SYBASE®
An SAP® Company

Tutorial: Android Object API Application Development

Sybase Unwired Platform 2.1 ESD#1

DOCUMENT ID: DC01734-01-0211-01
LAST REVISED: January 2012

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

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.

- Learn mobile business object (MBO) basics, and use this as a foundation to create mobile device applications and mobile workflow applications:
  - *Tutorial: Mobile Business Object Development*
- Create native 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 workflow package:
  - *Tutorial: Mobile Workflow Package Development*

The tutorials demonstrate a cross section of basic functionality, which includes creating MBOs, and using various Sybase Unwired WorkSpace development tools, independent development environments, and device types. Tutorial projects are available if you want the finished tutorial without going through the steps: *http://www.sdn.sap.com/irj/sdn/mobile?rid=/webcontent/uuid/40ea4956-b95c-2e10-11b3-e68c73b2280e*.

Task Flow

Use this tutorial to develop an Android device application using replication-based synchronization and custom coding. Test the application on an emulator.

Table 1. Eclipse tutorials

Task	Goals	Steps required to complete the task
Getting Started	• Install all required Unwired WorkSpace components and external resources. • Start Unwired Server and other platform services, if not already started. • Open the Mobile Development perspective, and become familiar with the views of the perspective, and the Mobile Application Diagram.	• *Installing Sybase Unwired Platform* on page 5 • *Starting Sybase Unwired WorkSpace* on page 5 • *(Optional) Learning Unwired WorkSpace Basics* on page 6 **Note:** These steps are prerequisites for the rest of this tutorial. You need to perform them only once.
Developing Database Mobile Business Objects	• Create a mobile application project and a connection to the database. • Create a mobile business object and deploy it to Unwired Server.	Complete the *Tutorial: Mobile Business Object Development*. **Note:** This tutorial is a prerequisite for the remaining steps. You need to perform it only once.

Task	Goals	Steps required to complete the task
Developing an Android Application	Generate Java code for the Android platform, create the user interface for the application, and run it on the emulator.	• *Installing the Android SDK* on page 9 • *Installing ADT in Unwired WorkSpace* on page 9 • *Generating Java Object API Code* on page 10 • *Creating the Android Project* on page 11 • *Creating the User Interface* on page 16 • *Creating a Launch Configuration for the Project* on page 18 • *Testing the Device Application on the Android Emulator* on page 20

Getting Started

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 Unwired Platform.

Before starting this tutorial, be sure you have all the requisite Unwired Platform components installed. For complete installation instructions, see:

- *Release Bulletin for Sybase Mobile SDK*
- *Installation Guide for Sybase Mobile SDK*
- *Release Bulletin for Runtime*
- *Installation Guide for Runtime*

- Install Sybase 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.
- Install Unwired Platform Runtime:
  - Data Tier (included with single-server installation)
  - Unwired Server

Starting Sybase Unwired WorkSpace

Start Unwired WorkSpace.
Select **Start** > **Programs** > **Sybase** > **Unwired Platform** > **Unwired WorkSpace**.

The Sybase Unwired WorkSpace opens in the Mobile Development perspective. The Welcome page displays links to product information, and to the product.

Next

To read more about Sybase Unwired WorkSpace concepts and tasks, select **Help** > **Help Contents** from the main menu.

Learning Unwired WorkSpace Basics

If you are already familiar with Eclipse, you will find Sybase Unwired Platform features are well integrated. If you are not familiar, you can quickly learn the basic layout of Unwired WorkSpace and the location of online help.

- From the Welcome page, select the **Development** icon to learn about the tasks you must perform. To close this page, click the **X**.
- You can reopen the Welcome page by selecting **Help** > **Welcome**.
- From Sybase Unwired WorkSpace, look at the area (window or view) that you will be working in to access, create, define, and update mobile business objects(MBOs).

Window	Description
WorkSpace Navigator view	This view displays mobile application project folders, each of which contains all project-related resources in subfolders, including MBOs, data source references to which the MBOs are bound, personalization keys, and so on. Use this view to review and modify MBO-related properties.
Enterprise Explorer view	A window that provides functionality to connect to various enterprise back-end systems; for example, database servers, SAP® servers, and Sybase Unwired Server.

Window	Description
Mobile Application Diagram	The Mobile Application Diagram is a graphical editor 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 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 data source being dropped on the Mobile Application 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 define relationships, by dragging-and-dropping the corresponding icon onto the Mobile Application Diagram 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 configuration is performed here.
Outline view	Displays an outline of the file that is currently open in the editor area, and lists structural elements. The contents are editor-specific.

