SYBASE®
An SAP® Company

Tutorial: Android Object API Application Development

Sybase Unwired Platform 2.2 SP03

DOCUMENT ID: DC01734-01-0223-01
LAST REVISED: April 2013
Copyright © 2013 by Sybase, Inc. All rights reserved.
This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.
Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.
Sybase trademarks can be viewed at the Sybase trademarks page at *http://www.sybase.com/detail?id=1011207*. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.
Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.
Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.
All other company and product names mentioned may be trademarks of the respective companies with which they are associated.
Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.
Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

Sybase Unwired Platform Tutorials 1
Getting Started with Unwired Platform 3
Installing Sybase Unwired Platform 3
Starting Sybase Unwired Platform Services 3
Starting Sybase Unwired WorkSpace 4
Connecting to Sybase Control Center 4
Learning Sybase Unwired WorkSpace Basics 5
Developing an Android Application 9
Installing the Android SDK 9
Installing ADT in Sybase Unwired WorkSpace 10
Generating Java Object API Code 11
Creating the Android Project 13
Adding Compiler and Library Resources 18
Copying Unwired Platform Files to Sample Project 19
Configuring Android Application Properties 19
Creating the User Interface 22
Creating a Launch Configuration for the Project 24
Testing the Device Application on the Android Emulator 26
Learn More About Sybase Unwired Platform 33
Index 35

Sybase Unwired Platform Tutorials

The Sybase® tutorials demonstrate how to develop, deploy, and test mobile business objects, device applications, online mobile applications (native OData and REST services based), and Hybrid App packages. You can also use the tutorials to demonstrate system functionality and train users.

- 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*

  Note: For all Object API tutorials, if you opt to use the Mobile Business Object example project instead of performing the Mobile Business Object Tutorial, you must deploy the mobile application project to Unwired Server as a prerequisite.
- 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 Object API Application Development*
  - *Tutorial: Windows Mobile Object API Application Development*
- Create a mobile business object, then develop a hybrid app package that uses it:
  - *Tutorial: Hybrid App Package Development*
- Create an OData mobile application with REST Services
  - *Tutorial: Android OData Application Development with REST Services*
  - *Tutorial: iOS OData Application Development with REST Services*

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 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*:

- *Release Bulletin*
- *Installation Guide for Sybase Mobile SDK*
- *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 Apps, and native OData SDK applications.
   - Sybase Unwired WorkSpace, the Eclipse-based development environment for MBOs and Hybrid Apps.

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 > (All) Programs > Sybase > Unwired Platform > Start Unwired Platform Services**.
The following services will be started:

- Sybase Control Center *<Version>*
- Sybase Unwired Cache DB
- Sybase Unwired SampleDB
- Sybase Unwired Server

Unwired Platform 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 enterprise information system (EIS) datasources and Unwired Server connections, develop Hybrid App applications (Hybrid Apps), and generate Object API code.
Select **Start > Programs > Sybase > Unwired Platform > Sybase 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 Sybase Unwired WorkSpace concepts and tasks, select **Help > Help Contents**.

Connecting to Sybase Control Center

Open Sybase Control Center 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
- Assign/Unassign a hybrid application to a device

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

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

   Note: If 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 Sybase 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 Sybase 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.
  - To close the Welcome page, click **X**.
  - Reopen the Welcome page by selecting **Help > Welcome**.
  - To learn about tasks you must perform, select the **Development Process** icon.
- In Sybase 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 using that code and sample files, and test the application's functionality on an emulator.

Prerequisites

Note: This tutorial was created using Sybase Unwired Platform 2.2 SP02, Android SDK r21.0.0, ADT Plugin for Eclipse 21.0, and run on an Android 4.0.3 - API Level 15 target emulator. If you use a different version, some steps may vary.

1. Complete the tasks in *Getting Started with Unwired Platform*.
2. Either:
   - create the MBO project by completing *Tutorial: Mobile Business Object Development*, or
   - download and deploy the MBO SUP101 example project (complete project files) from the SAP® Community Network: *http://scn.sap.com/docs/DOC-8803*.

   Note: If you upgrade Sybase Mobile SDK after completing the tutorial, you can convert the project to the current SDK by importing the earlier project into the Unwired Workspace and then accepting the confirmation prompt.
3. (Optional) To use as a reference and copy source code when completing this tutorial, download the Android SUP 101 example project (source code only) from the SAP® Community Network: *http://scn.sap.com/docs/DOC-8803*.
4. Download the supported versions of the Android SDK and Android Development Tools (ADT).

   See the *Supported Hardware and Software* guide 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 mobile business objects that are deployed to Unwired Server.

Installing the Android SDK

Install the Android SDK.

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

See *Google Android Versions for Object API* 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 tools (SDK Tools and SDK Platform-tools) and the Android API.
4. Launch the **Android Virtual Device Manager**, and create an Android virtual device to use as your emulator.

Installing ADT in Sybase Unwired WorkSpace

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

See *Google Android Versions for Object API* at *http://sybooks.sybase.com/sybooks/sybooks.xhtml*. Navigate to the appropriate version of Sybase Unwired Platform.

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

   Note: If you see a security warning about the authenticity or validity of the software, click **OK**.
7. When the installation completes, restart Sybase Unwired WorkSpace.
8. For first-time installations:
   a) 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`.
   b) 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. Code generation creates the business logic, attributes, and operations for the mobile business objects (MBOs) 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 Sybase Unwired WorkSpace, open the **SUP101** mobile application project.

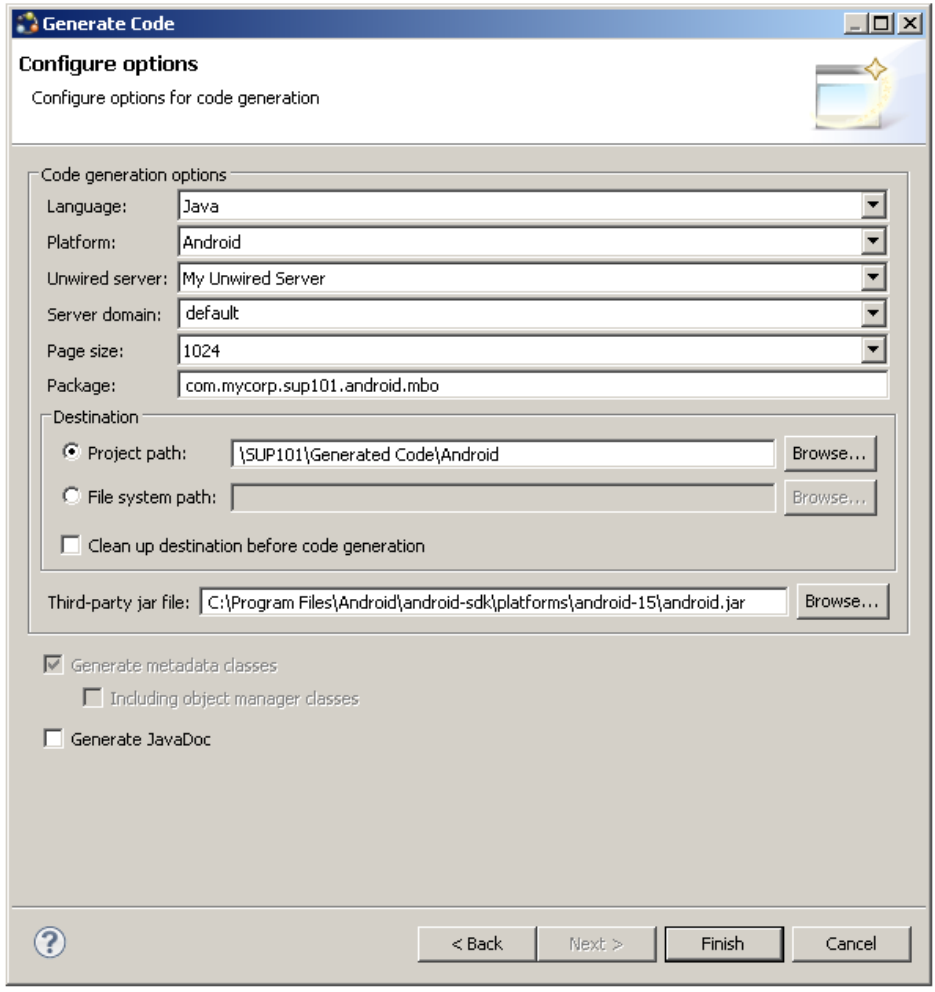
   In WorkSpace Navigator, right-click the **SUP101** folder and select **Open in Diagram Editor**.
2. In WorkSpace Navigator, expand **SUP101**. Under **Generated Code**, add a folder named `Android`.

