



**Tutorial: Android Object API Application
Development**

Sybase Unwired Platform 2.1

ESD #2

DOCUMENT ID: DC01734-01-0212-01

LAST REVISED: March 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
Task Flow	3
Getting Started with Unwired Platform	5
Installing Sybase Unwired Platform	5
Starting Sybase Unwired Platform Services	5
Starting Sybase Unwired WorkSpace	6
Connecting to Sybase Control Center	6
Learning Unwired WorkSpace Basics	7
Developing an Android Application	11
Installing the Android SDK	11
Installing ADT in Unwired WorkSpace	11
Generating Java Object API Code	12
Creating the Android Project	14
Configuring Android Application Properties	19
Copying Unwired Platform Files to Sample Project	20
Modifying the Android Manifest File	21
Creating the User Interface	22
Creating a Launch Configuration for the Project	24
Testing the Device Application on the Android Emulator	27
Learn More About Sybase Unwired Platform	35
Index	37

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® SDN: <http://www.sdn.sap.com/irj/sdn/mobile?rid=/webcontent/uuid/40ea4956-b95c-2e10-11b3-e68c73b2280e>.

- 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*

Task Flow

Use this tutorial to develop, deploy, and test a mobile device application on a simulator or an emulator.

Task	Goals	Steps
Getting started	<ul style="list-style-type: none"> • Install Sybase Mobile SDK and Sybase Unwired Platform Runtime. • Start Unwired Server and other platform services, if not already started. • Start Sybase Control Center. • Start Sybase Unwired WorkSpace, open the Mobile Development perspective, and become familiar with the views of the perspective and the Mobile Application Diagram. 	<ul style="list-style-type: none"> • Installing Sybase Unwired Platform • Starting Unwired Platform Services • Starting Sybase Unwired WorkSpace • Connecting to Sybase Control Center • (Optional) Learning Unwired WorkSpace Basics <p>These steps are prerequisites for the rest of this tutorial. You need to perform them only once.</p>
Developing database mobile business objects	<ul style="list-style-type: none"> • Create a mobile application project and a connection to the database. • Create two mobile business objects, and create a relationship between them. • Deploy the mobile business objects to Unwired Server. 	<p>Complete the <i>Tutorial: Mobile Business Object Development</i>, or obtain the completed example project.</p> <hr/> <p>Note: This tutorial is a prerequisite for the remaining steps. You need to perform it only once. If you want to download the final outcome of a tutorial without performing it, the associated example project is available on SAP® SDN: http://www.sdn.sap.com/irj/sdn/mobile?rid=/webcontent/uuid/40ea4956-b95c-2e10-11b3-e68c73b2280e</p>

Task Flow

Task	Goals	Steps
Developing a native device application	<ul style="list-style-type: none">• Install the Android SDK• Install ADT in Unwired WorkSpace• Generate Java Object API code• Create the user interface• Test the device application on the Android emulator	<ul style="list-style-type: none">• Installing the Android SDK• Installing ADT in Unwired WorkSpace• Generating Java Object API Code• Creating the Android Project• Creating the User Interface• Creating a Launch Configuration for the Project• Testing the Device Application on the Android Emulator

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, be sure you have all the requisite Unwired Platform components installed. For complete installation instructions, see the Sybase Unwired Platform documentation at <http://sybooks.sybase.com/nav/summary.do?prod=1289>.

- *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 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.

How you start Unwired Platform services depend on the options you selected during installation. In some cases, 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 perform tasks that include creating mobile business objects (MBOs), managing database and server connections, developing Mobile Workflow applications, and generating 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 service does not open, make sure that the service is started. See the *Installation Guide for Runtime*.

2. In Sybase Control Center, 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 not familiar 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 up and running.
 - 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	<p>Use this view to create Mobile Application projects, and review and modify MBO-related properties.</p> <p>This view displays mobile application project folders, each of which contains all project-related resources in subfolders, including MBOs, datasource references to which the MBOs are bound, personalization keys, and so on.</p>
Enterprise Explorer view	<p>A view that provides functionality to connect to various enterprise information systems (EIS), such as database servers, SAP® back ends, and Unwired Server.</p>

Window	Description
<p>Mobile Application Diagram</p>	<p>The Mobile Application Diagram is a graphical editor where you create and define mobile business objects.</p> <p>Use the Mobile Application Diagram to create MBOs (including attributes and operations), then define relationships with other MBOs. You can:</p> <ul style="list-style-type: none"> • Create MBOs in the Mobile Application Diagram using Palette icons and menu selections – either bind or defer binding to a data source, when creating an MBO. For example, you may want to model your MBOs before creating the data sources to which they bind. This MBO development method is sometimes referred to as the top-down approach. • Drag items from Enterprise Explorer and drop them (drag and drop) onto the Mobile Application Diagram to create the MBO – quickly creates the operations and attributes automatically based on the datasource artifact being dropped on the Mobile Application Diagram. <p>Each new mobile application project generates an associated mobile application diagram.</p>
<p>Palette</p>	<p>The Palette is accessed from the Mobile Application Diagram and provides controls, such as the ability to create MBOs, add attributes and operations, and define relationships, by dragging-and-dropping the corresponding icon onto the Mobile Application Diagram or existing MBO.</p>
<p>Properties view</p>	<p>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 configuration is performed here.</p>
<p>Outline view</p>	<p>Displays an outline of the active file and lists structural elements. The contents are editor-specific.</p>
<p>Problems view</p>	<p>Displays problems, errors, or warnings that you may encounter. This is a valuable source for collecting troubleshooting information.</p>

Window	Description
Error Log view	Displays error log information. This is a valuable source for collecting troubleshooting information.

