SYBASE®
An SAP® Company

Tutorial: Android Object API Application Development

Sybase Unwired Platform 2.1 ESD #3

DOCUMENT ID: DC01734-01-0213-01
LAST REVISED: May 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.

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*

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.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 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.
Enterprise Explorer view	A view that provides functionality to connect to various 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 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 datasource, when creating an MBO. For example, you may want to model your MBOs before creating the datasources to which they bind. This MBO development 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 attributes automatically based on the datasource artifact 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 active file and lists structural 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 messages to adjust MBO properties and configurations to avoid problems, and use as a valuable source for collecting troubleshooting information when reporting issues to Customer Service and Support.
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 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**.
7. 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

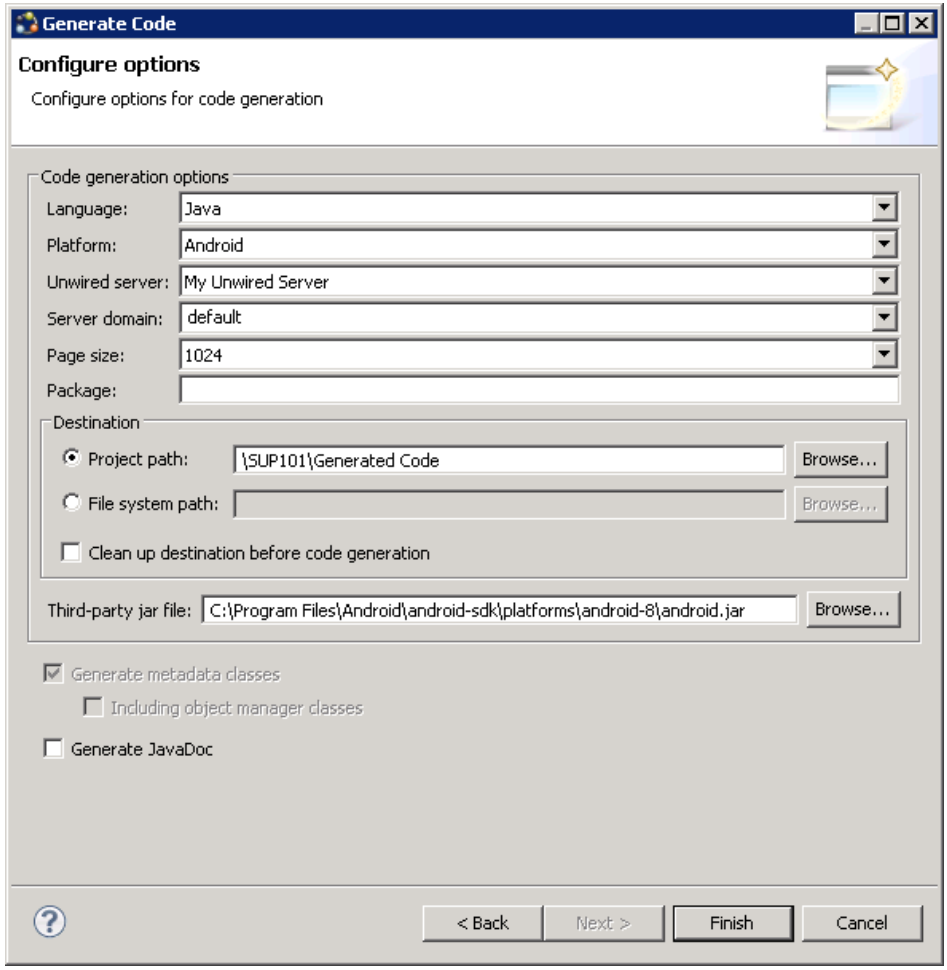
1. In Unwired WorkSpace, open the **SUP101** mobile application project.

   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.



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. Add library resources and set other application properties.

