

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

DOCUMENT ID: DC01734-01-0213-01

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. [®] 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 an Android Application	9
Installing the Android SDK	9
Installing ADT in Unwired WorkSpace	10
Generating Java Object API Code	11
Creating the Android Project	13
Copying Unwired Platform Files to Sample	
Project	17
Configuring Android Application Properties	18
Creating the User Interface	21
Creating a Launch Configuration for the Project	22
Testing the Device Application on the Android	
Emulator	25
Learn More About Sybase Unwired Platform	33
Index	35

Contents

Sybase Unwired Platform Tutorials

The Sybase[®] 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 [®] 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.
	 Use the Mobile Application Diagram to create MBOs (including attributes and operations), then define relationships with other MBOs. You can: 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. 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.
	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 an Android Application

Generate code for the Android platform, develop an Android device application with that code and sample files, and test the application's functionality on an emulator.

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*
- Download the supported versions of the Android SDK and Android Development Tools (ADT).

See the *Supported Hardware and Software* guide for links to the supported versions on Product Documentation at *http://sybooks.sybase.com/sybooks/sybooks.xhtml*. Navigate to the appropriate version of Sybase Unwired Platform.

Task

Create an Android native application that communicates with the database mobile business objects that are deployed to Unwired Server. This tutorial was created using Android SDK r18 and ADT Plugin for Eclipse 18.0.0 on an Android 2.2 emulator. If you use a different version, some steps may vary.

Installing the Android SDK

Install the Android SDK.

- 1. Confirm your system meets the requirements at *http://developer.android.com/sdk/ requirements.html*.
- 2. Download and install the supported version of the SDK starter package.

See the *Supported Hardware and Software* guide for links to the supported versions on Product Documentation at *http://sybooks.sybase.com/sybooks/sybooks.xhtml*. Navigate to the appropriate version of Sybase Unwired Platform.

- **3.** Launch the **Android SDK Manager** and install the Android SDK tools, Platform tools, and Android API (compatibility package for the device version).
- **4.** Launch the **Android Virtual Device Manager**, and create an Android virtual device to use as your emulator.

Installing ADT in Unwired WorkSpace

Install the supported version of Android Development Tools (ADT) in the Sybase Unwired WorkSpace Eclipse environment.

See the *Supported Hardware and Software* guide for links to the supported versions on Product Documentation at *http://sybooks.sybase.com/sybooks/sybooks.xhtml*. Navigate to the appropriate version of Sybase Unwired Platform.

- 1. Start Unwired WorkSpace, then select Help > Install New Software.
- 2. In the Available Software window, click Add.
- 3. In the Add Repository dialog, enter ADT Plugin for the name, and https://dl-ssl.google.com/android/eclipse/ for the location.
- 4. In the Available Software dialog, select Developer Tools, then click Next.
- 5. In the Install Details window, a list of downloadable tools, click Next.
- 6. Accept the license agreements, then click **Finish**.

Note: If you get a security warning about the authenticity or validity of the software, click **OK**.

- When the installation completes, restart Unwired WorkSpace. If you installed ADT for the first time, you see the Welcome to Android Development window.
- 8. (First-time installations) In Welcome to Android Development, select Use existing SDKs, then browse to where the Android SDK is installed, by default, C:\Program Files \Android\android-sdk.

Click Next.

9. Click Finish.

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.
- In WorkSpace Navigator, verify the Java Compiler level is set correctly:
 - 1. Select Window > Preferences > Java > Compiler.
 - 2. In the Compiler compliance level list, select 1.6 if it does not already appear.
 - 3. Click Apply, then OK.

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 an Android 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 continue without a configuration.
- 5. In the Select Mobile Business Objects window, select the **Customer** MBO, then click Next.
- 6. In the Configure options window, specify these values and click Finish.

Option	Description
Language	Select Java.
Platform	Select Android.
Unwired server	Select My Unwired Server.

Option	Description
Server domain	Select default .
Page size	Select 1024.
Package	(Optional) Enter 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	Click Browse to open an android.jar,by default located in C:\Program Files \Android\android-sdk\plat- forms\android-xx.
Generate JavaDoc	Unselect for this tutorial.