Window	Description
Problem view	Displays problems, errors, or warnings that you may encounter. This is a valuable source for collecting troubleshooting information.
Error Log view	Displays error log information. This is a valuable source for collecting troubleshooting information.

- To access the online help, select **Help > Help Contents** from the main menu bar. Expand any of the documents that appear in the left pane. Some documents are for Sybase Unwired Platform, while others are for the Eclipse development environment.

Developing an Android Application

Generate code for the Android platform, develop an Android device application with code, and test its functionality.

Prerequisites

Complete these tasks:

- 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

The device application 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-15.0.1.zip*.
2. Start Eclipse, then select **Help > Install New Software**.
3. Click **Add**, in the top-right corner.

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 **Development 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 code generation batch file to generate object API code for the SUP101 mobile application project.

Prerequisites

You must be connected to both My Sample Database and My Unwired Server. Code generation fails if the server-side (run-time) enterprise information system (EIS) data sources referenced by the MBOs in the project are not running and available to connect to when you generate object API code.

Task

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

   In WorkSpace Navigator, highlight the `SUP101` folder.

   Note: If you do not see the SUP101 project in Workspace Navigator, you must first complete the *Tutorial: Developing Mobile Business Objects*, which is a prerequisite for this tutorial.
2. Optional: If you are performing other tutorials, add a new folder to the project in which to generate code for each device platform.
   For example, in WorkSpace Navigator, expand **SUP101** and add an `Android` folder under `Generated Code`. The `Generated Code` directory was created during the MBO Tutorial.
3. In Windows Explorer, locate `<domain name>_package.jar` in your mobile project folder. For the SUP101 example, the project is deployed to the default domain, and the deploy jar file is in the following location: `C:\Documents and Settings \user\workspace\SUP101\Deployment\.pkg.profile \My_Unwired_server\default_package.jar`.
4. Make sure that the JAR file contains this file: `deployment_unit.xml`.

5. Use a utility to extract the `deployment_unit.xml` file to another location.
6. From `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Utils\bin`, run the `codegen.bat` utility, specifying the following parameters:

   ```
   codegen.bat -java -client -android -ulj <path>
   \deployment_unit.xml [-output <output_dir>] [-doc]
   ```

   - The `-output` parameter allows you to specify an output directory. If you omit this parameter, the output goes into the `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Utils\genfiles` directory, assuming `codegen.bat` is run from the `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Utils\bin` directory. You can direct the output to the `Android` folder you created in the optional step.
   - The `-doc` parameter specifies that documentation is generated for the generated code.

   Ignore these warnings:

   ```
   log4j:WARN No appenders could be found for logger ...
   log4j:WARN Please initialize the log4j system properly.
   ```

Creating the Android Project

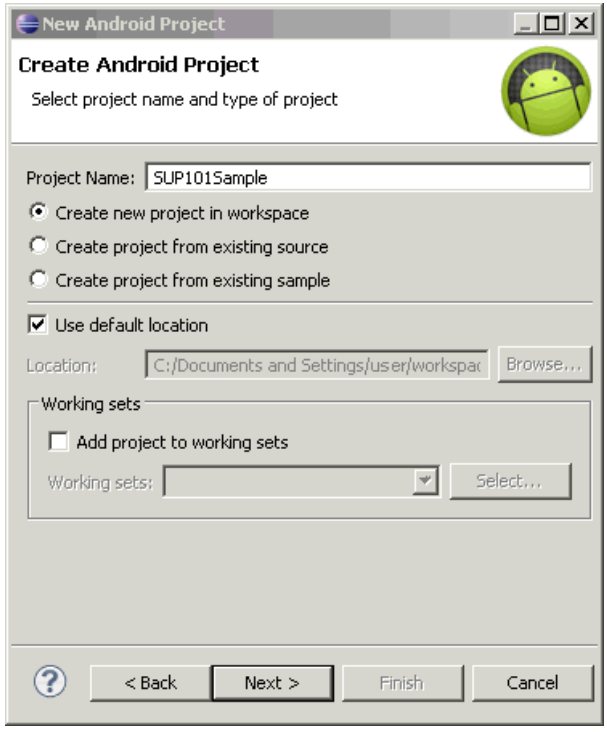
Create a new Android project in the Android Plugin for Eclipse, where you add functionality and finish developing the application.

Prerequisites

In the Unwired WorkSpace, you need to set the SDK locations.

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 Create Android Project wizard, use these values and click **Next**.
   - Project Name - enter `SUP101Sample`.
   - Verify **Create new project in workspace** is selected.
   - Verify **Use default location** is selected, for example, `C:/Documents and Settings/user/workspace/SUP101Sample`.