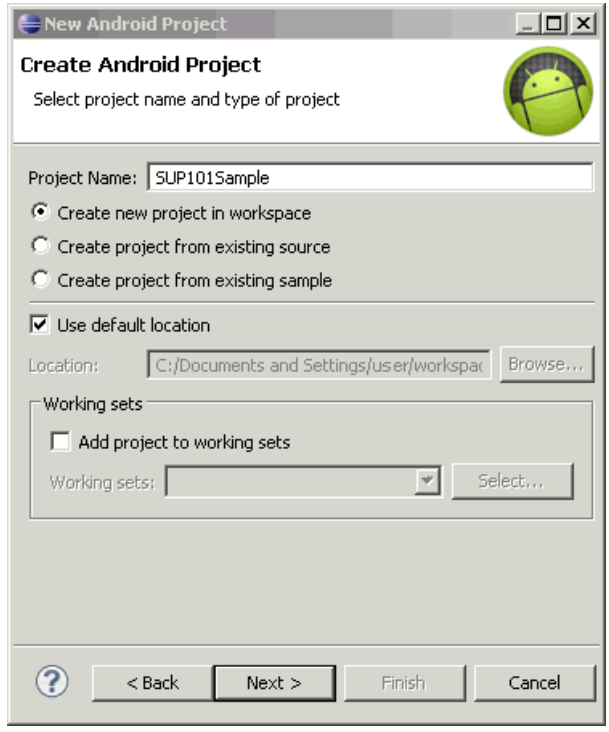
Prerequisites

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.

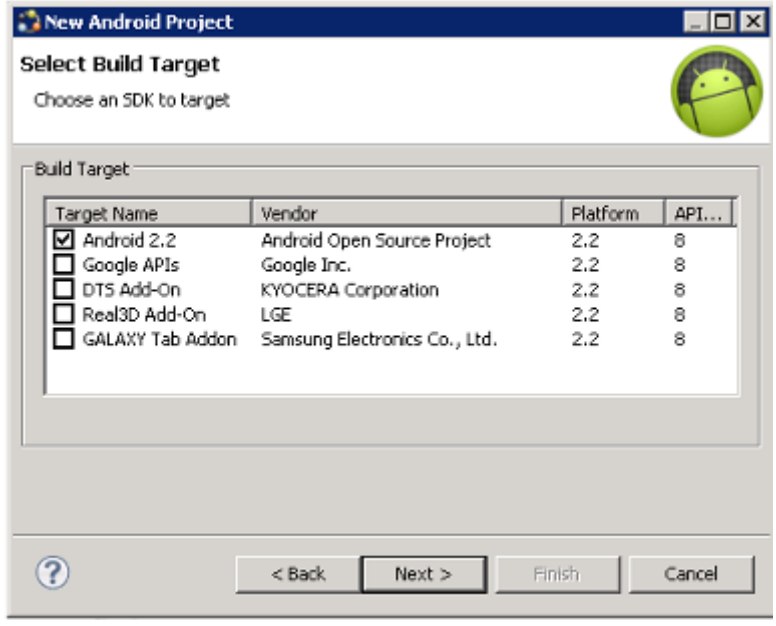
- If you are viewing this guide online from the Sybase Product Documention Web site, click *SUP_Android_Custom_Dev_Tutorial_code.zip* to access the ZIP archive containing the text files.
- If you are viewing this guide as a PDF, go to the Sybase Product Documentation Web site at *http://sybooks.sybase.com/sybooks/sybooks.xhtml* . 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**.
4. 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.