   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**.

   Ignore any warning about unresolved mobile business object dependencies.
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	Enter `com.mycorp.sup101.android.mbo`.
Project path	Enter `\SUP101\Generated Code\Android`.
Third-party jar file	Click **Browse** to open an `android.jar`, by default located in `C:\Program Files\Android\android-sdk\platforms\android-`*xx*.
Generate JavaDoc	Unselect for this tutorial.



7. In the Success dialog, click **OK**.
   In the `Generated Code` directory, you see an `\Android\src\` folder.

Creating the Android Project

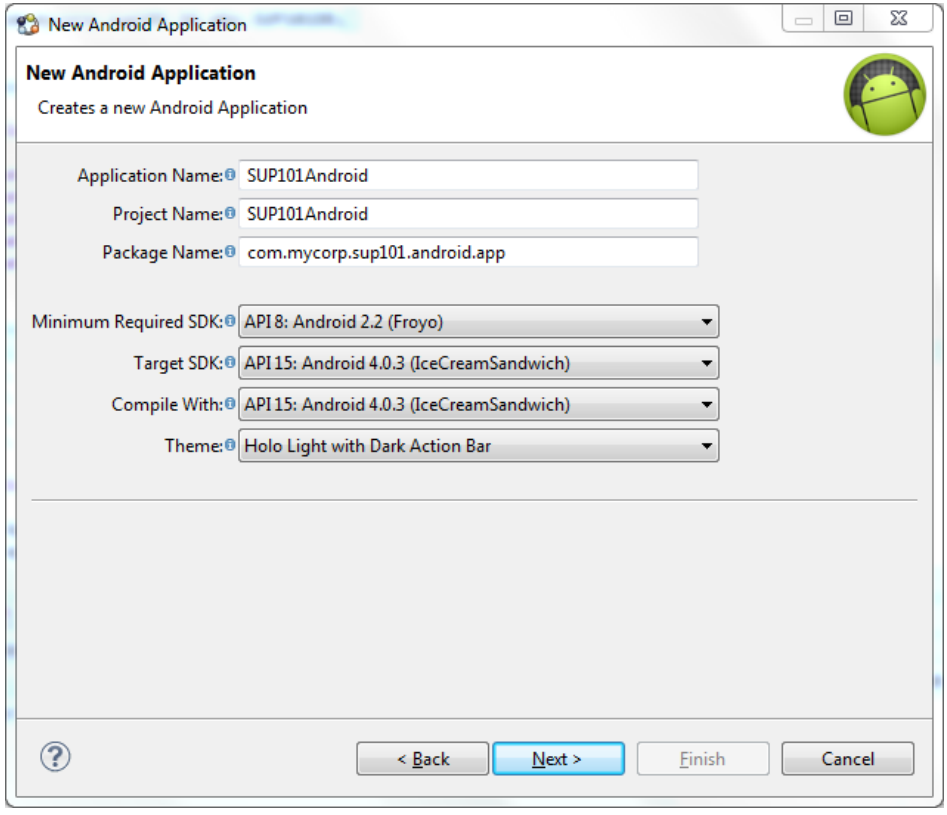
Create a new Android project in Sybase Unwired WorkSpace. Add library resources to the project and set other application properties.

Prerequisites

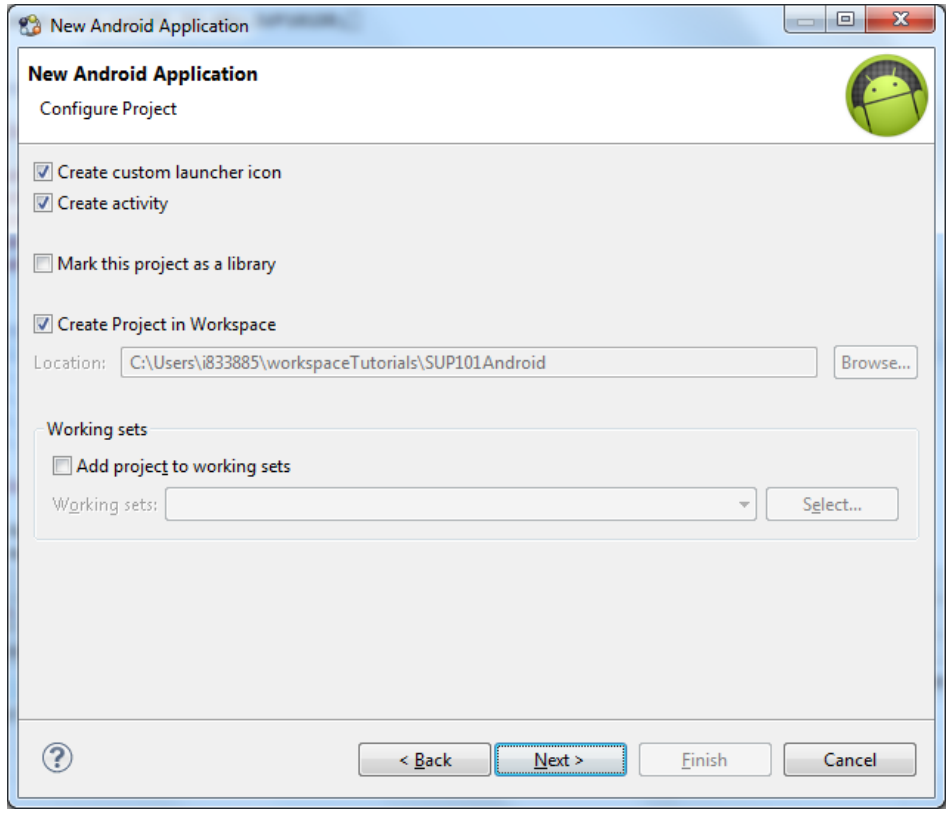
To help create your project—and in a subsequent topic, build the user interface—download the SUP101 Android Object API (2.2 SP02) example project from the SAP Community Network (SCN) Web site at *http://scn.sap.com/docs/DOC-8803*.