🛟 Generate Code	:				_ 🗆 🗵
Configure option Configure options I	ns for code generation				
Code generation Language: Platform: Unwired server: Server domain: Page size: Package: Destination Project pat C File system Clean up de Third-party jar fi Generate met Generate met	options Java Java Android My Unwired Server default 1024 f SUP101\Generated Co path: Style=Stination before code generation le: C:\Program Files\Android\at adata classes g object manager classes aDoc	de n ndroid-sdk\platforr	ms\android-8\ar	ndroid.jar	Image: state
?		< Back	Next >	Finish	Cancel

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

Creating the Android Project

Create a new Android SUP101Sample project in Unwired WorkSpace. Add library resources and set other application properties.

Prerequisites

Obtain text files from the SUP_Android_Custom_Dev_Tutorial_code.zip file to help create the project and, in a subsequent topic, build the user interface.

- If you are viewing this guide online from the Sybase Product Documention Web site, click *SUP_Android_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*. Click the link for the appropriate Sybase Unwired Platform version. Navigate to this topic in the tutorial, and click the link for the ZIP file to access the text files.

Task

- 1. Start Unwired WorkSpace.
- 2. In Unwired WorkSpace Preferences, set the Android SDK location.
- 3. Select File > New > Project.
- Select Android > Android Project, then Next. Depending on the Android version you are using, the information you provide in the next several steps may be in one or two screens.
- 5. In the New Android Project wizard, use these values and click Next.
 - Project Name enter SUP101Sample.
 - Select Create new project in workspace.
 - Change Use default location if it does not display the appropriate workspace location.

🖶 New Android Project
Create Android Project Image: Constraint of the project Select project name and type of project Image: Constraint of the project
Project Name: SUP101Sample
 Create new project in workspace
C Create project from existing source
C Create project from existing sample
☑ Use default location
Location: C:/Documents and Settings/user/workspac Browse
Working sets
Add project to working sets
Warking sets: Select
Cancel

6. In the Select Build Target window, click an Android version 2.2 or later and Next.

📸 New Android Project			_ 🗆 🗵
Select Build Target Choose an SDK to target			0
Build Target			
Target Name	Vendor	Platform	API
Android 2.2 Google APIs DTS Add-On Real3D Add-On GALAXY Tab Addon	Android Open Source Project Google Inc. KYOCERA Corporation LGE Samsung Electronics Co., Ltd.	2.2 2.2 2.2 2.2 2.2 2.2	8 8 8 8
? 	< Back Next >	Finish	Cancel

- 7. In the Application Info window, verify or enter the information and click Finish.
 - Application Name SUP101Sample
 - Package Name enter com.sybase.sup.samples.objectapi
 - Create Activity SUP101SampleActivity

In the left pane, you should see the Package Explorer with the SUP101Sample project listed. In the src folder a default Sample Activity class is automatically generated for the project.

Tip: To correct a mispelled package name, right-click the package and select **Refactor** > **Rename** to change the name and update all references.

- 8. Add a compiler resource and library resources to the root directory of the Android project:
 - a) In Package Explorer, select **SUP101Sample** and add a libs folder.
 - b) In Windows Explorer, browse to C:\Sybase\UnwiredPlatform \MobileSDK213\ObjectAPI\Android.
 - c) Copy the armeabi folder and all JAR files.
 - d) In the libs folder, paste the armeabi folder and JAR files.
- 9. In Package Explorer, modify the build path to point to the correct location for the ClientLib.jar, sup-client.jar, and UltraLiteJNI12.jar files for the project:
 - a) Select the **SUP101Sample** project.
 - b) Select **Project > Properties > Java Build Path**.
 - c) Select the **Libraries** tab.
 - d) Click Add JARs.
 - e) In JAR Selection, expand the SUP101Sample\libs folder.
 - f) Select the JAR files, then click **OK**.
 - g) Click **OK** again.