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* on page 5.
- Complete *Tutorial: Mobile Business Object Development*, which provides the foundation tasks for this tutorial.

Task

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

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 SDK starter package from <http://developer.android.com/sdk/index.html>.
3. Launch the **Android SDK Manager** and install the Android SDK tools, platform, and compatibility package for Android.
4. Launch the **Android Virtual Device Manager**, and create an Android virtual device to use as your simulator.

Installing ADT in Unwired WorkSpace

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

1. Download the ADT Plugin for Eclipse at <http://dl.google.com/android/ADT-16.0.1.zip>.
2. Start Unwired WorkSpace, then select **Help > Install New Software**.
3. In the Available Software window, click **Add**.
4. In the Add Repository dialog, click **Archive**.

5. Select the ADT Plugin for Eclipse ZIP file.
6. Enter a Name for the local update site, such as Android Plugin, then click **OK**.
7. In the Available Software dialog, select **Developer Tools**, then click **Next**.
8. In the next window, a list of downloadable tools, click **Next**.
9. Accept the license agreements, then click **Finish**.

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

10. When the installation completes, restart Unwired WorkSpace.

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**.

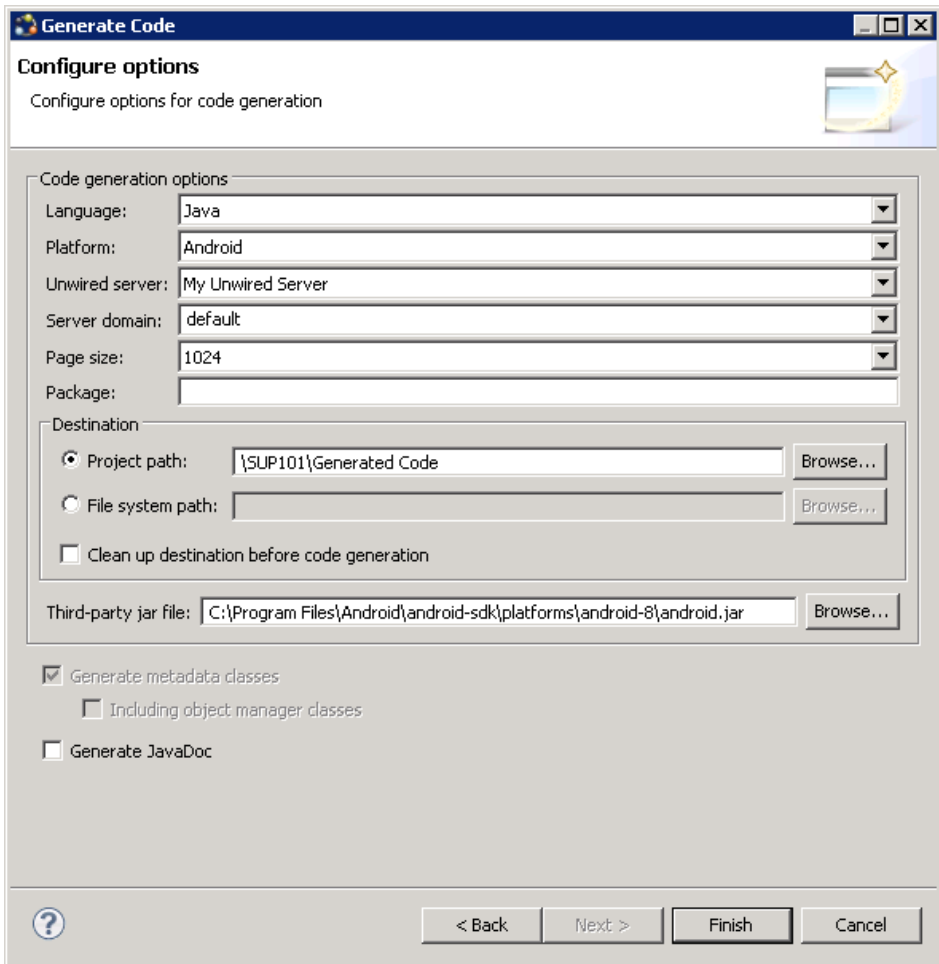
Task

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 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 .
Server domain	Select default .
Page size	Select 1024 .
Package	(Optional) Enter a unique name for the Java package.
Project path	Leave the default <code>\SUP101\Generated Code</code> , or browse to another folder you created for the device platform in Step 2.
Third-party jar file	Click Browse to open an <code>android.jar</code> .
Generate JavaDoc	Deselect for this tutorial.



7. 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. You add library resources and set other application properties.