Task

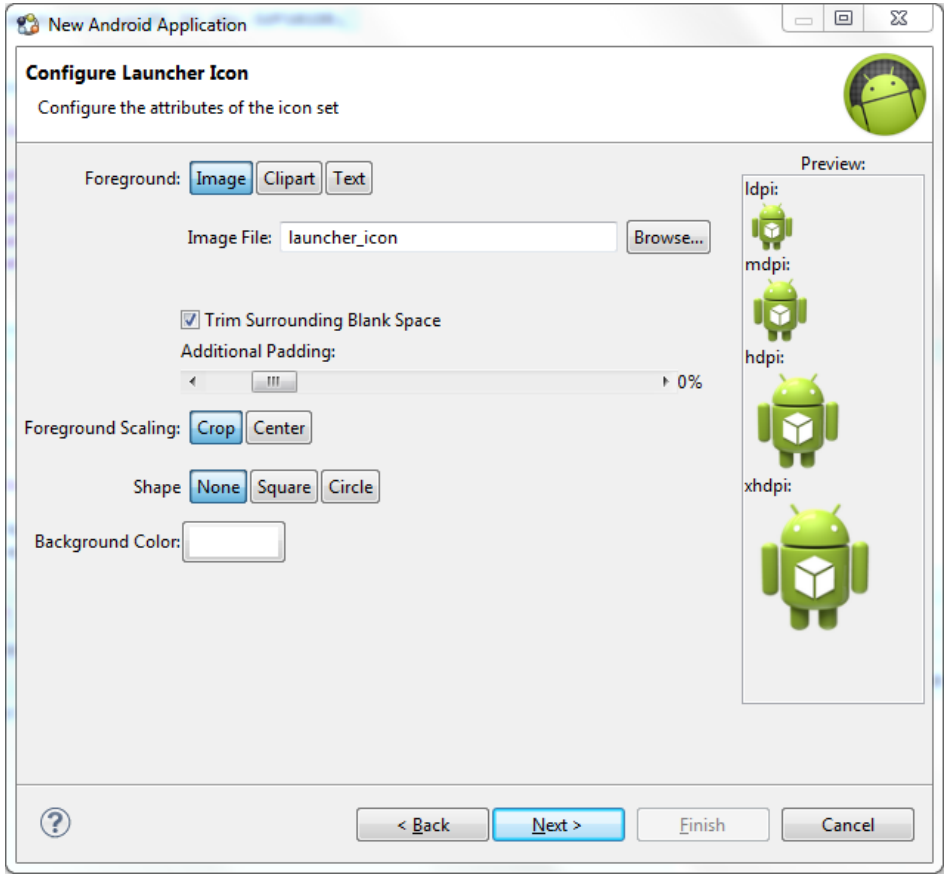
1. Start Sybase Unwired WorkSpace.
2. In Sybase Unwired WorkSpace Preferences, set the Android SDK location.
3. Select **File > New > Project**.
4. Select **Android** > **Android Application Project**, then **Next**.
   Depending on the Android version you are using, the information you provide in the next several steps may be in one or two screens.
5. In the New Android Application window, use these values and click **Next**.
   - Application Name – enter `SUP101Android`.
   - Project Name – enter `SUP101Android`.
   - Package Name – enter `com.mycorp.sup101.android.app`.
   - Minimum Required SDK – accept the default.
   - Target SDK – select the Android SDK used for the tutorial.
   - Compile With – select the Android SDK used for the tutorial.
   - Theme – accept the default.



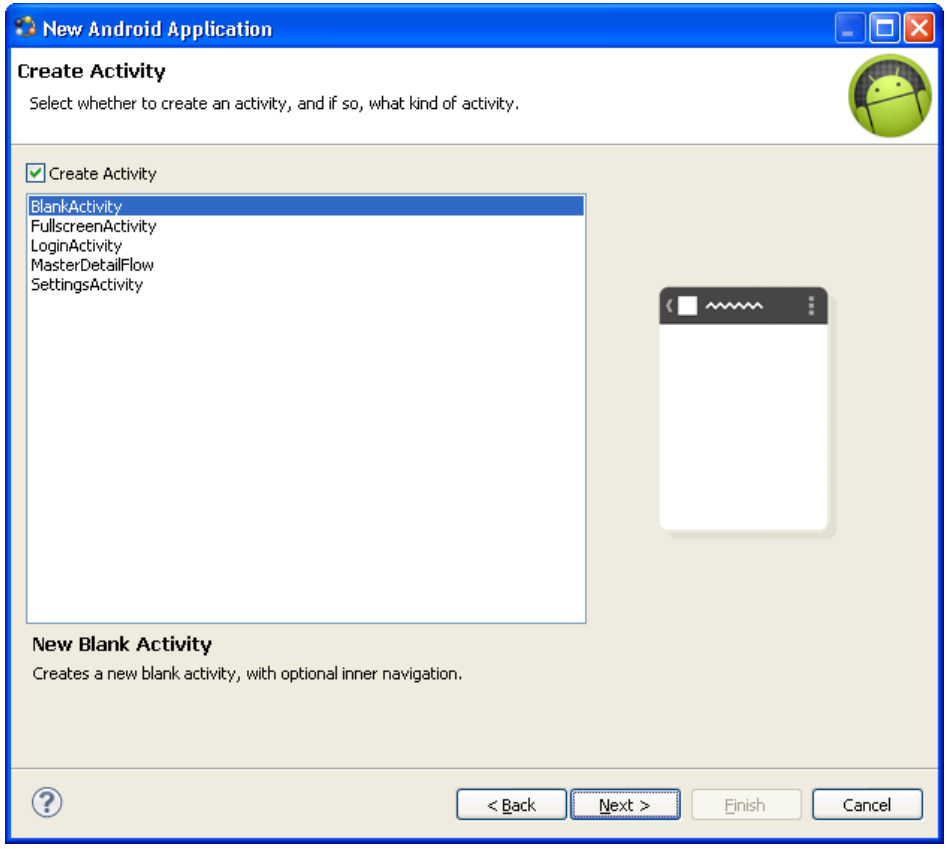
6. In the New Android Application window, use these values and click **Next**.
   - Select **Create custom launcher icon**, **Create activity**, and **Create Project in Workspace**.
   - Unselect **Mark this project as a library** and **Add project to working sets**.



7. In the Configure Launcher Icon window, accept the default settings and click **Next**.

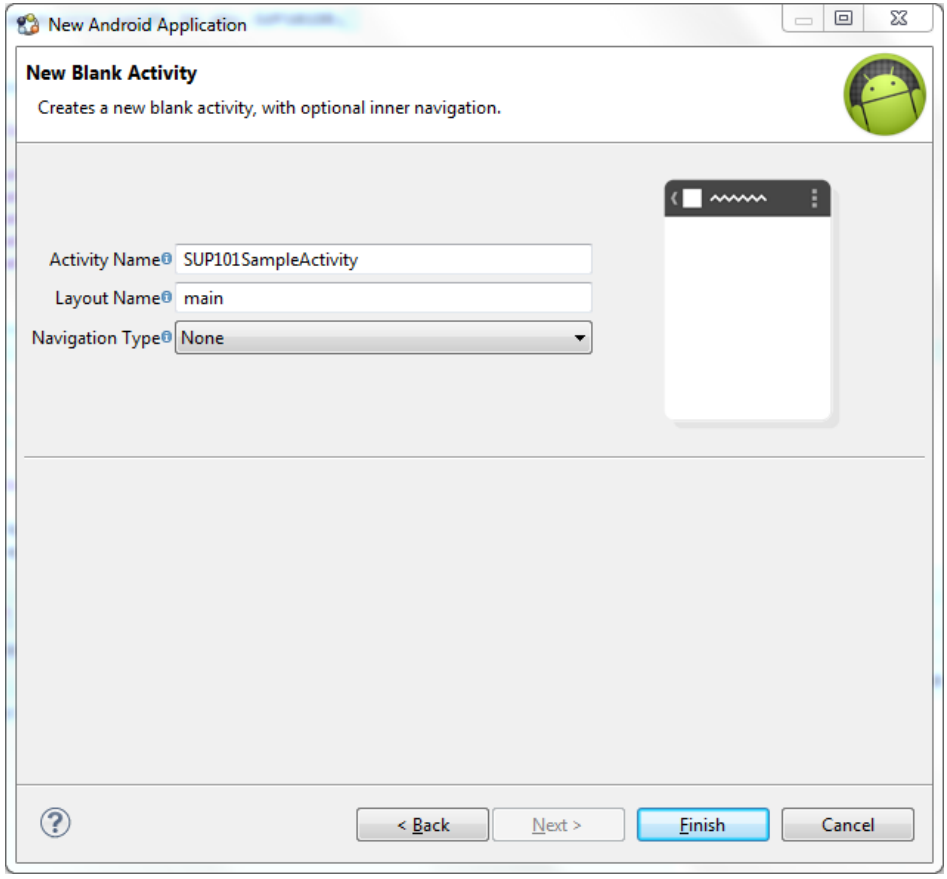


8. In the Create Activity window select **Create Activity**, then select **BlankActivity**, and click **Next**.



9. In the New Blank Activity window, use these values and click **Finish**.

- Activity Name – enter `SUP101SampleActivity`.
- Layout Name – enter `main`.
- Navigation Type – accept the default of `None`.