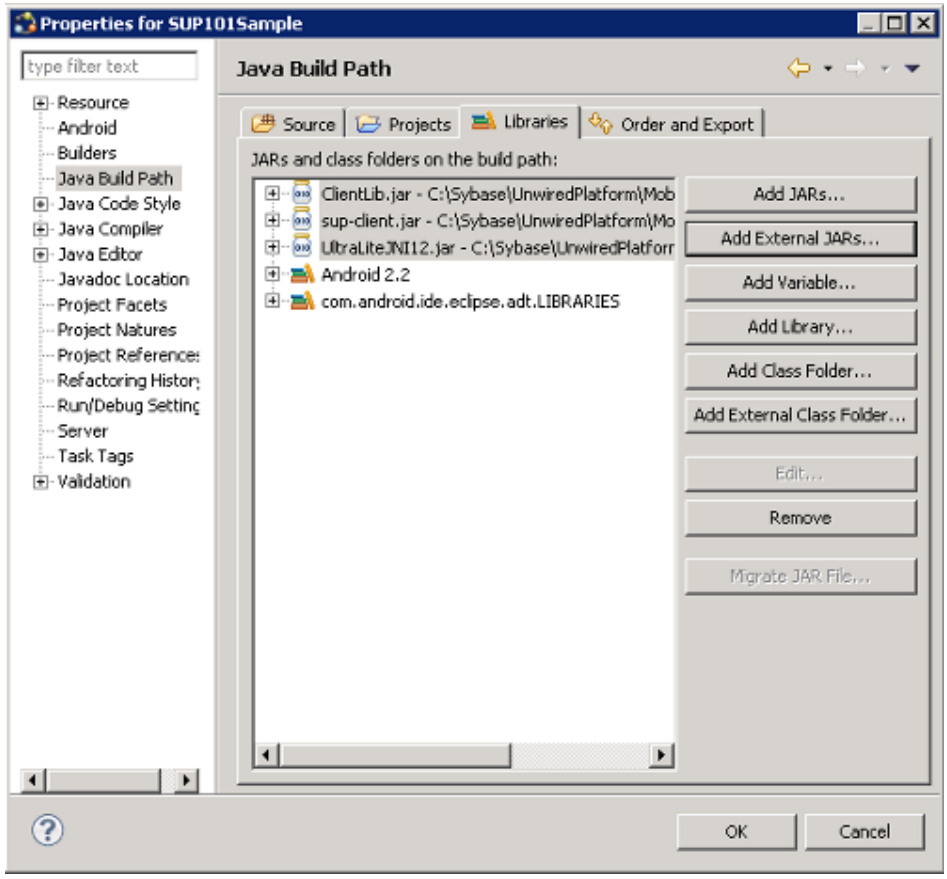


5. In the Select Build Target window, click an Android version 2.2 or higher and **Next.**
6. In the Application Info window, use these values and click **Finish.**
   - Application Name - verify **SUP101Sample** is entered.
   - Package Name - enter `com.sybase.sup.samples.objectapi`.
   - Verify **Create Activity** checkbox is selected and `SUP101SampleActivity` is entered.

   In the left-hand pane, you should see the Package Explorer with the SUP101Sample project listed.

   A Default Activity class is automatically generated.
7. In the 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. In the Package Explorer, modify the Java Compiler compliance level:
   a) Select the **SUP101Sample** project.
   b) Select **Project** > **Properties** > **Java Compiler**.
   c) Select the **Enable project specific settings** checkbox.
   d) From the **Compiler compliance level** list, select **1.6**.
   e) Click **OK**.
9. Add a compiler resource to the root directory of your Android project:
   a) In Windows Explorer, browse to the `C:\Sybase\UnwiredPlatform\MobileSDK\ObjectAPI\Android` directory.
   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.
10. In the 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 <manifest>.