Prerequisites

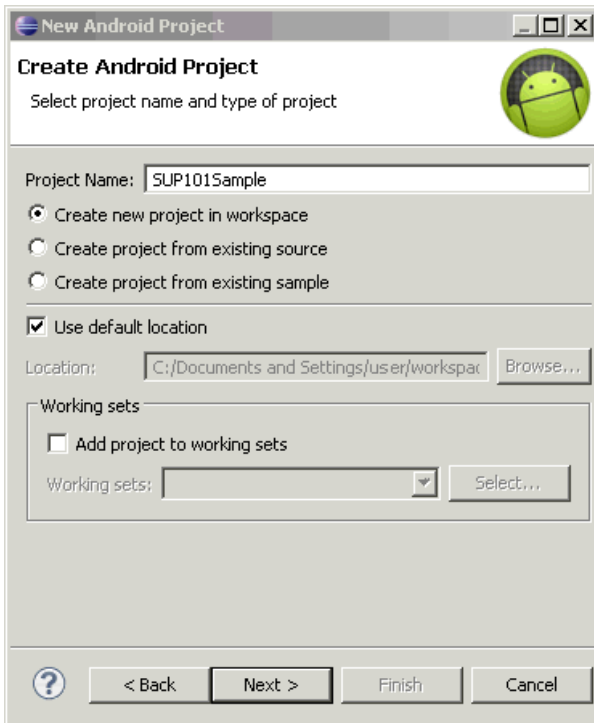
- In Unwired WorkSpace Preferences, set the Android SDK Location.

- 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. In this section you can use a text file to help modify the Android Manifest file.
 - If you are viewing this guide online from the Sybase Product Documentation 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/nav/summary.do?prod=1289>. Click the link for the appropriate Sybase Unwired Platform version. Then, navigate to this topic in the tutorial, and click the link for the ZIP file to access the text files.

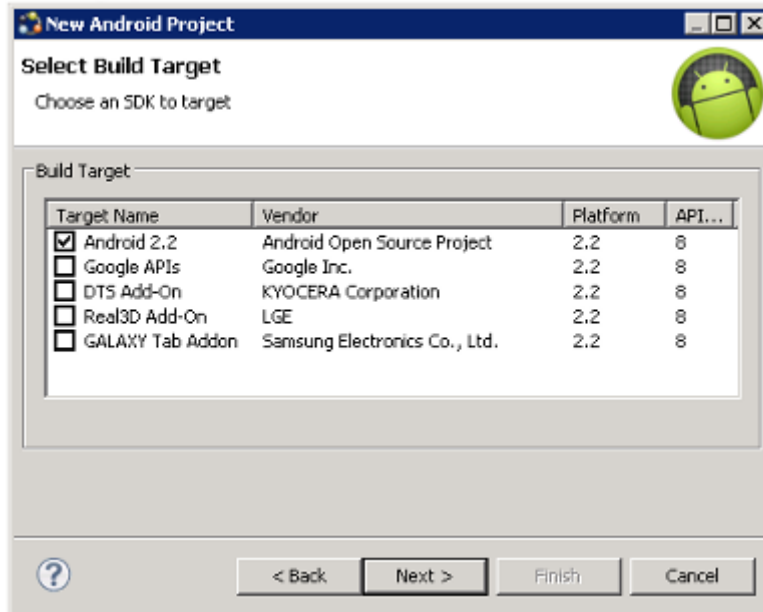
Task

1. Start Unwired WorkSpace.
2. Select **File > New > Project**.
3. Select **Android > Android Project** and **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.
4. In the New Android Project wizard, use these values and click **Next**.
 - Project Name – enter `SUP101Sample`.
 - Select **Create new project in workspace** if it is not already selected.
 - Select **Use default location** if it is not already selected with, for example, `C:/Documents and Settings/user/workspace/SUP101Sample`.