The left pane of the Workspace Navigator should list the SUP101Android project. In the `src` folder, a default Sample Activity class was automatically generated when you created the project.

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

Adding Compiler and Library Resources

Add compiler and library resources to the Android project.

1. Add a compiler resource to the root directory of the project:
   a) In Windows Explorer, browse to *SUP_HOME*`\MobileSDK22\ObjectAPI\Android` and copy the `armeabi` folder and these JAR files: `AfariaSLL.jar`, `ClientLib.jar`, `sup-client.jar`, and `UltraLiteJNI12.jar`.

b) In Workspace Navigator, expand **SUP101Android**, select the `libs` folder, and paste the `armeabi` folder and JAR files into it.

2. Add library resources to the project:
   a) In Workspace Navigator, right-click the **SUP101Android** project, click **Properties**, and select **Java Build Path**.
   b) Click the **Libraries** tab and select **Add JARs**.
   c) In the JAR Selection window, expand the `SUP101Android\libs` folder and select **AfariaSSL.jar**, **ClientLib.jar**, **sup-client.jar**, and **UltraLiteJNI12.jar**.
   d) Close the windows.

Copying Unwired Platform Files to Sample Project

Copy the object API code you generated using the Generate Code wizard for Android.

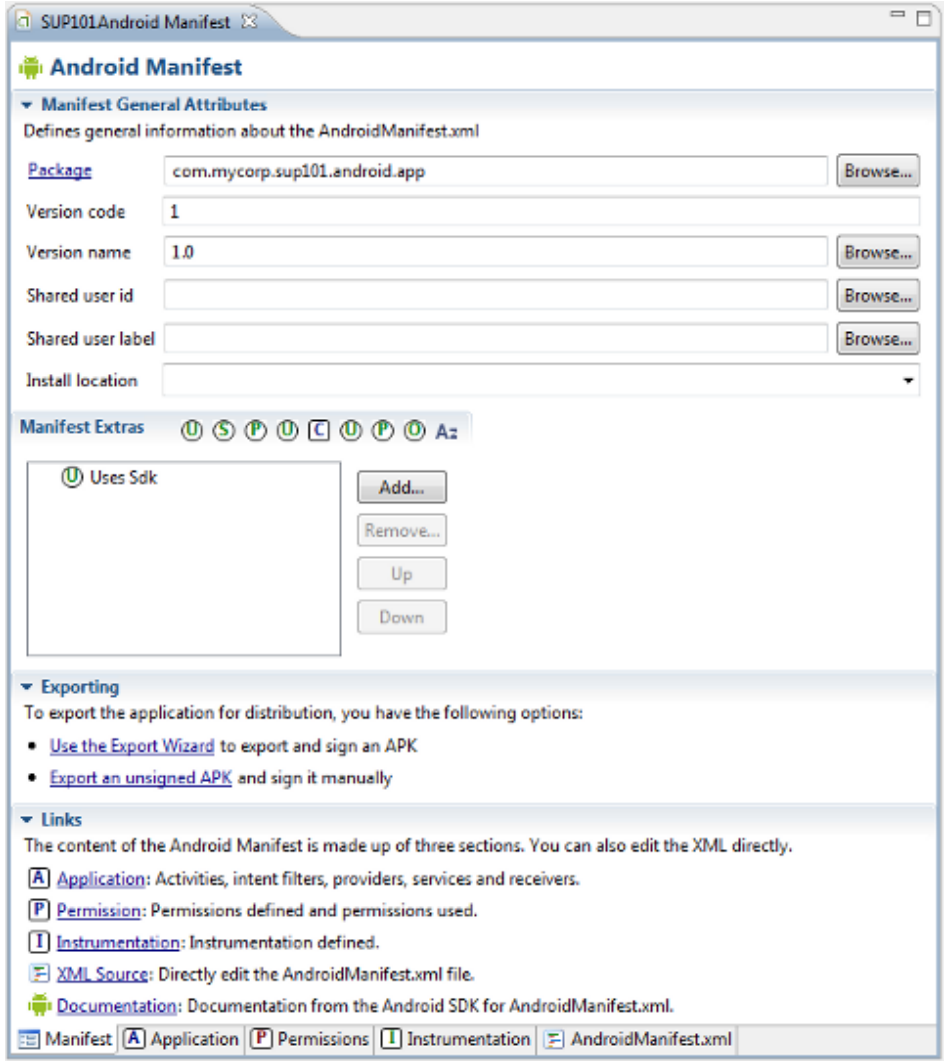
1. In Workspace Navigator, go to the **SUP101** project and copy the `com` folder in `\Generated Code\Android\src\`.
2. Go to the **SUP101Android** project and paste the `com` folder in to the `src` directory. Select **Yes to All** to copy over existing folders.

Configuring Android Application Properties

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

1. In Workspace Navigator, expand the **SUP101Android** 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 Android SDK version on which you want to run the application.



5. Select **File** > **Save**.

Next

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

Adding User Permissions and a Class to the Android Manifest File

Add user permissions to the Android project. Also add a Detail Activity class to the `AndroidManifest.xml` file. This declaration launches a customer detail screen where you can make changes when you test the application.