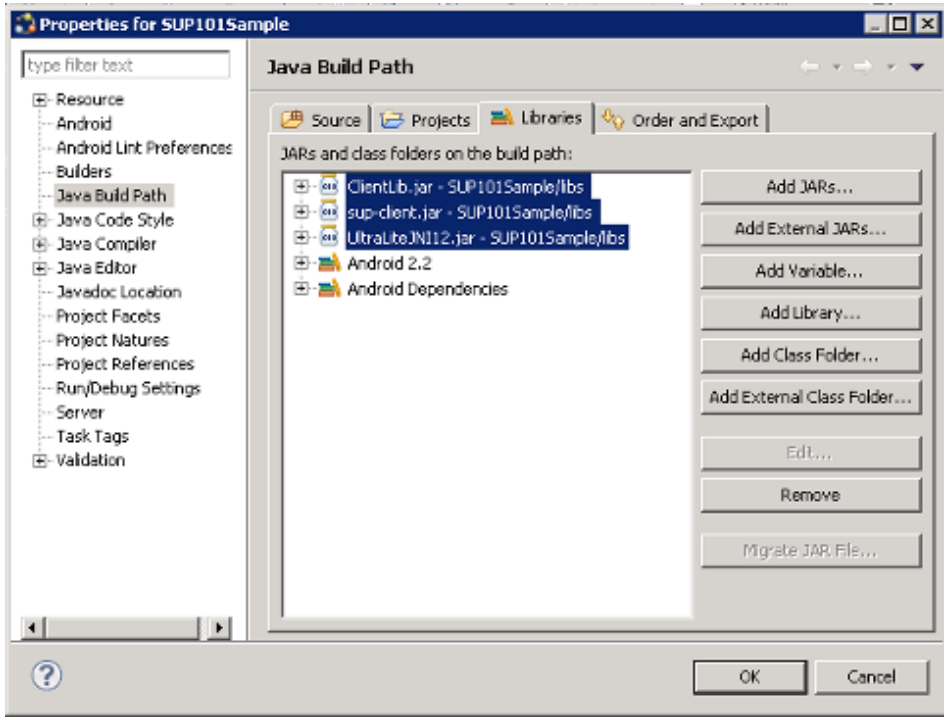
6. In the Select Build Target window, click an Android version 2.2 or later and **Next**.



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.

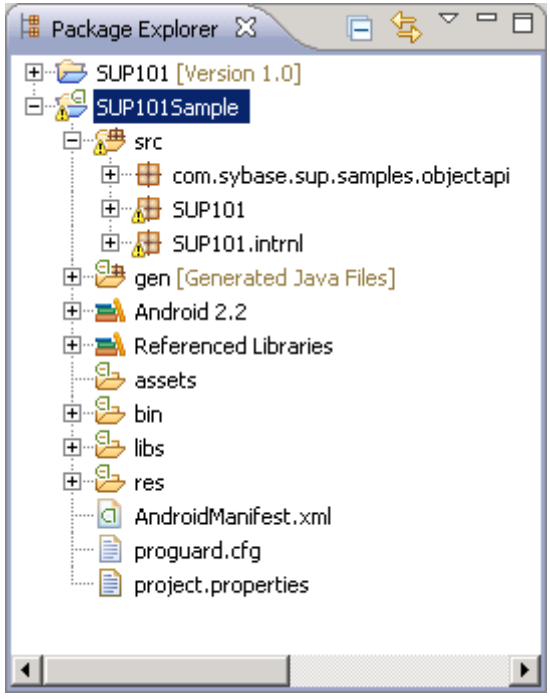
   Tip: To correct a mispelled package name, right-click the package and select **Refactor** > **Rename** to change the name and update all references.
8. Add a compiler resource and library resources to the root directory of the Android project:
   a) In Package Explorer, select **SUP101Sample** and add 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.



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.

