *http://scn.sap.com/docs/DOC-8803*.

#### Samples

Sample applications are fully developed, working applications that demonstrate the features and capabilities of Sybase Unwired Platform.

Check the SAP<sup>®</sup> Development Network (SDN) Web site regularly for new and updated samples: *https://cw.sdn.sap.com/cw/groups/sup-apps*.

#### Online Help

See the online help that is installed with the product, or available from the Product Documentation Web site.

#### Developer Guides

Learn best practices for architecting and building device applications:

- *Mobile Data Models: Using Data Orchestration Engine* provides information about using Sybase Unwired Platform features to create DOE-based applications.
- *Mobile Data Models: Using Mobile Business Objects* provides information about developing mobile business objects (MBOs) to fully maximize their potential.

Use the appropriate API to create device applications:

- Developer Guide: Android Object API Applications
- Developer Guide: BlackBerry Object API Applications
- Developer Guide: iOS Object API Applications
- Developer Guide: Windows and Windows Mobile Object API Applications
- Developer Guide: Mobile Workflow Packages

Customize and automate:

• *Developer Guide: Unwired Server Management API* – customize and automate system administration features.

Javadoc and HeaderDoc are also available in the installation directory.

# Index

application properties 14

## В

BlackBerry application description 14 BlackBerry Java Plug-in for Eclipse installing 9 BlackBerry project, creating 12 BlackBerry Simulator 20 build path 12

## С

cod files CommonClientLib.cod 14 MessagingClientLib.cod 14 MocaClientLib.cod 14 sup-client2.cod 14 UltraLiteJ12.cod 14 code generation 10 CommonClientLib.cod 14 CustomerDBCallback.java 15 CustomerList.java 15 CustomerSample.java 15

## D

descriptor file 12

## Ε

example projects 1

## J

```
JAR files
adding 12
sup-client2.jar 12
UltraLiteJ12.jar 12
Java
path 9
```

perspective 15 system variable 9 Java class, creating CustomerDBCallback.java 15 CustomerList.java 15 CustomerSample.java 15 CustomerSampleScreen.java 15 ObjectAPIUtil.java 15 Java Object API code, generating 10

## L

launch configuration 18

## Μ

MessagingClientLib.cod 14 mobile business object tutorial 1 Mobile Workflow package tutorial 1 MocaClientLib.cod 14

## 0

Object API tutorials 1 ObjectAPIUtil.java 15

## Ρ

project build path 12

## S

samples downloading 25 sup-client2.cod 14 sup-client2.jar 12 SUP101Sample application running 20 testing 20 updating data 20 yiewing data 20 Sybase Control Center, connecting to 4 Sybase Mobile SDK installing 3

#### Index

Sybase Unwired Platform documentation resources 25 getting started 3 installing 3 Sybase Unwired WorkSpace basics 5 how to access online help 5 starting 4

## Т

testing 20 troubleshooting information 5 tutorials 1 downloading 25

## U

UltraLiteJ12.cod 14 UltraLiteJ12.jar 12 Unwired Platform Runtime installing 3 Unwired Platform services 3 Unwired WorkSpace basics 5 user interface creating 15