Developing an Android Application



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

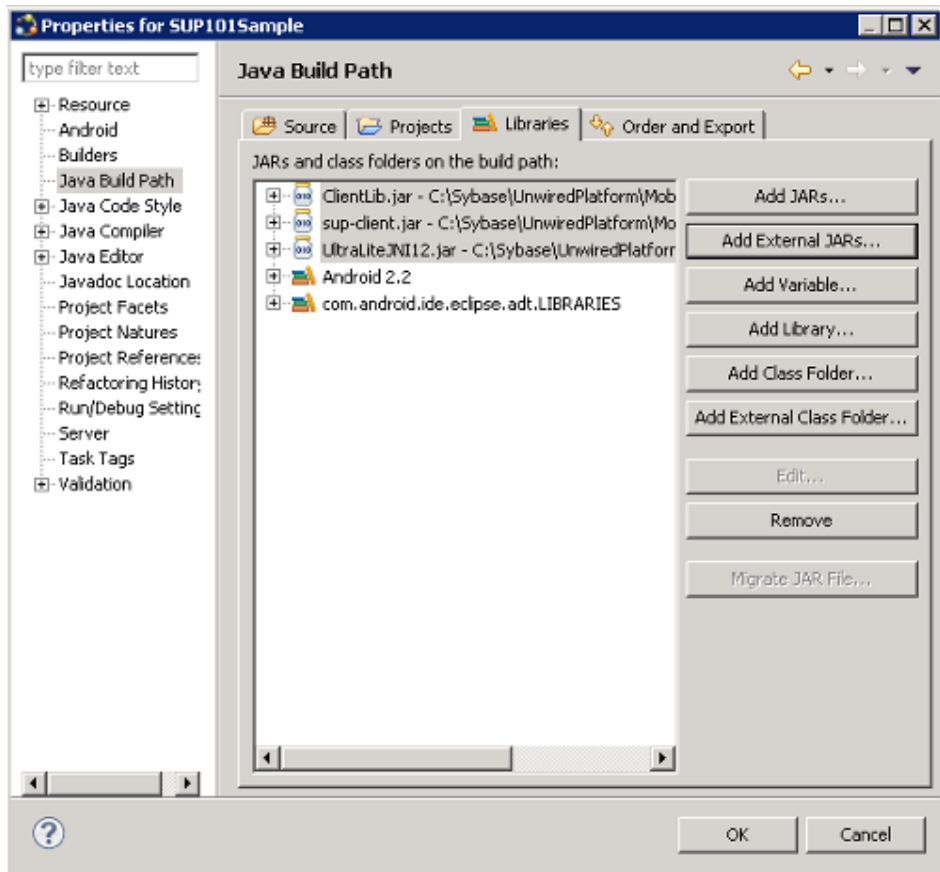


6. In the Application Info window, use these values and click **Finish**.
 - Application Name – enter `SUP101Sample` if it does not already appear.
 - Package Name – enter `com.sybase.sup.samples.objectapi`.
 - Click **Create Activity** and enter `SUP101SampleActivity` if these do not already appear.

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

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

7. 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 External JARs** and browse to `C:\Sybase\UnwiredPlatform\MobileSDK\ObjectAPI\Android`.
 - e) Select all the JAR files, then click **Open**.
 - f) Click **OK**.



8. Add a compiler resource to the root directory of the Android project:
 - a) In Windows Explorer, browse to C:\Sybase\UnwiredPlatform\MobileSDK\ObjectAPI\Android.
 - b) Copy the armeabi folder.
 - c) In Package Explorer, select **SUP101Sample** and add a libs folder.
 - d) In the libs folder, paste the armeabi folder.
9. In Package Explorer, add user permissions to the project:
 - a) Expand the **SUP101Sample** project.
 - b) Double-click the AndroidManifest.xml file.
 - c) Select the **AndroidManifest.xml** tab.
 - d) Add permissions to the AndroidManifest.xml file as a child element of the <manifest> element. You can use the AndroidManifest.xml file from the ZIP archive to cut and paste the following text:

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

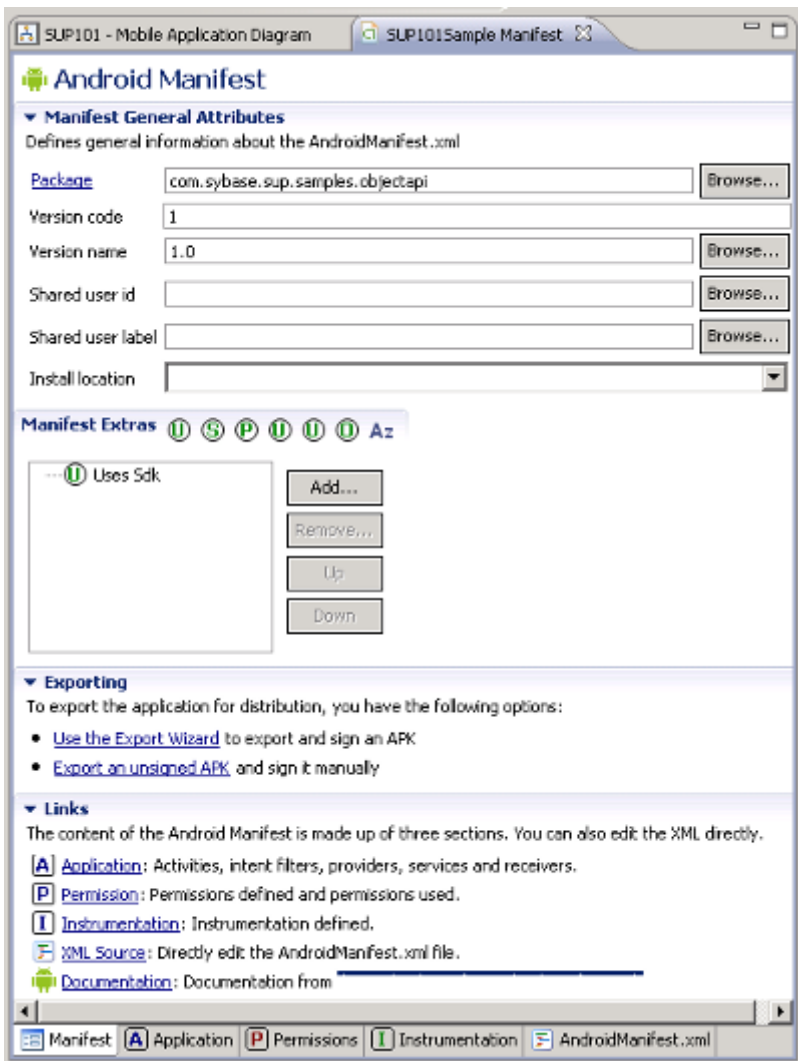
e) Select **File > Save**.

Configuring Android Application Properties

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

For this tutorial, you would keep the default settings.

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. This is an area where you would change the general attributes, export options, and content of the `AndroidManifest.xml` file.
5. Click **File > Close** to close the SUP101Sample Manifest file without any changes.