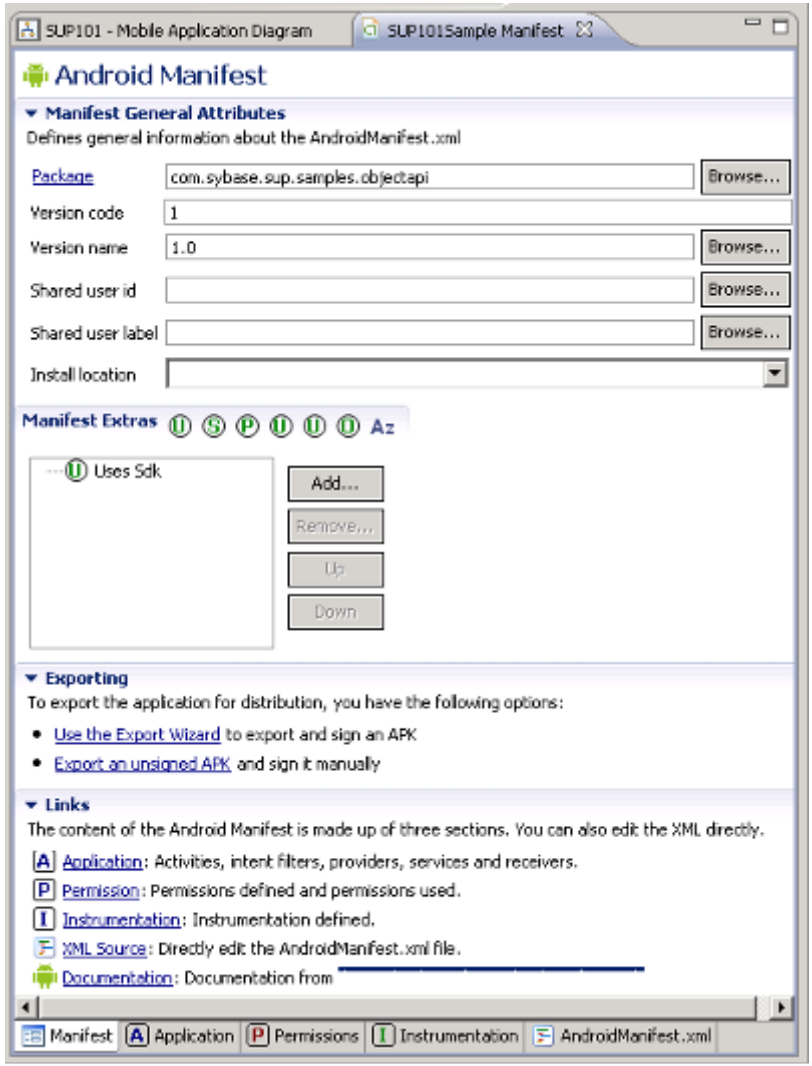


Configuring Android Application Properties

(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.



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 **AndroidManifest.xml** tab, add these values to the `AndroidManifest.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.