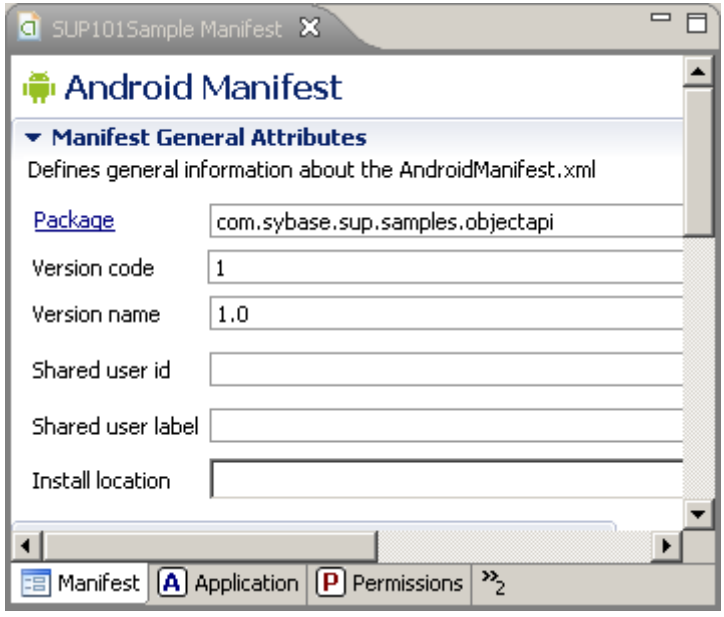
```
<uses-permission
android:name="android.permission.INTERNET">
  </uses-permission>
<uses-permission
android:name="android.permission.READ_PHONE_STATE">
  </uses-permission>
```

e) Select **File** > **Save**.

Configuring Android Application Properties

Indicate the general Android properties used in this application.

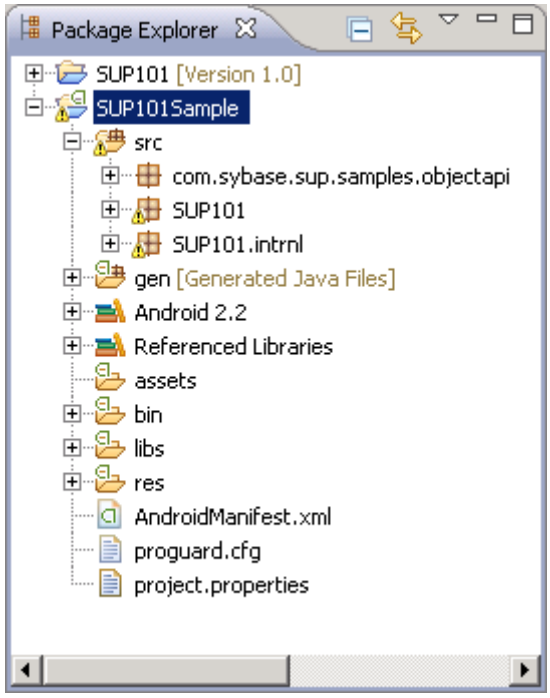
1. In the Package Explorer, expand the **SUP101Sample** project.
2. Double-click the `AndroidManifest.xml` file.
3. Select the **Manifest** tab.
4. (Optional) You can change, for example, the **Version** and other variables.
5. Click **File** > **Save** to save the SUP101Sample manifest file.



Copying Unwired Platform Files to Sample Project

Copy generated object API code to the SUP101Sample project.

1. In Windows Explorer go to `<UnwiredPlatform_InstallDir>\MobileSDK\ObjectAPI\Utils\genfiles\java\src` or the output directory that you specified when you ran `codegen.bat`, and copy the generated code files.
2. In Package Explorer, paste the folder into the `src` directory in the `SUP101Sample` project folder. You will see the `SUP101` and `SUP101.intrnl` folders.



Modifying the Android Manifest File

Add Detail Activity to the `AndroidManifest.xml` file.

1. In the Package Explorer, double-click the `AndroidManifest.xml` file.
2. Select the **AndroidManifest.xml** tab.
3. Add the following values to the `AndroidManifest.xml` file:

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

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".SUP101SampleActivity" >
            <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 to create the user interface for the SUP101Sample application.