1. If needed, open the Android manifest file.
2. Select the **AndroidManifest.xml** tab.
3. Replace the code with the source code from the AndroidManifest.xml file you downloaded from the SAP Community Network (SCN) Web site, also provided below:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/
android"
    package="com.mycorp.sup101.android.app"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />
   <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.READ_PHONE_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            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>
```

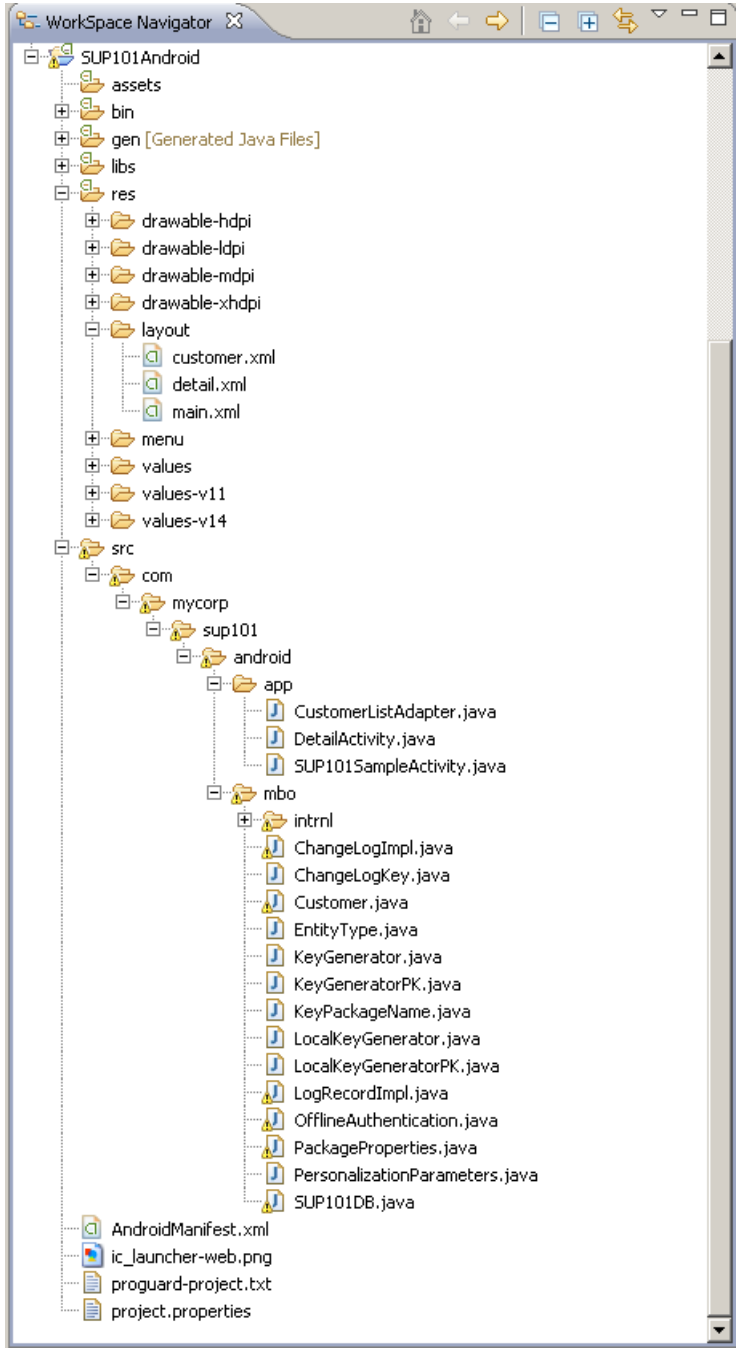
4. Select **File > Save**.

Creating the User Interface

Copy the Java code files, which provide the functionality and layout of the user interface, from the SUP101 Android Project example project archive to the SUP101Android project.

1. In Windows Explorer, browse to the directory where you saved the SUP101 Android Project example project.
2. Copy these Java files: `CustomerListAdapter.java`, `DetailActivity.java`, and `SUP101SampleActivity.java`.
3. In Workspace Navigator, go to **SUP101Android** and expand `\src\com\mycorp\sup101\android\app`, then paste the copied Java files, copying over any existing files.
4. Modify the host IP address in the `SUP101SampleActivity.java` file to point to the Unwired Server.
   a) In Workspace Navigator, expand the **SUP101Android** project.
   b) Under the `\src\com\mycorp\sup101\android\app` folder, double-click the `SUP101SampleActivity.java` file.
   c) Modify the host IP address, and verify the username and password are valid.
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 `SUP101Android` project folder, go to the `res\layout` directory and paste the copied XML files, copying over any existing files.

   The SUP101Android project directory should look like this:


WorkSpace Navigator
SUP101Android
assets
bin
gen [Generated Java Files]
libs
res
drawable-hdpi
drawable-ldpi
drawable-mdpi
drawable-xhdpi
layout
customer.xml
detail.xml
main.xml
menu
values
values-v11