   ```
   <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>
   ```

2. Select **File** > **Save**.
   The XML file should look like this:

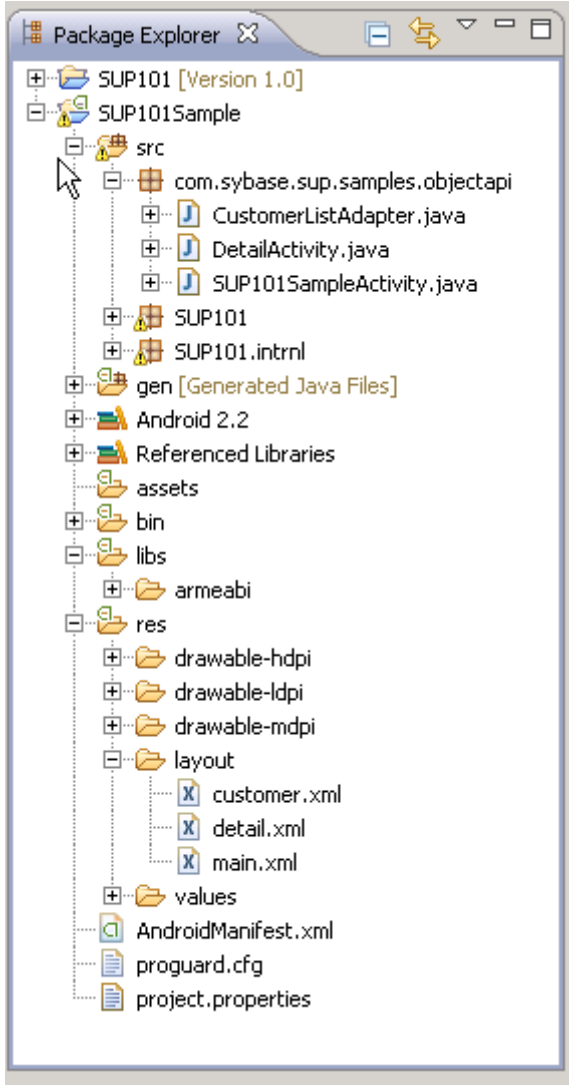
   ```
   <?xml version="1.0" encoding="utf-8"?>
   <manifest xmlns:android="http://schemas.android.com/apk/res/
   android"
       package="com.sybase.sup.samples.objectapi"
       android:versionCode="1"
       android:versionName="1.0" >

       <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 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, 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. 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