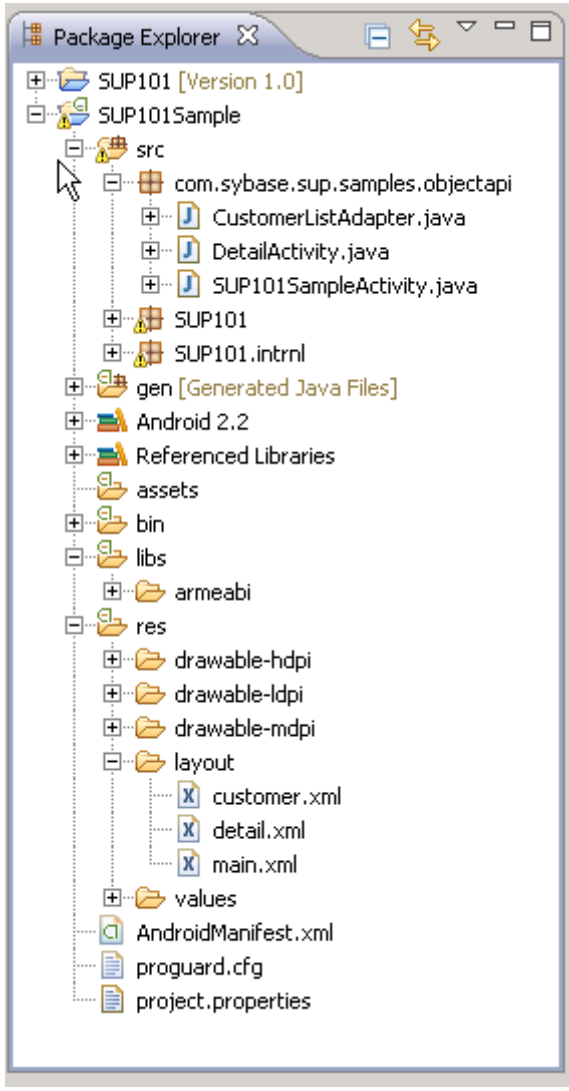
Prerequisites

Obtain the text files that contain the code snippets you need to build the user interface from the `SUP_Android_Custom_Dev_Tutorial_code.zip` file. The code snippets are used to create these classes: `CustomerListAdapter`, `DetailActivity`, and `SUP101SampleActivity`.

- 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/nav/summary.do?prod=1289*. Click the link for the Sybase Unwired Platform version that you want. Then, navigate to this topic in the tutorial, and click the link for the zip file to access the text files.
- When you created a new Android project at the beginning of the tutorial, a Default Activity class was automatically generated.

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, paste the copied Java files into the `src \com.sybase.sup.samples.objectapi` directory in the `SUP101Sample` project folder. Choose to copy over any existing files.
4. If you installed the Sybase Unwired Platform server on a system other than the localhost, you need to 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. Paste the copied XML files into the `res\layout` directory in the `SUP101Sample` project folder. Choose to copy over any existing files.



Creating a Launch Configuration for the Project

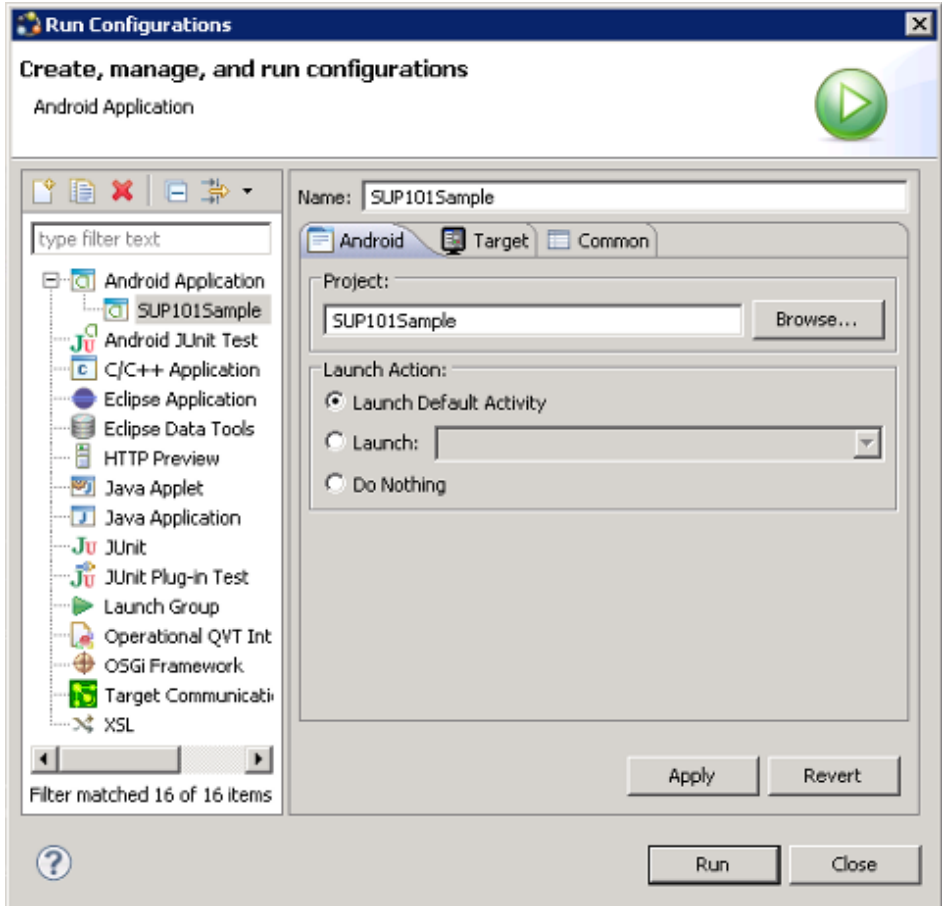
Create and configure a new launch configuration for the SUP101Sample project.