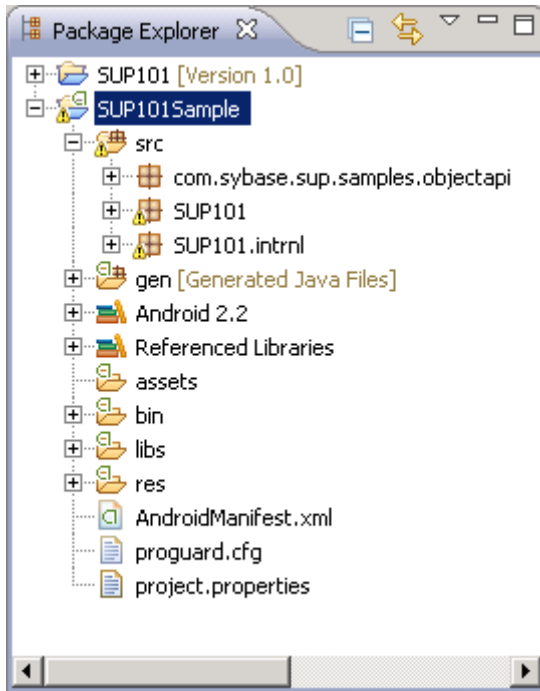


Copying Unwired Platform Files to Sample Project

Copy the generated object API code to the SUP101Sample project. You had created the code previously in the Generate Code wizard.

1. In Windows Explorer, go to `C:\Documents and Settings\\workspace\\Generated Code\src`, and copy the generated code files.
2. In Package Explorer, go to the the SUP101Sample project and paste the folder into the `src` directory.

You see the SUP101 and SUP101.intrnl folders.



Modifying 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 Package Explorer, double-click the `AndroidManifest.xml` file.
2. Select the **AndroidManifest.xml** tab.
3. Add these values to the `AndroidManifest.xml` file. You can use the `AndroidManifest.xml` file from the ZIP archive to cut and paste the entire `<activity>` element:

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

4. 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/
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" />
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Creating the User Interface

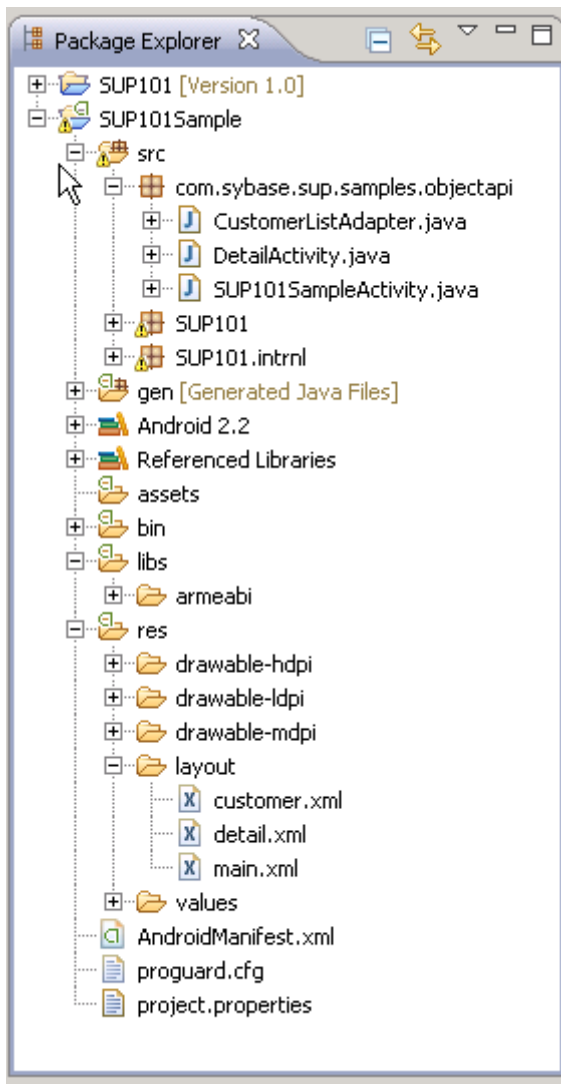
Copy sample files from the ZIP archive to the SUP101Sample application. The files provide the functionality and layout of the user interface.

Prerequisites

Locate the files you extracted from the SUP_Android_Custom_Dev_Tutorial_code.zip file. You will use the text files, which contain code snippets, to build the user interface. You copy and paste the code snippets to create these classes: CustomerListAdapter, DetailActivity, and SUP101SampleActivity.

Task

1. In Windows Explorer, browse to the directory where you saved the 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`. Paste the copied Java files. Copy over any existing files.
4. If you installed the Sybase Unwired Platform 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, double-click the `SUP101SampleActivity.java` file.
 - c) Modify the HOST IP address, and **Save**.
5. Browse to the directory where you saved the ZIP file.
6. 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. Copy over any existing files.