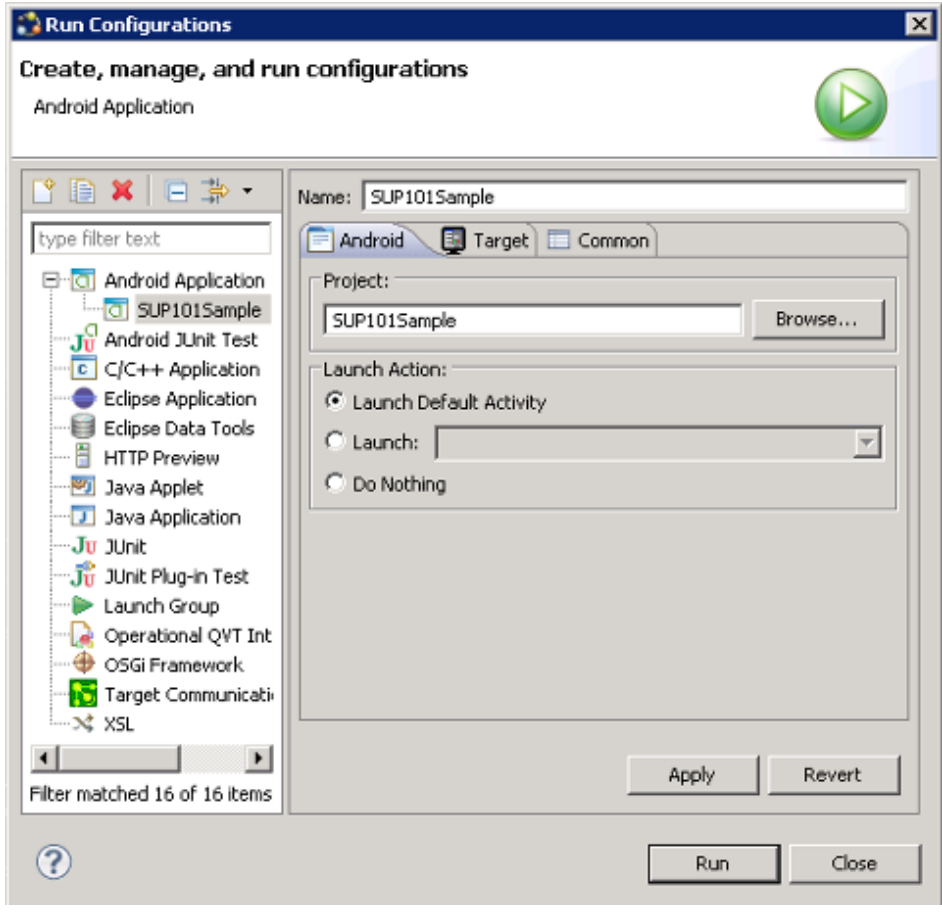
Create and define a new launch configuration for the SUP101Sample project. The configuration defines how the application launches and the target Android platform.

Prerequisites

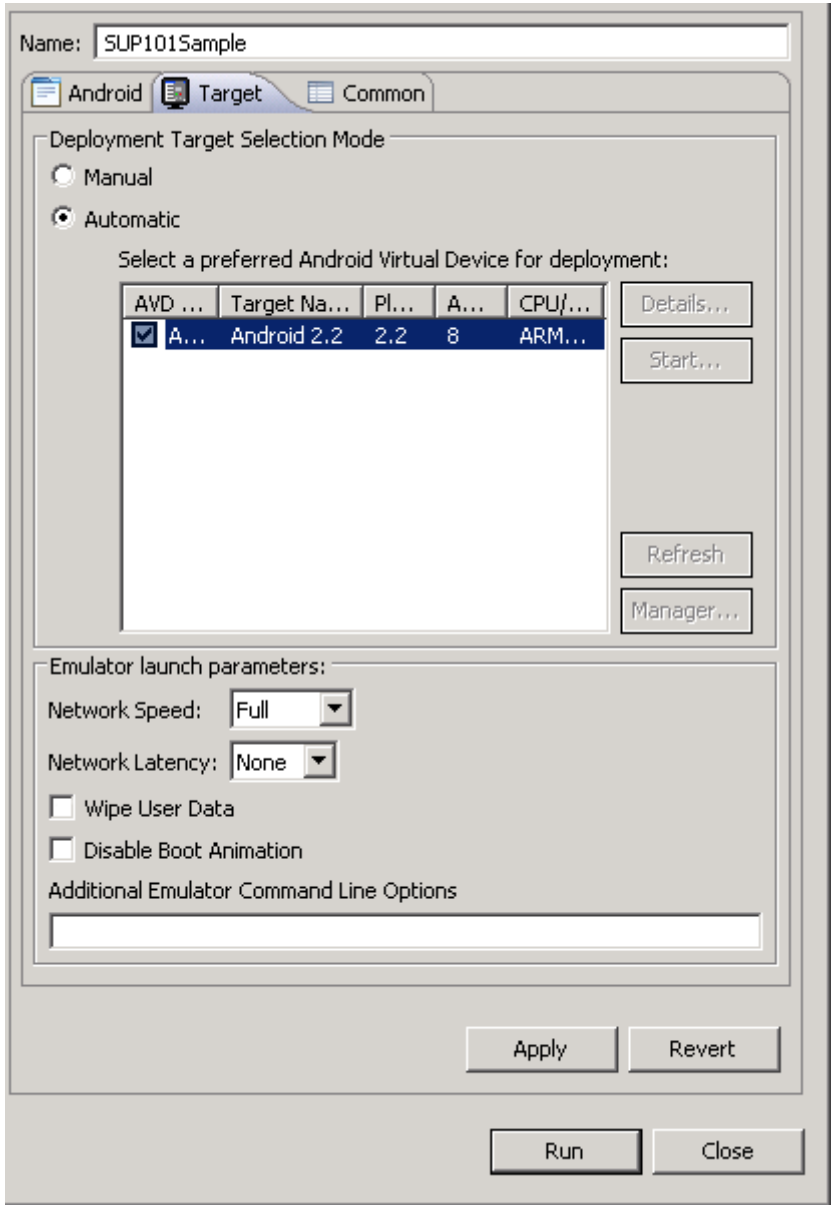
In the Unwired WorkSpace Window menu, use the AVD Manager to add a new target Android Virtual Device (AVD) for the launch configuration.

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**.



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

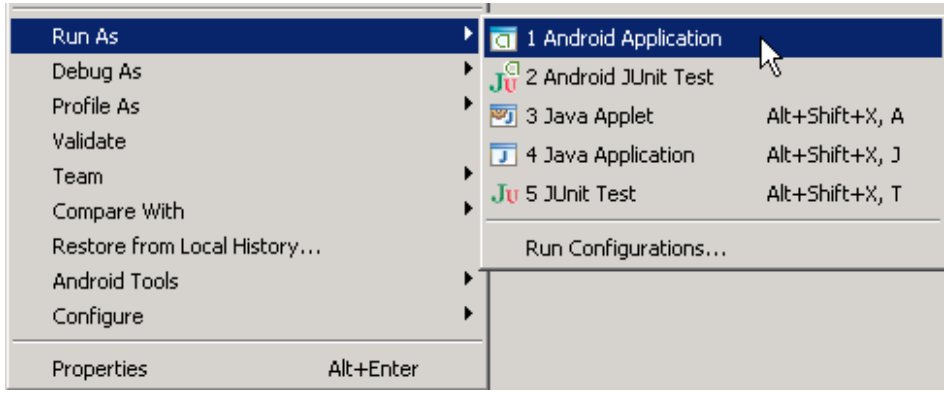


8. Click **Apply**, then **Close**.

Testing the Device Application on the Android Emulator