values-v14
src
com
mycorp
sup101
android
app
CustomerListAdapter.java
DetailActivity.java
SUP101SampleActivity.java
mbo
intrnl
ChangeLogImpl.java
ChangeLogKey.java
Customer.java
EntityType.java
KeyGenerator.java
KeyGeneratorPK.java
KeyPackageName.java
LocalKeyGenerator.java
LocalKeyGeneratorPK.java
LogRecordImpl.java
OfflineAuthentication.java
PackageProperties.java
PersonalizationParameters.java
SUP101DB.java
AndroidManifest.xml
ic_launcher-web.png
proguard-project.txt
project.properties

Creating a Launch Configuration for the Project

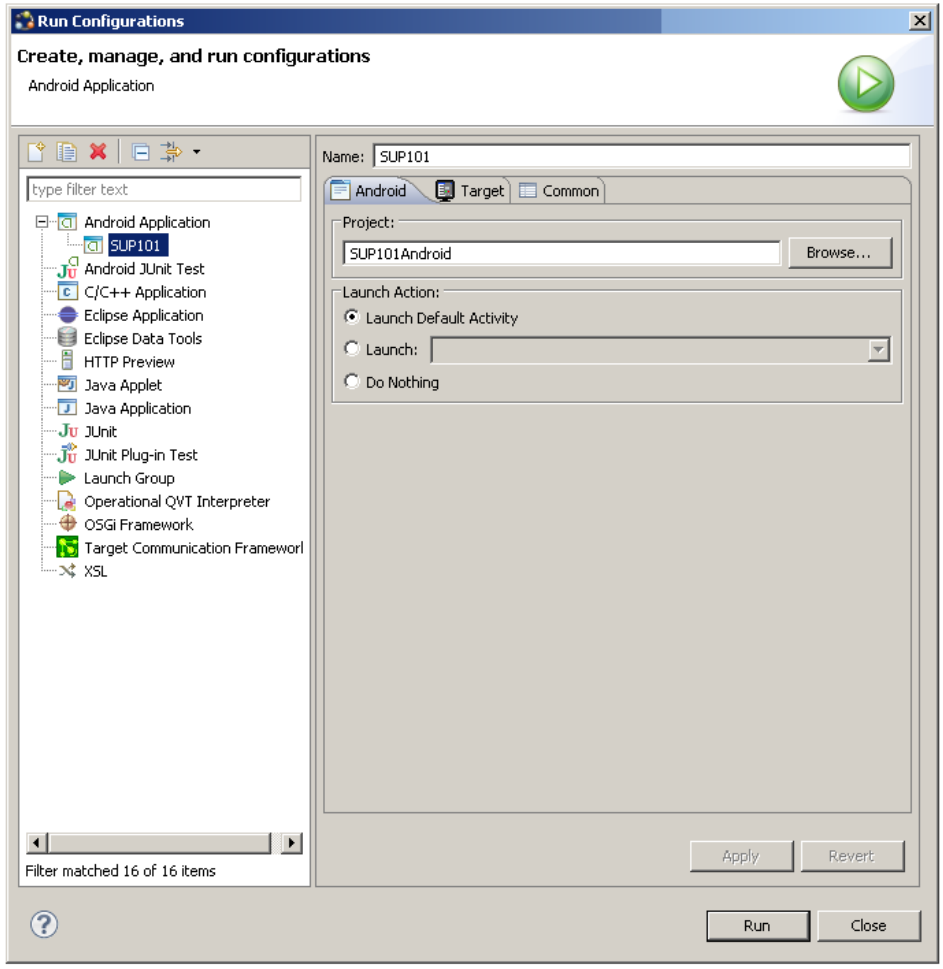
Create a new launch configuration for the SUP101Android project. The configuration specifies how the application launches, and defines the target Android platform.

Prerequisites

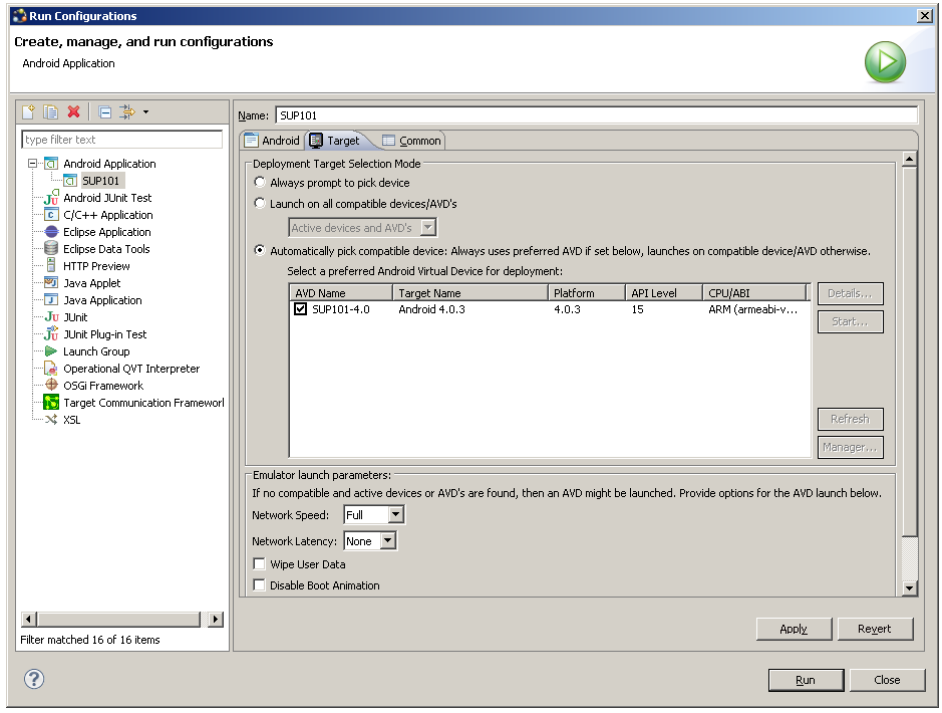
In the Sybase Unwired WorkSpace, use the AVD Manager to add a new target Android Virtual Device (AVD).

Task

1. In Workspace Navigator, right-click the **SUP101Android** project, and select **Run As** > **Run Configurations**.
2. Right-click **Android Application** and select **New**.
3. In the Name field, enter `SUP101`.
4. In the Android tab, click **Browse** and select **SUP101Android**. Click **OK**.
5. In the Launch Action area, select **Launch Default Activity**.



6. In the Target tab, select a deployment target. For example, select Automatically pick compatible device, then specify an AVD for deployment. Accept all other default settings.

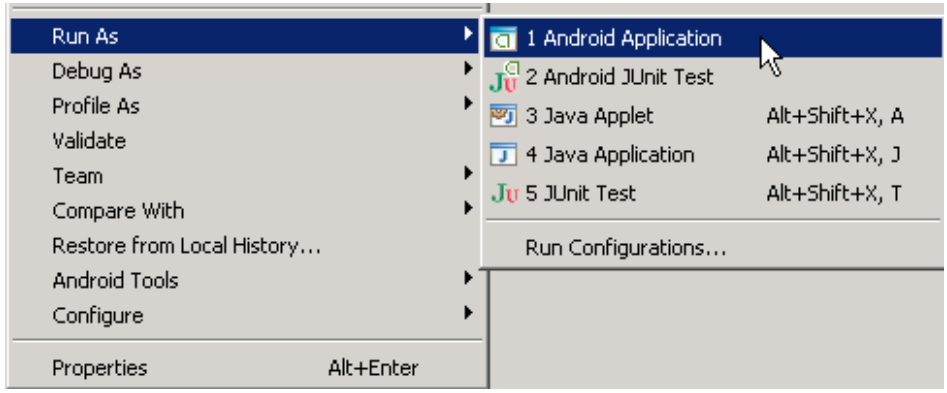


7. Click **Apply**, then **Close**.

Testing the Device Application on the Android Emulator

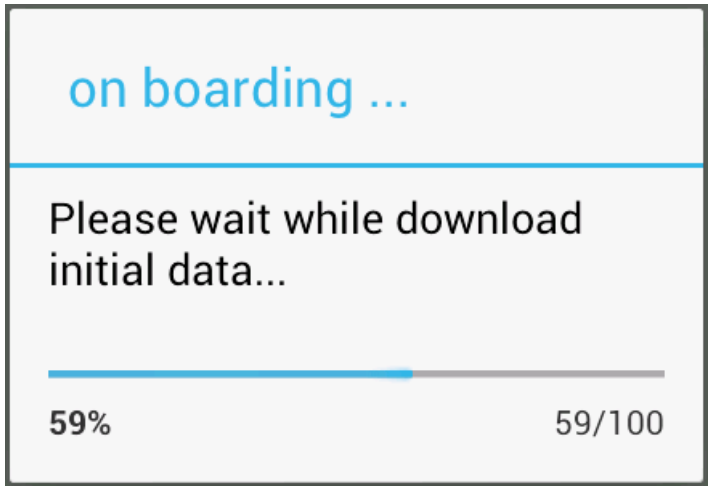
Run the SUP101Android application on the Android emulator, and change customer information to update the interface.

1. In WorkSpace Navigator, right-click **SUP101Android** and select **Run As** > **Android Application**.



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

The application activation (on boarding) image indicates that the application is registering data from Unwired Server.



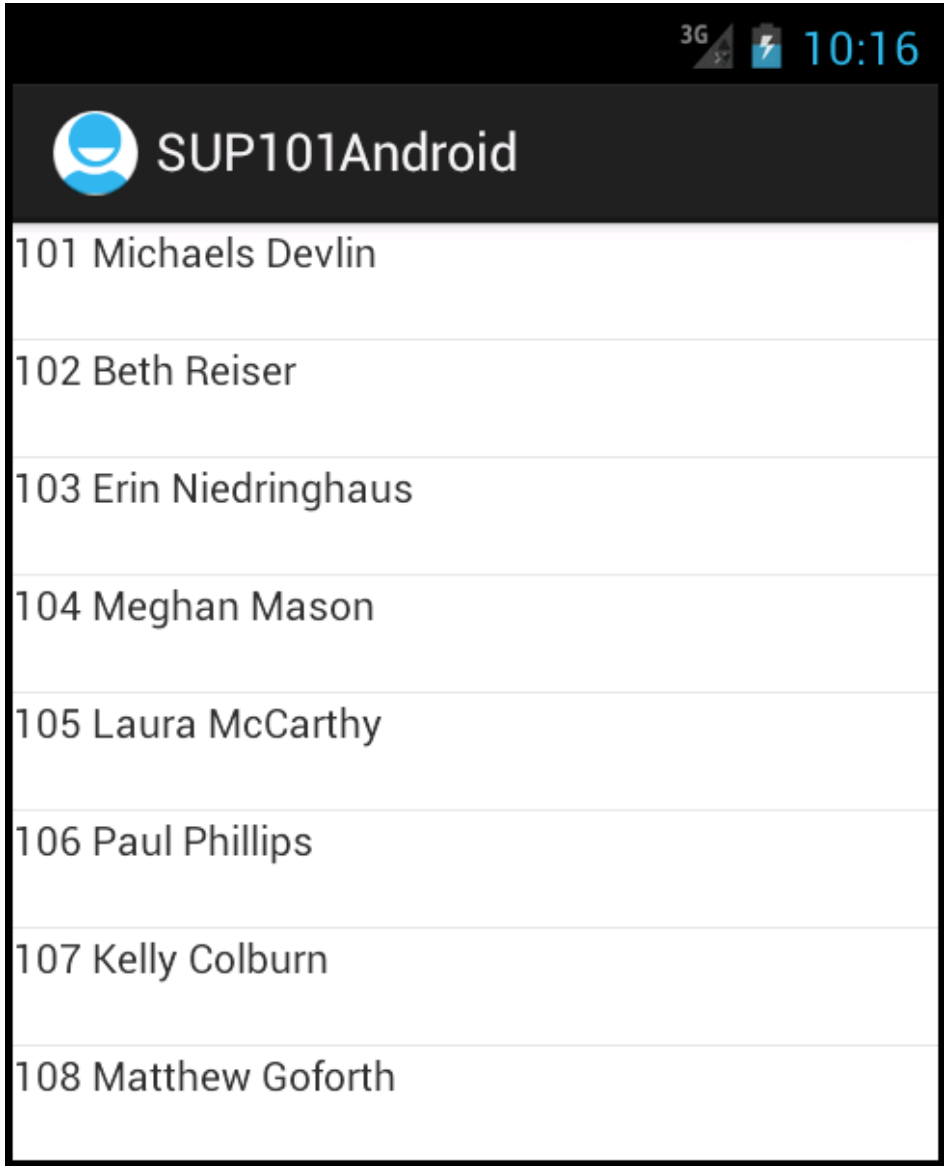