Properties for SUP101Sa	mple	_ 🗆 🗵
type filter text	Java Build Path	$(-,\cdot, -,\cdot, -, -, -, -, -, -, -, -, -, -, -, -, -,$
 Resource Android Android Lint Preferences Builders Dava Build Path Dava Code Style Dava Editor Dava Editor Dava Editor Dava Editor Project Facets Project References Run/Debug Settings Server Task Tags Vaidation 	Source Projects Libraries Image: Order and JARs and class folders on the build path: Image: Order Lib.jar SUP101Sample/lbs Image: Order Lib.jar SUP101Sample/lib.jar Image: Order Lib.jar SUP101Sample/lib.jar Image: Order Lib.jar SUP101Sample/lib.jar Image: Order	Add JARs Add External JARs Add External JARs Add Variable Add Ubrary Add Class Folder Add External Class Folder Edt, Remove Migrate JAR, Elle
?		OK Cancel

Copying Unwired Platform Files to Sample Project

Copy the object API code you generated using the Generate Code wizard to the SUP101Sample project.

- 1. In Windows Explorer, go to the workspace directory, by default, C:\Documents and Settings\user\workspace\SUP101\Generated Code\src, and copy the SUP101 folder (generated code files).
- 2. In Package Explorer, go to the the SUP101Sample project and paste the SUP101 folder into the src directory.

You see the SUP101 and SUP101.intrnl folders.

Developing an Android Application



Configuring Android Application Properties

(Optional) Review the Android Manifest window, where you define the general Android properties used in an application.

- 1. In Package Explorer, expand the SUP101Sample project.
- 2. Double-click the AndroidManifest.xml file.
- 3. Select the Manifest tab.
- 4. Review the options in the Android Manifest window, where you can change the general attributes, export options, and content of the AndroidManifest.xml file.

Tip: Click **Uses Sdk** to indicate the API level for the minimum SDK version on which you want to run the application.

🔠 SUP101 - Mobile	Application Diagram	3UP101Sample Mar	nifest 🛙	
🐥 Android	Manifest	4		
▼ Manifest Gen	eral Attributes			
Defines general in	formation about the Andro	pidManifest.coml		
Package	com.sybase.sup.samples	s.objectapi		Browse
Yersion code	1			
Version name	1.0			Browse
Shared user id				Browse
Shared user label				Browse
Instal location				•
Manifest Extras	0 S P O O (Az		
① Uses Sdk	Add. Removi Up Down			
 Exporting To export the app 	lication for distribution, vo	u have the following opt	ions:	
Use the Export	t Wizard to export and sig	in an APK		
Export an unsi	igned APK and sign it man	ually		
▼ Links			i had a	
The content of the	e Android Manifest is made	e up of three sections. Yo	ou can also edit the XP	4L drectly.
D Permission : P	ectivities, intent niters, pro ermissions defined and ne	rmissions used.	eivers.	
I Instrumentat	ion: Instrumentation defin	red.		
F XML Source:	Directly edit the AndroidM	anifest.xml file.		
👾 Documentatio	n: Documentation from			
•				
🔚 Manifest 🖪 /	Application Permission	s II Instrumentation	AndroidManifest.	xml

Next

Modify the Android manifest file to add a Detail Activity class.

Adding User Permissions to the Android Project Manifest

Add user permissions to the Android project in the Android Manifest File.

- 1. If needed, open the Android Manifest.
- 2. Select the AndroidManifest.xml tab.
- 3. Add permissions to the AndroidManifest.xml file as a child element of the <manifest> element. You can use the AndroidManifest.xml file from the

SUP_Android_Custom_Dev_Tutorial_code.zip file to copy and paste the text.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.READ_PHONE_STATE" />
```

4. Select File > Save.

Adding a Class to the Android Manifest File

Add a Detail Activity class to the AndroidManifest.xml file. This declaration causes the application to launch a customer detail screen where you can make changes when you test the application.

1. In the Android Manifest Android Manifest.xml tab, add these values to the Android Manifest.xml file.

You can use the AndroidManifest.xml file from the