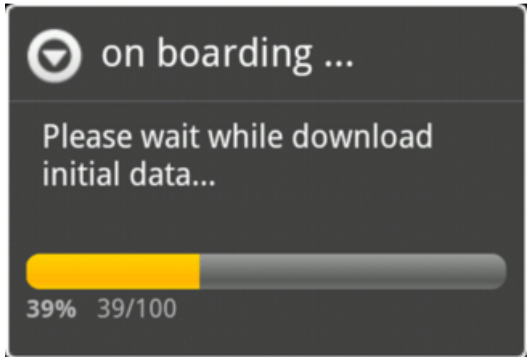
Run the SUP101Sample application on the Android emulator, and change customer information to update the interface.

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 Unwired Server in the background.



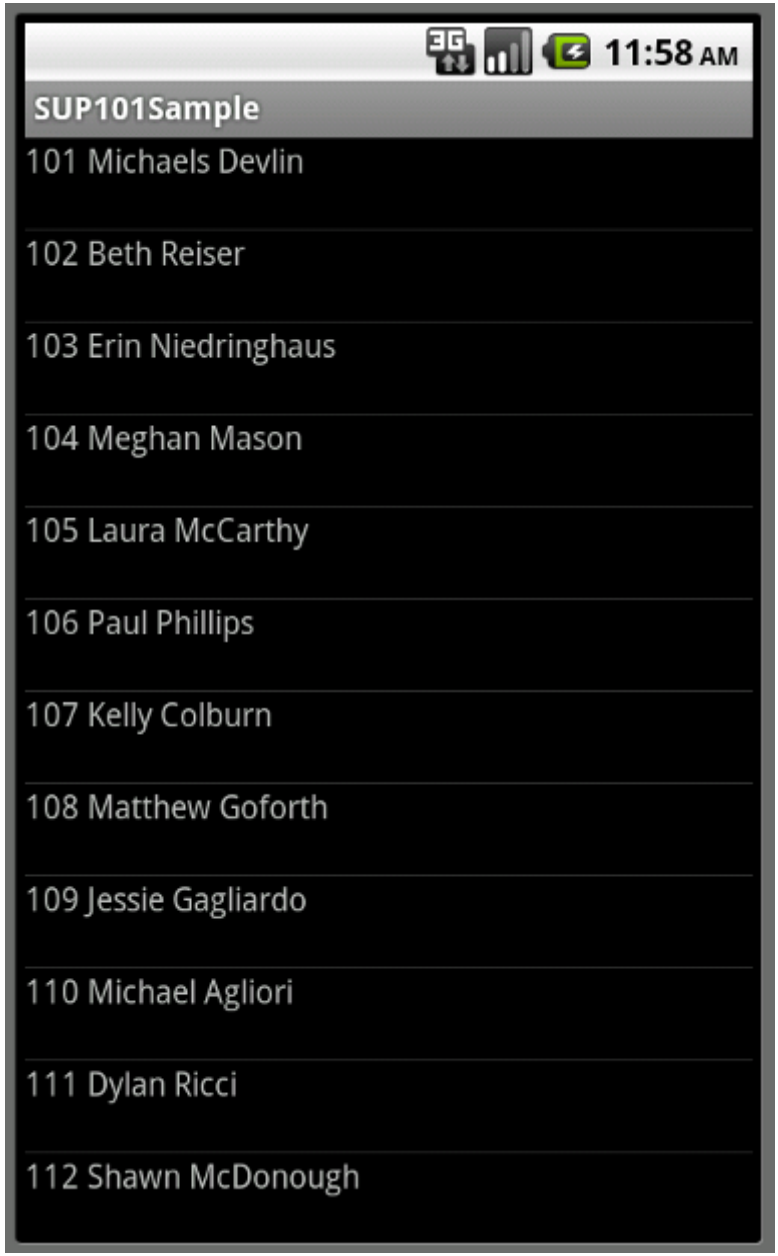
In the initialization process, the system enables the operation to target change notifications using:

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

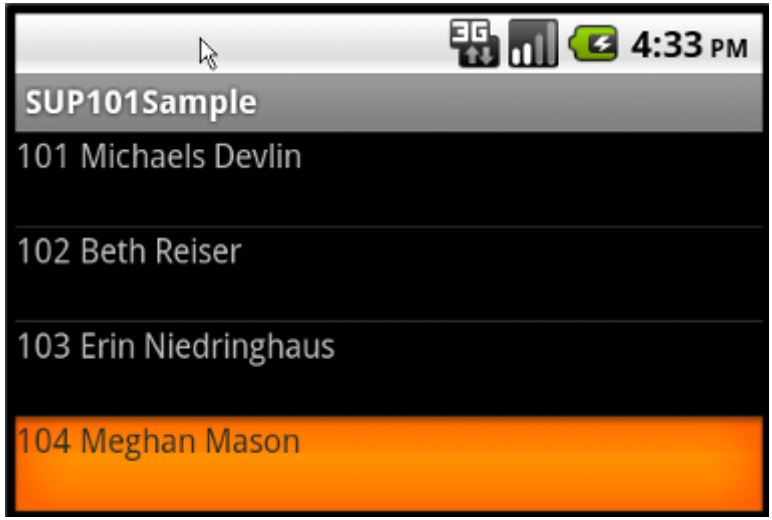
When the data finishes synchronizing, the device application shows the SUP101Sample Application with a list of customer data in a ListView control. You can scroll through the customer list to see more data and to make changes. The data loads from the database on demand.