Creating a Launch Configuration for the Project

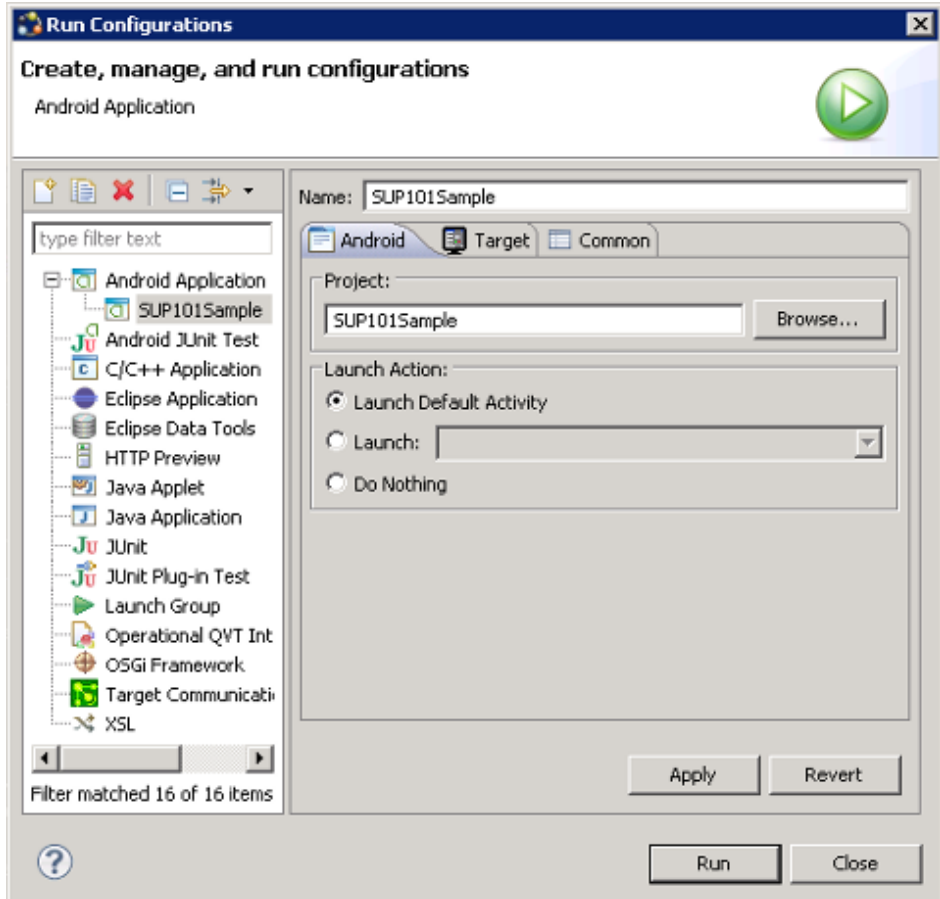
Create and define a new launch configuration for the SUP101Sample project. The configuration defines how the application will launch and the target Android platform.

Prerequisites

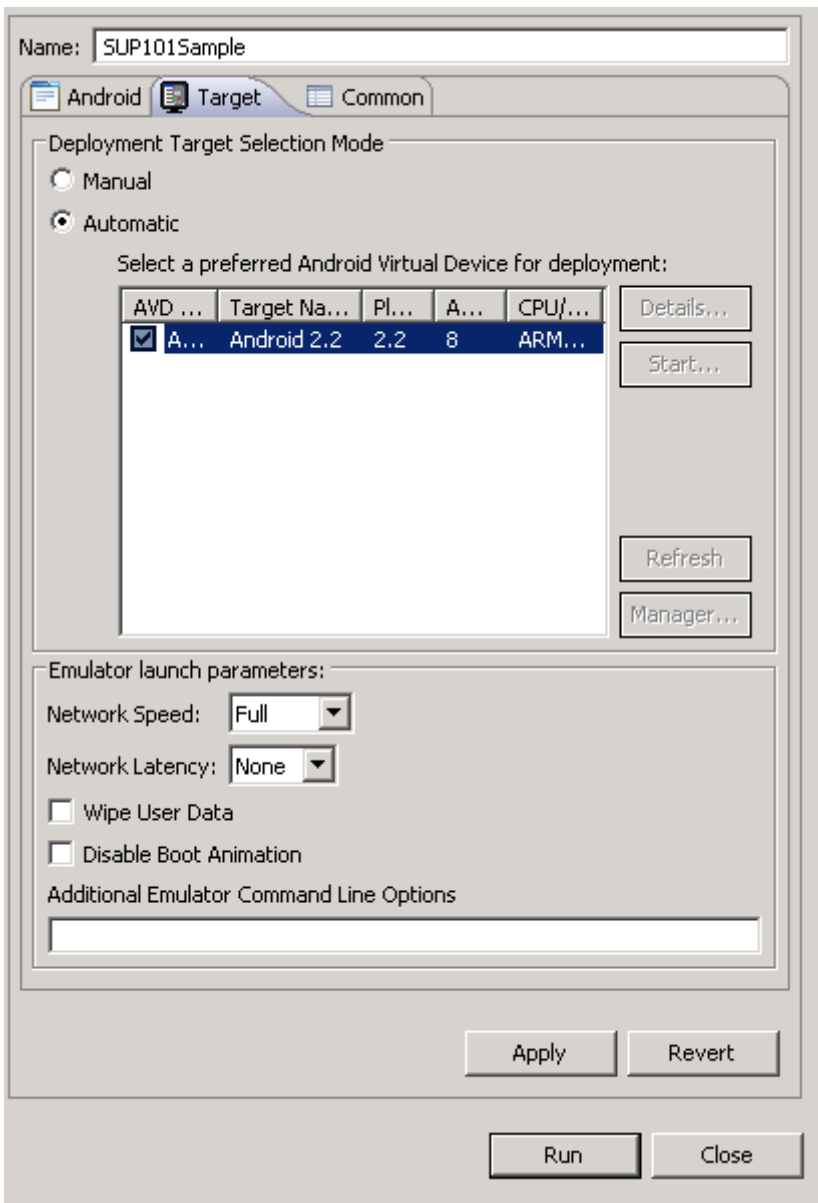
In the Unwired WorkSpace Window menu, use the AVD Manager to add a new target Android Virtual Device (AVD) for the launch configuration.