During initialization, the system enables the operation to target change notifications using:

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

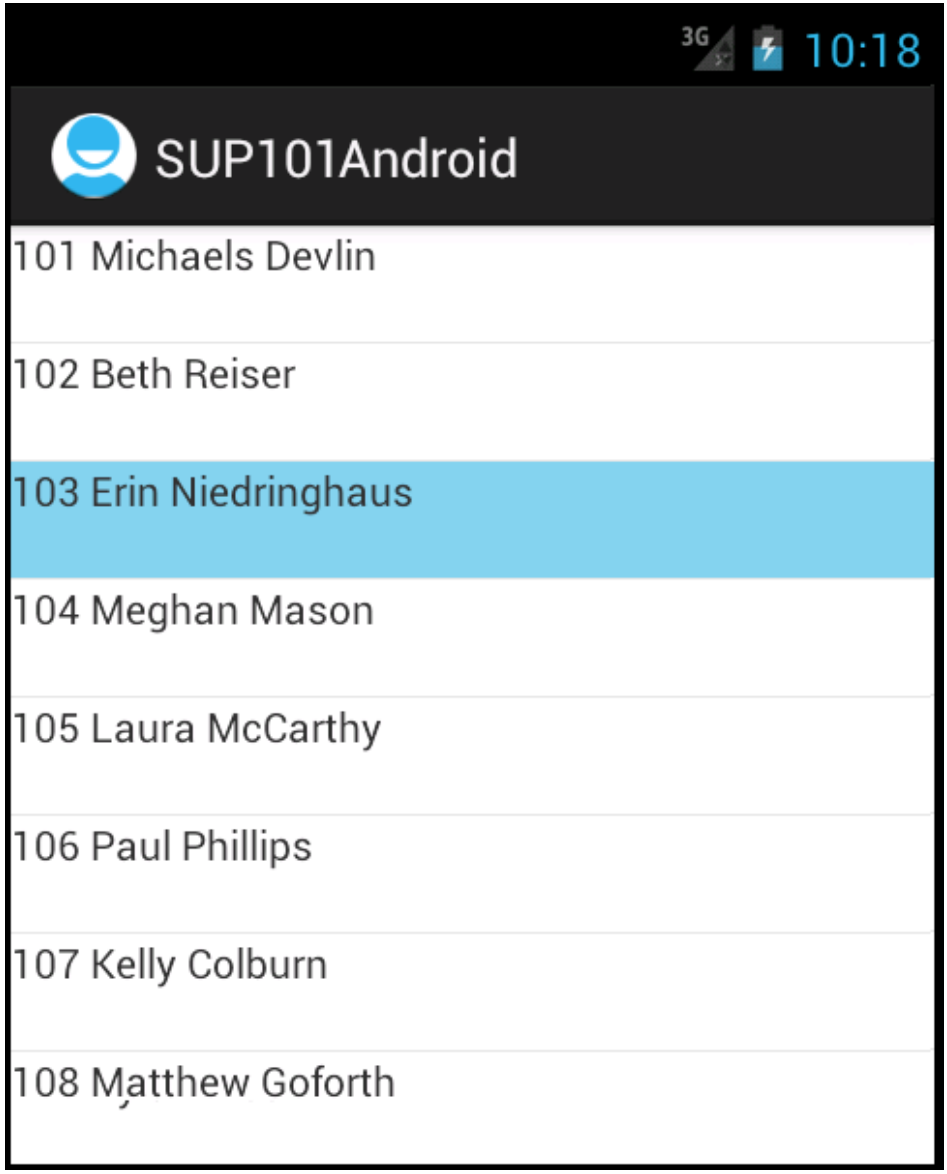
When the data finishes synchronizing, the device application shows the SUP101Android application with a list of customer data in a ListView control. You can scroll through the customer list to see more data and to make changes. The data loads from the database on demand.

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

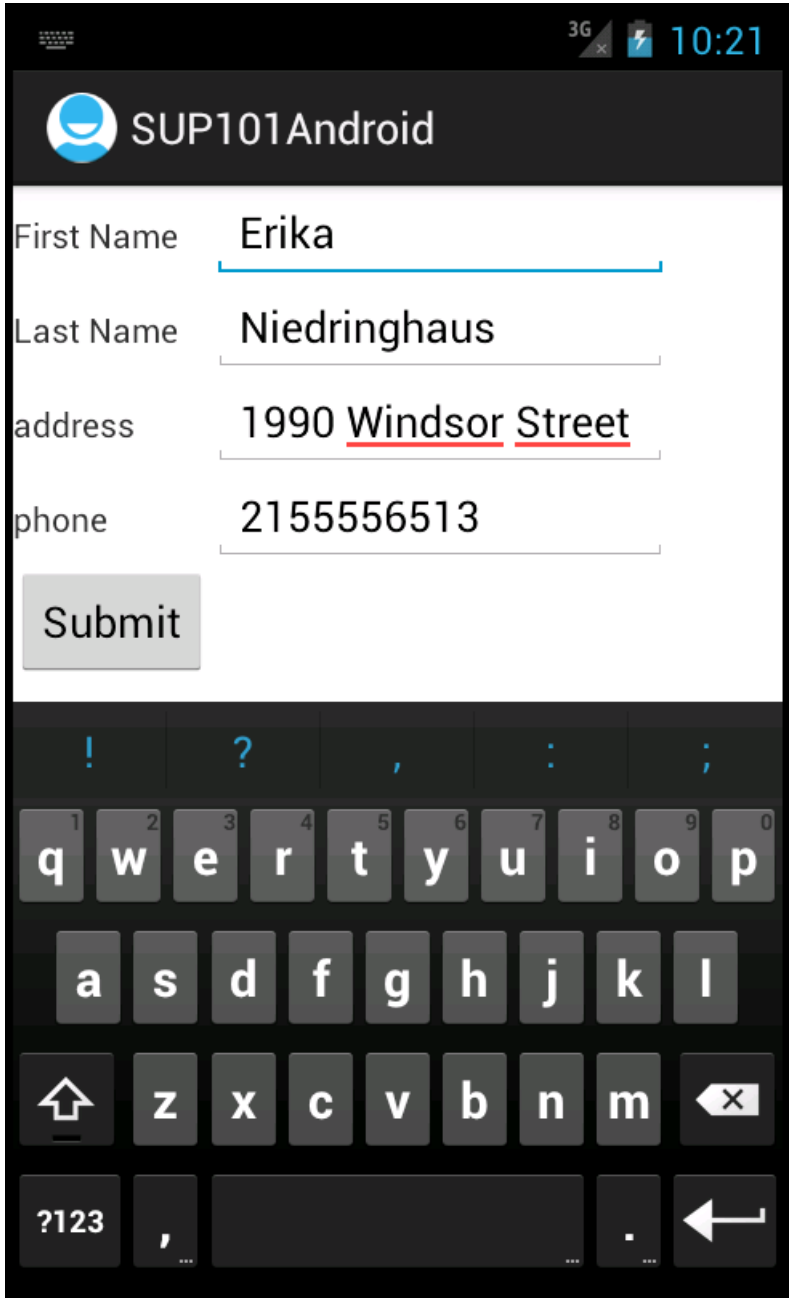
When the application queries the customer list, it uses an `SUP101DB.executeQuery()` API to get only columns that are needed, such as (fname, lname...), instead of the entire customer object; this results in better performance.



2. Select the customer to update.



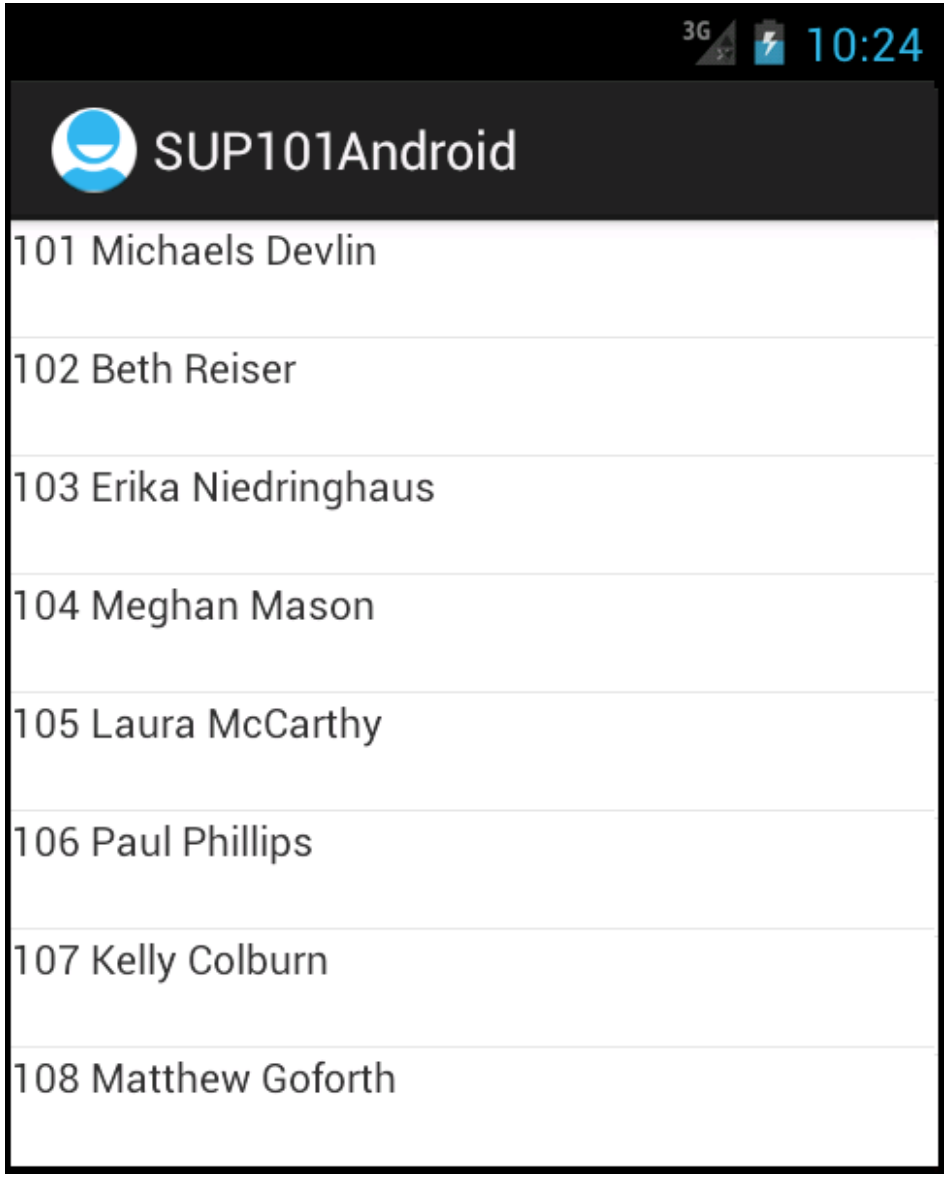
3. In the customer detail screen, change the first name of the customer and click **Submit**.



The Submit button is mapped to the synchronize operation using `SUP101DB.beginSynchronize`. The synchronization occurs in the background, so the user interface is unaffected.

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

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



4. Close the emulator to stop the SUP101Android 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 Product Documentation Web site regularly for updates: *http://sybooks.sybase.com/sybooks/sybooks.xhtml?id=1289&c=firsttab&a=0&p=categories*, 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 contains source code for its associated tutorial. It does not contain the completed tutorial project. 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.
- *SAP Mobile WorkSpace: Mobile Business Object Development* – provides information about using SAP Mobile Platform to develop MBOs and generate Object API code that can be used to create native device applications and Hybrid Apps.

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: Hybrid Apps*
- *Developer Guide: OData SDK*
- *Developer Guide: REST API Applications*