Prerequisites

Use the AVD Manager to create a target 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 Applications** and select **New**.
3. Enter the **Name:** `SUP101Sample`.
4. In the Android tab, click **Browse** and select **SUP101Sample**. Click **OK**.
5. Select a **Launch Action**.

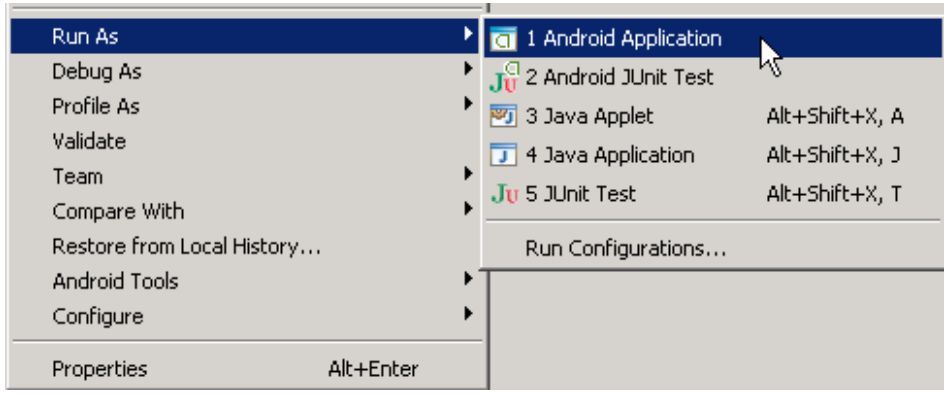


6. In the **Target** tab, select a **Deployment Target**.
   For example, select **Automatic** and an **AVD**.
7. Click **Apply**, then **Close**.

Testing the Device Application on the Android Emulator

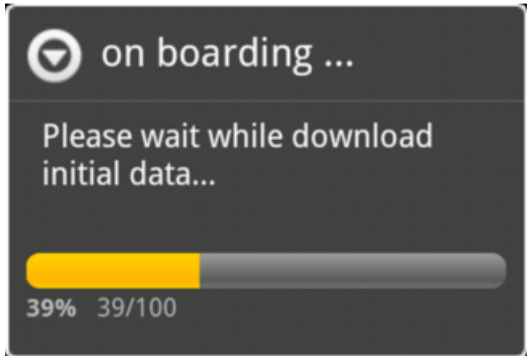
Run and test the SUP101Sample application on the Android Emulator.

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



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 operation to target change notifications is enabled using:

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

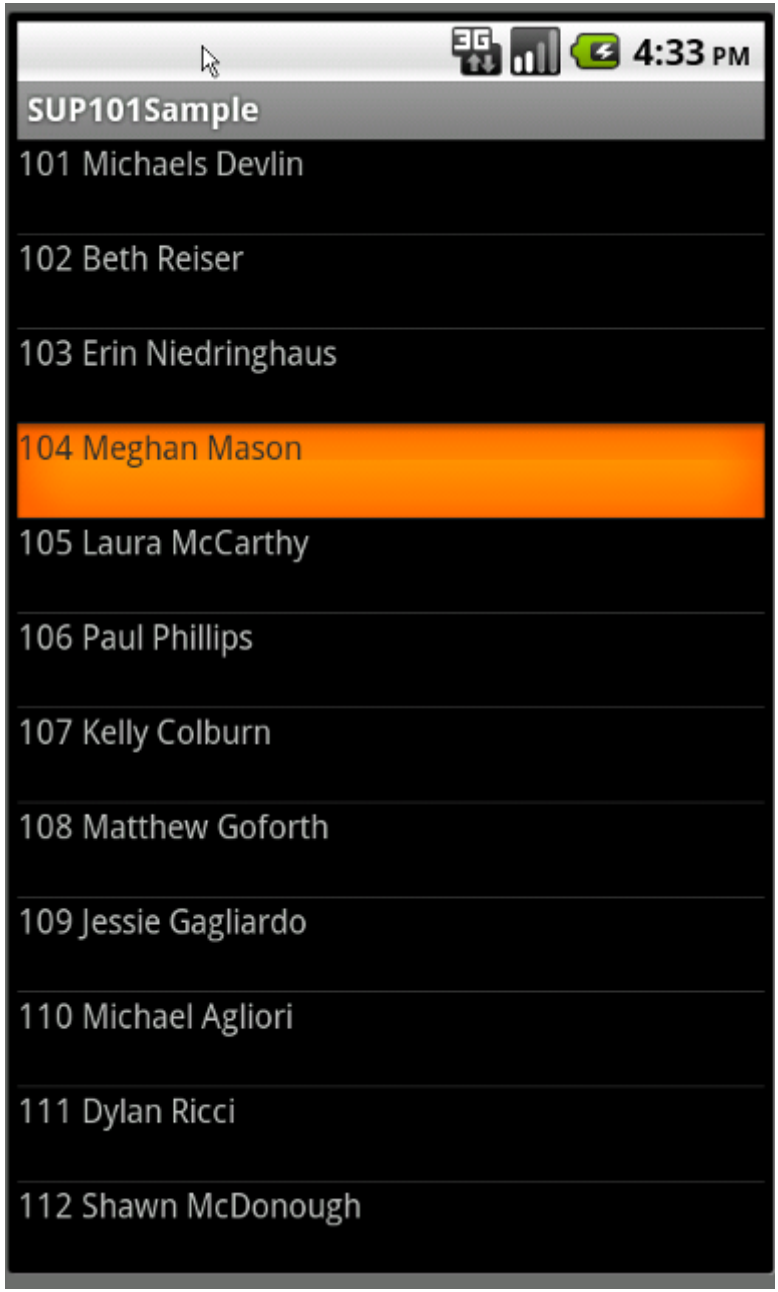
When the data finishes synchronizing, the device application shows the SUP101Sample Application with a list of customer data in ListView control. You can scroll through the

customer list to see more data and to make changes. The data loads from the database on demand.

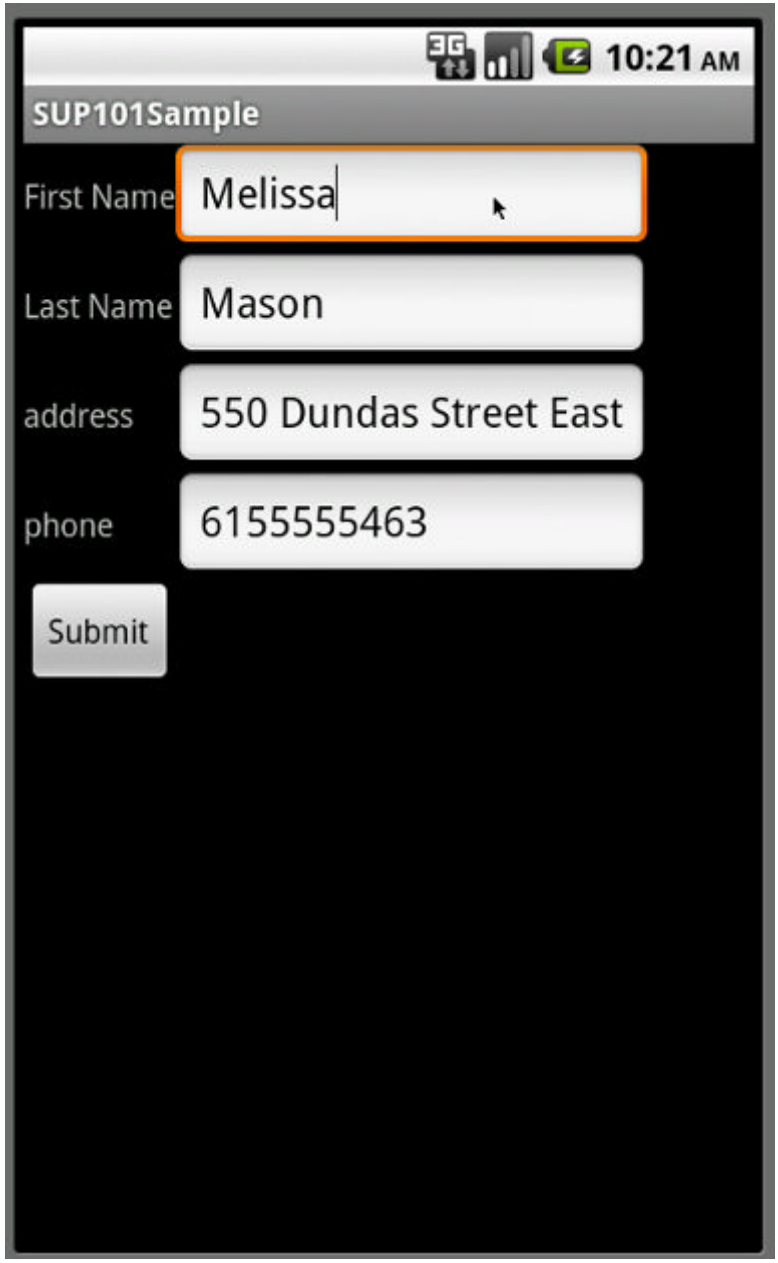
Note: The sample application uses a small buffer (30 customers). In commercial applications, you can use a large buffer (1,000 customers) based on user data.