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** if it is not already selected.



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

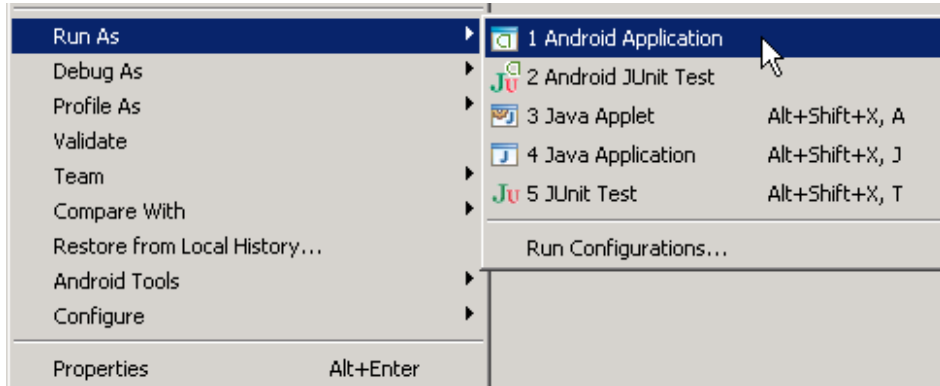


8. Click **Apply**, then **Close**.

Testing the Device Application on the Android Emulator

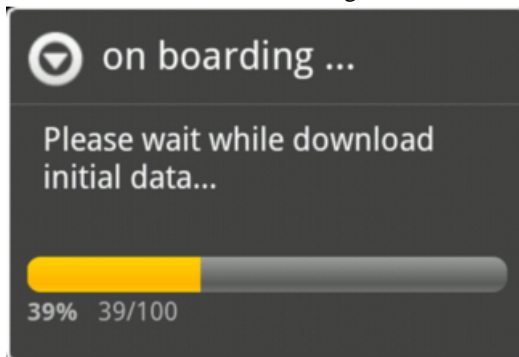
Run the SUP101Sample application on the Android emulator, and change customer information to update the interface.

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



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 the server in the background.



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

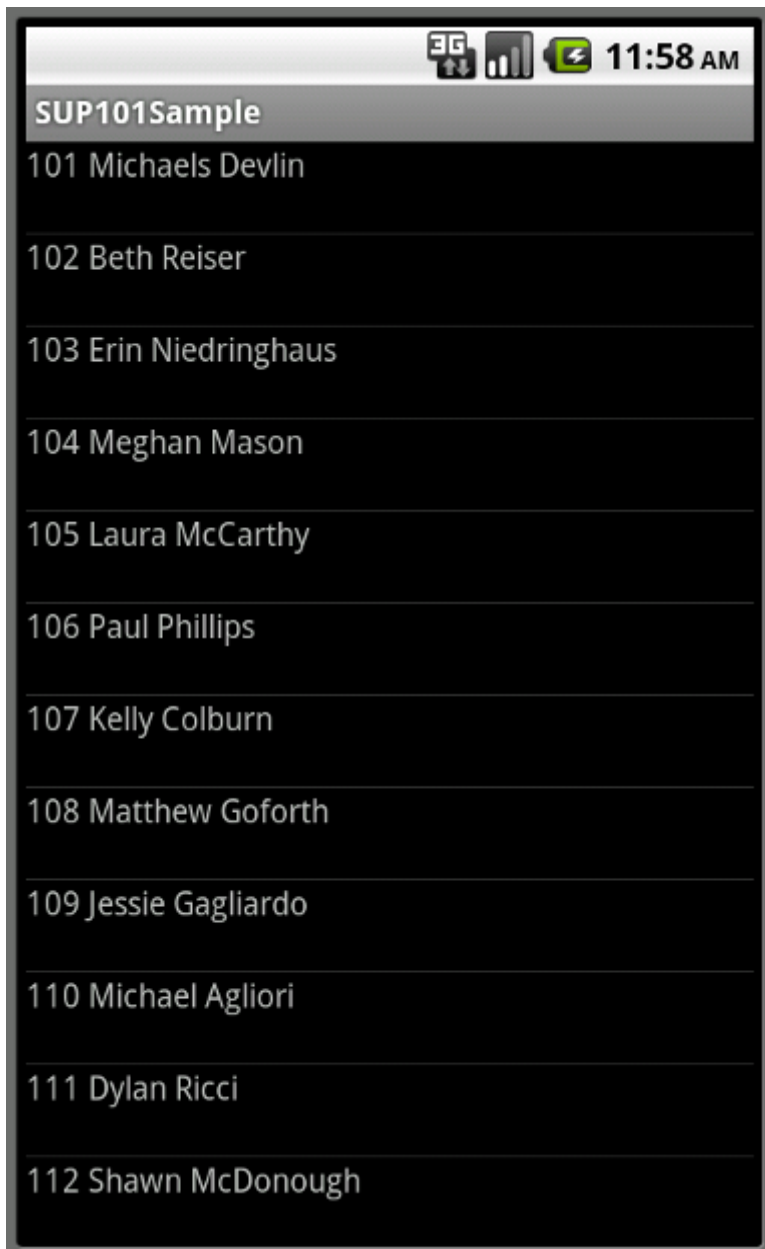
```
SynchronizationGroup
sg=SUP101DB.getSynchronizationGroup("default");
sg.setEnableSIS(true);
sg.save();
```