Customize and automate:

- *Developer Guide: Unwired Server Runtime > 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 10
Android application
    attributes 19
Android project 13
    manifest file 21
    src folder 19
Android SDK 9
AndroidManifest.xml 13, 19
    Detail Activity 21
application properties 19

B

build path 13

C

ClientLib.jar 13
compiler resource 18
customer.xml 22
CustomerListAdapter.jar 22

D

default_package.jar 11
deployment target, launch 24
deployment_unit.xml 11
Detail Activity 21
detail.xml 22
DetailActivity.jar 22

E

example projects 1

G

Generate Code wizard 11
generated object API code 11
    using 19
generating code 11

H

Hybrid App package tutorial 1

J

JAR files
    ClientLib.jar 13
    sup-client.jar 13
    UltraLiteJNI12.jar 13
Java class, creating 22
Java files
    CustomerListAdapter.jar 22
    DetailActivity.jar 22
    SUP101SampleActivity.jar 22
Java object API code, generating 11
Java perspective 22
JDK 9

L

launch configuration 24
layout files
    customer.xml 22
    detail.xml 22
    main.xml 22
library resource 18

M

main.xml 22
manifest file 13, 21
mobile business object tutorial 1

O

Object API tutorials 1

P

project build path 13
properties, application 19

R

resources
    compiler 18

library 18
run configurations 24

S

samples
downloading 33
simulator 9
src folder
Android project 19
Sybase Unwired WorkSpace 19
sup-client.jar 13
SUP101Android application, testing 26
SUP101SampleActivity.jar 22
SUP101SampleProject 13
Sybase Control Center
connecting to 4
Sybase Mobile SDK
installing 3
Sybase Unwired Platform
documentation resources 33
getting started 3
installing 3
Sybase Unwired WorkSpace
basics 5
how to access online help 5
starting 4
Sybase Unwired WorkSpace basics 5
Sybase Unwired WorkSpace, src folder 19

T

troubleshooting information 5
tutorials 1
downloading 33

U

UltraLiteJNI12.jar 13
Unwired Platform Runtime
installing 3
Unwired Platform services 3
user interface, creating 22

V

virtual devices 9

X

XML files
customer.xml 22
detail.xml 22
main.xml 22