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 whole customer object which results in better performance.

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

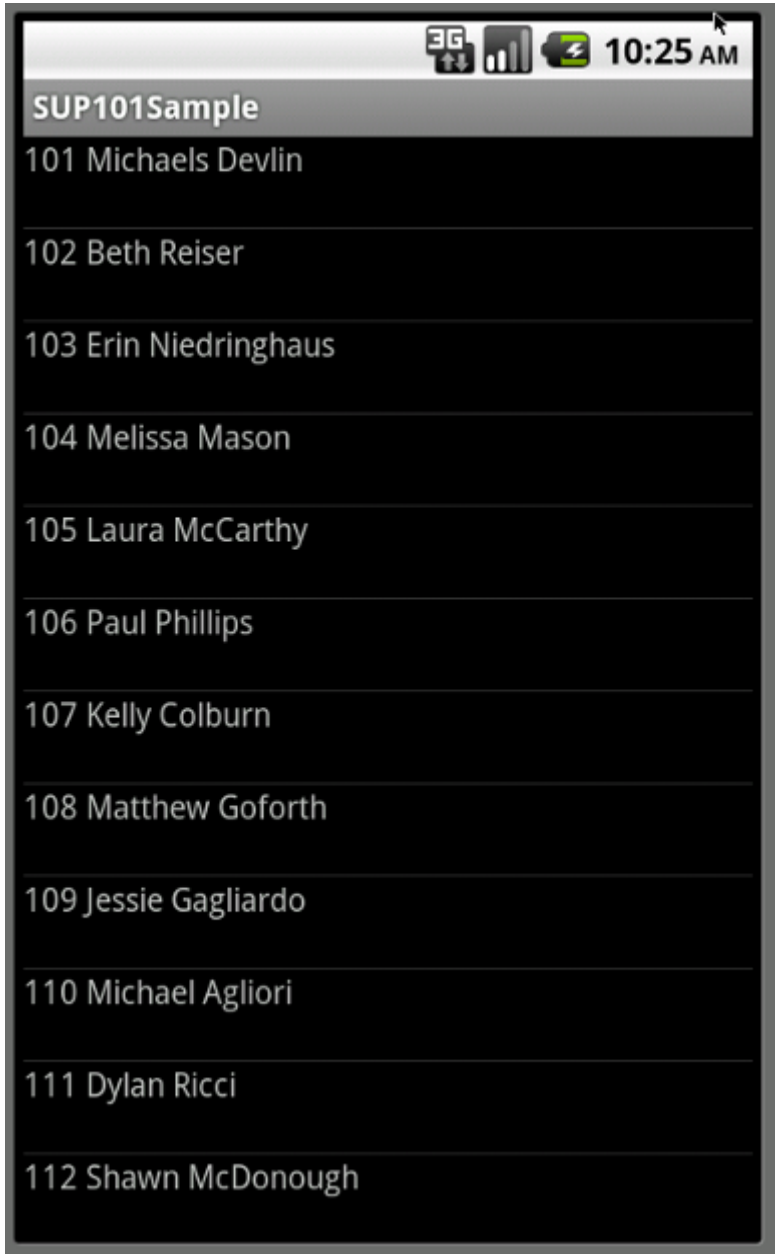


3. In the customer detail screen, change the first name of the customer and click **Submit**.



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 initiates Notifications from the server. The device application uses a ChangeLogAPI, specifically **GenericList<ChangeLog>changeLogs=SUP101DB.getChangeLogs(query);**, to query those managed items and use them to update the user interface if needed.



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: access *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.

Tutorial Projects

Tutorial projects are available for download, if you want the finished tutorial without going through the steps. Download tutorial 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 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:

- *Developer Guide: Unwired Server Management API* – customize and automate system administration features.
- *Developer Guide: Unwired Server* – customize and automate server-side implementations for device applications, and administration, such as data handling.

Javadoc and HeaderDoc are also available in the installation directory.

Index

L

M

O

P

R

S

T

U

V

W

X