Developing an Android Application

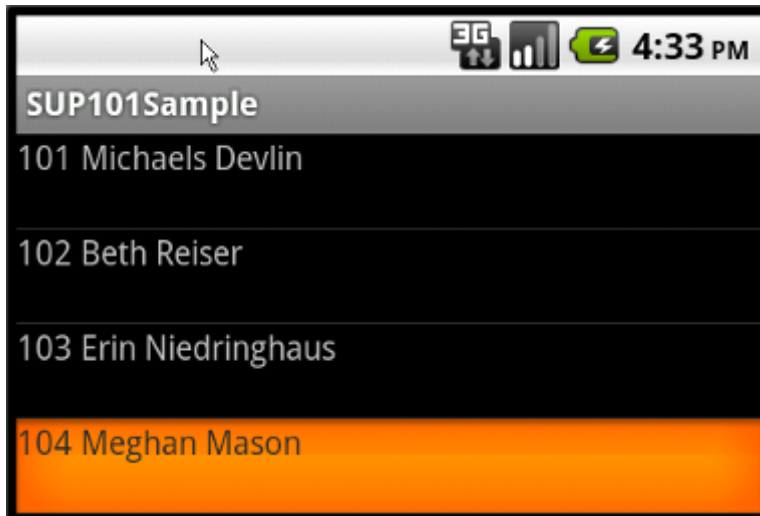
When the data finishes synchronizing, the device application shows the SUP101Sample Application with a list of customer data in ListView control. You can scroll through the customer list to see more data and to make changes. The data loads from the database on demand.

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.



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**.

SUP101Sample

First Name

Last Name

address

phone

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



4. Close the emulator 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/nav/summary.do?prod=1289>, then navigate to the most current version.

Tutorials

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

Example Projects

Example projects are available for download, if you want the finished tutorial without going through the steps. Download example projects from: <http://www.sdn.sap.com/irj/sdn/mobile?rid=/webcontent/uuid/40ea4956-b95c-2e10-11b3-e68c73b2280e>.

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 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 how to develop mobile business objects (MBOs) to fully maximize their potential.

Learn about using the 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:

Learn More About Sybase Unwired Platform

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

Javadoc and HeaderDoc are also available in the installation directory.

Index

A

- ADT Plugin for Eclipse, installing 11
- Android application
 - attributes 19
 - developing 11
- Android emulator 27
- Android project
 - creating 14
 - manifest file 21
 - src folder 20
- Android SDK 11
- AndroidManifest.xml 14, 19
 - Detail Activity 21
- application properties 19

B

- build path 14

C

- ClientLib.jar 14
- compiler, adding 14
- configurations
 - launch 24
 - run 24
- customer.xml 22
- CustomerListAdapter.jar 22

D

- default_package.jar 12
- deployment target, launch 24
- deployment_unit.xml 12
- Detail Activity 21
- detail.xml 22
- DetailActivity.jar 22

E

- emulator 27
- example projects 1

G

- Generate Code wizard 12

- generated object API code 12
 - using 20
- generating code 12

J

- JAR files
 - adding 14
 - ClientLib.jar 14
 - sup-client.jar 14
 - UltraLiteJNI2.jar 14
- Java class, creating 22
- Java files
 - CustomerListAdapter.jar 22
 - DetailActivity.jar 22
 - SUP101SampleActivity.jar 22
- Java object API code, generating 12
- Java perspective 22
- JDK 11

L

- launch configuration 24
- layout files
 - customer.xml 22
 - detail.xml 22
 - main.xml 22

M

- main.xml 22
- manifest file 14, 21
- mobile business object tutorial 1
- Mobile Workflow package tutorial 1

O

- Object API tutorials 1

P

- project build path 14
- properties, application 19

Index

R

run configurations 24

S

samples, how to download 35

simulator 11

src folder

 Android project 20

 Unwired WorkSpace 20

sup-client.jar 14

SUP101Sample application, testing 27

SUP101SampleActivity.jar 22

SUP101SampleProject 14

Sybase Control Center, connecting to 6

Sybase Mobile SDK

 installing 5

Sybase Unwired Platform

 documentation resources 35

 getting started 5

 installing 5

Sybase Unwired WorkSpace

 basics 7

 how to access online help 7

 starting 6

T

task flow 3

troubleshooting information 7

tutorials 1

tutorials, how to download 35

U

UltraLiteJNI12.jar 14

Unwired Platform Runtime

 installing 5

Unwired Platform services 5

Unwired WorkSpace basics 7

Unwired WorkSpace, src folder 20

user interface, creating 22

user permissions 14

V

virtual devices 11

X

XML files

 customer.xml 22

 detail.xml 22

 main.xml 22

 manifest 21