```
SUP_Android_Custom_Dev_Tutorial_code.zip file to cut and paste the entire <activity> element.
```

2. Select File > Save.

The XML file should look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/</pre>
android"
    package="com.sybase.sup.samples.objectapi"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="8" />
   <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.READ_PHONE_STATE" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".SUP101SampleActivity"
            android:label="@string/app name" >
            <intent-filter >
             <action android:name="android.intent.action.MAIN" />
                <category
android:name="android.intent.category.LAUNCHER" />
```

```
</intent-filter>
</activity>
</activity android:name=".DetailActivity"
android:label="@string/app_name">
<intent-filter>
</action android:name="android.intent.action.MAIN" />
</actegory
android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
</manifest>
```

Creating the User Interface

Copy the Java code files, which provide the functionality and layout of the user interface, from the SUP_Android_Custom_Dev_Tutorial_code.zip archive to the SUP101Sample application.

- 1. In Windows Explorer, browse to the directory where you saved the SUP_Android_Custom_Dev_Tutorial_code.zip file.
- 2. Copy these Java files: CustomerListAdapter.java, DetailActivity.java, and SUP101SampleActivity.java.
- 3. In Package Explorer, go to SUP101Sample\src \com.sybase.sup.samples.objectapi, then paste the copied Java files, copying over any existing files.
- 4. If you installed Unwired Server on a remote system, that is, not on the local system running Sybase Unwired Platform, you must modify the HOST IP address in the SUP101SampleActivity. java file to point to the server.
 - a) In Package Explorer, expand the SUP101Sample project.
 - b) Under the \src\com.sybase.sup.samples.objectapi folder, doubleclick the SUP101SampleActivity.java file.
 - c) Modify the HOST IP address, and Save.
- 5. Browse to the directory where you saved the ZIP file.
- 6. From the zip file, copy the sample layout XML files: customer.xml, detail.xml, and main.xml.
- 7. In the SUP101Sample project folder, go to the res\layout directory and paste the copied XML files, copying over any existing files.



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 Android platform.

Prerequisites

In the Unwired WorkSpace Window menu, use the AVD Manager to add a new target Android Virtual Device (AVD) for the launch configuration.

Task

- 1. In Package Explorer, right-click the **SUP101Sample** project, and select **Run As > Run Configurations**.
- 2. Right-click Android Application and select New.
- 3. In the Name field, enter SUP101Sample.
- 4. In the Android tab, click Browse and select SUP101Sample. Click OK.
- 5. In the Launch Action area, select Launch Default Activity.

🛟 Run Configurations		×
Create, manage, and ru Android Application	in configurations	
Image: Supervised state state Image: Supervised state <td>Name: SUP101Sample Project: SUP101Sample Browse. Launch Action: © Launch Default Activity © Launch: © Do Nothing Apply</td> <td>rt</td>	Name: SUP101Sample Project: SUP101Sample Browse. Launch Action: © Launch Default Activity © Launch: © Do Nothing Apply	rt
0	Run Cl	lose

- 6. In the **Target** tab, select a deployment target. For example, select **Automatic** and an AVD for deployment.
- 7. Keep the other default settings.

Name: SUP101Sample					
Android 🗊 Target 🔲 Common					
Deployment Target Selection Mode					
C Manual					
Automatic					
A Android 2.2 2.2 8 ARM Start					
Refresh Manager					
Emulator launch parameters:					
Network Speed: Full					
Network Latency: None 💌					
🔲 Wipe User Data					
Disable Boot Animation					
Additional Emulator Command Line Options					
Apply Revert					
Run Close					

8. Click Apply, then Close.

Testing the Device Application on the Android Emulator

Run the SUP101Sample application on the Android emulator, and change customer information to update the interface.

1. In Package Explorer, right-click the **SUP101Sample** and select **Run As > Android Application**.

Run As	l		🗂 1 Android Application	N
Debug As			🔓 2 Android JUnit Test	M.
Profile As	I		🛐 3 Java Applet	Alt+Shift+X, A
Validate			J 4 Java Application	Alt+Shift+X, J
Team	I		Tu 5 Il Init Test	Alt+Shift+X, T
Compare With	l l	۱ <u>-</u>	,	
Restore from Local History			Run Configurations	
Android Tools	1	۲		
Configure	l l	•		
Properties	Alt+Enter			

Note: It may take several minutes for the Android emulator's home screen to appear.