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.

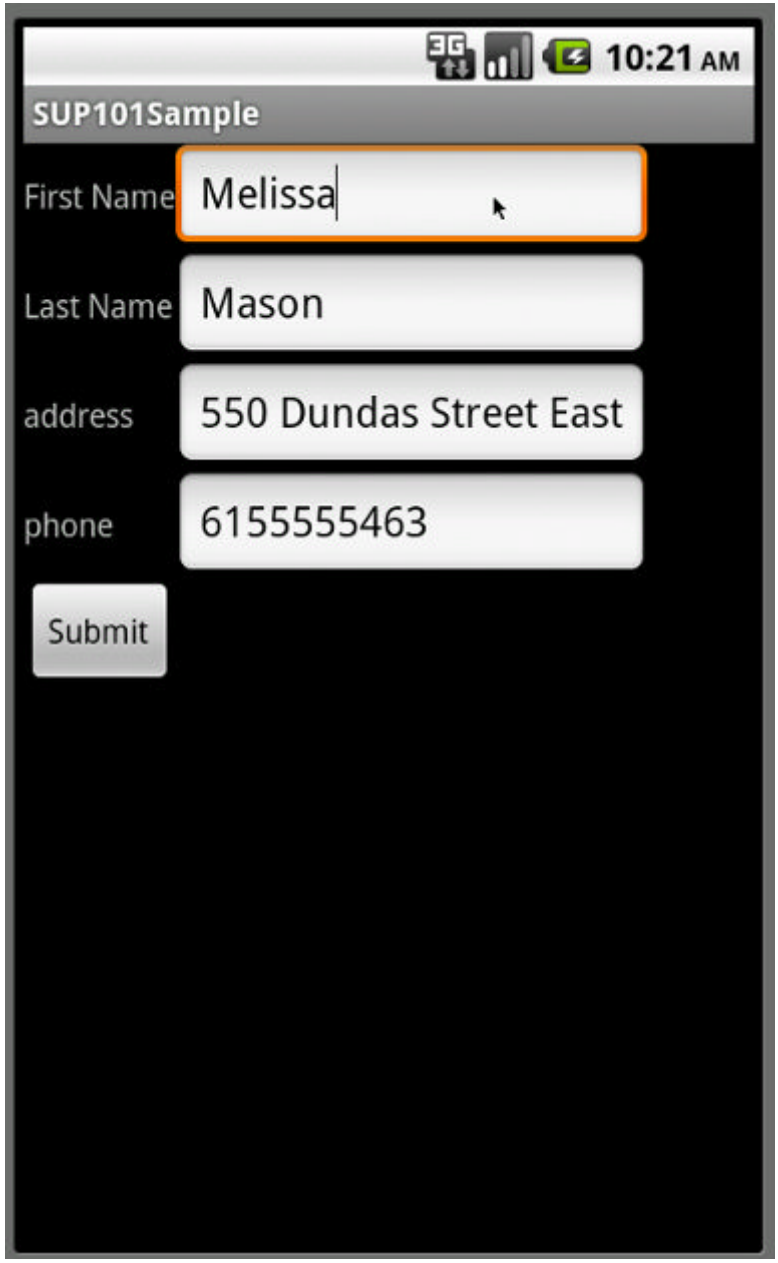
When the application queries the customer list, it uses a `SUP101DB.executeQuery()` API to get only columns that are needed, such as (fname, lname...), instead of the entire customer object; this results in better performance.



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 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);
```

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/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

T

U

V

X