The On Boarding image indicates that the application is registering and synchronizing data from Unwired 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();
```

When the data finishes synchronizing, the device application shows the SUP101Sample Application with a list of customer data in a ListView control. You can scroll through the customer list to see more data and to make changes. The data loads from the database on demand.

Note: The sample application illustrates a device application with a small buffer (30 customers). In commercial applications, you can use a large buffer (1,000 customers) based on user data.

When the application queries the customer list, 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.

🚯 📶 💶 11:58 ам
SUP101Sample
101 Michaels Devlin
102 Beth Reiser
103 Erin Niedringhaus
104 Meghan Mason
105 Laura McCarthy
106 Paul Phillips
107 Kelly Colburn
108 Matthew Goforth
109 Jessie Gagliardo
110 Michael Agliori
111 Dylan Ricci
112 Shawn McDonough

2. To change customer information, select the customer, for example, Meghan Mason.



3. In the customer detail screen, change the first name of the customer and click **Submit**.

🔣 📶 亿 10:21 ам						
SUP101Sample						
First Name	Melissa 🖌					
Last Name	Mason					
address	550 Dundas Street East					
phone	6155555463					
Submit						

The Submit button is mapped to the synchronize operation using SUP101DB.beginSynchronize. The synchronization occurs in the background so the user interface is not affected.

Any back-end changes initiate notifications from the server. The device application uses a ChangeLog API to query those managed items and use them to update the user interface if needed.

GenericList<ChangeLog> changeLogs=SUP101DB.getChangeLogs(query);

Fi 📊 💶 10	:25 AM
SUP101Sample	
101 Michaels Devlin	
102 Beth Reiser	
103 Erin Niedringhaus	
104 Melissa Mason	
105 Laura McCarthy	
106 Paul Phillips	
107 Kelly Colburn	
108 Matthew Goforth	
109 Jessie Gagliardo	
110 Michael Agliori	
111 Dylan Ricci	
112 Shawn McDonough	

4. Close the emulator to stop the SUP101Sample application.

Developing an Android 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[®] 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[®] 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

ADT Plugin for Eclipse, installing 10 Android application attributes 18 Android project 13 manifest file 20 src folder 17 Android SDK 9 AndroidManifest.xml 13, 18 Detail Activity 20 application properties 18

В

build path 13

С

ClientLib.jar 13 customer.xml 21 CustomerListAdapter.jar 21

D

default_package.jar 11 deployment target, launch 22 deployment_unit.xml 11 Detail Activity 20 detail.xml 21 DetailActivity.jar 21

Ε

example projects 1

G

Generate Code wizard 11 generated object API code 11 using 17 generating code 11

J

JAR files ClientLib.jar 13 sup-client.jar 13 UltraLiteJNI12.jar 13 Java class, creating 21 Java files CustomerListAdapter.jar 21 DetailActivity.jar 21 SUP101SampleActivity.jar 21 Java object API code, generating 11 Java perspective 21 JDK 9

L

launch configuration 22 layout files customer.xml 21 detail.xml 21 main.xml 21

Μ

main.xml 21 manifest file 13, 20 mobile business object tutorial 1 Mobile Workflow package tutorial 1

0

Object API tutorials 1

Ρ

project build path 13 properties, application 18

R

run configurations 22

S

samples downloading 33 simulator 9

src folder Android project 17 Unwired WorkSpace 17 sup-client.jar 13 SUP101Sample application, testing 25 SUP101SampleActivity.jar 21 SUP101SampleProject 13 Sybase Control Center, connecting to 4 Sybase Mobile SDK installing 3 Sybase Unwired Platform documentation resources 33 getting started 3 installing 3 Sybase Unwired WorkSpace basics 5 how to access online help 5 starting 4

Т

troubleshooting information 5 tutorials 1 downloading 33

U

UltraLiteJNI12.jar 13 Unwired Platform Runtime installing 3 Unwired Platform services 3 Unwired WorkSpace basics 5 Unwired WorkSpace, src folder 17 user interface, creating 21

V

virtual devices 9

X

XML files customer.xml 21 detail.xml 21 main.xml 21