SYBASE®
An SAP® Company

Tutorial: BlackBerry Object API Application Development

Sybase Unwired Platform 2.1 ESD #3

DOCUMENT ID: DC01214-01-0213-01
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 a BlackBerry Application

Generate code for the BlackBerry platform, develop a BlackBerry device application with that code and sample files, and test the application's functionality on a simulator.

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*
- Review the *Supported Hardware and Software* requirements for this release.

Task

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

Installing the BlackBerry Java Plug-in for Eclipse

The BlackBerry Java Plug-in for Eclipse enables you to finish developing the BlackBerry application using BlackBerry smartphone-specific development, debugging, and simulation tools.

1. Download the BlackBerry Java Plug-in for Eclipse (full installer) from *http://us.blackberry.com/developers/javaappdev/*.
2. Run the downloaded installer and follow the instructions to install the BlackBerry Java Plug-in for Eclipse. Accept the default installation folder: do not specify the Unwired WorkSpace Eclipse directory.

Setting System and Path Variables

Set the Java system variable and add Java to the system path.

1. Set the JAVA_HOME system variable:

a) Right-click the computer icon for the system running the BlackBerry Java Plug-in, and select **Properties**.
b) In System Properties, select **Advanced**, then click **Environment Variables**.
c) In System variables, select **JAVA_HOME** and click **Edit**.
d) In the Edit System Variable dialog, add a JDK directory path.

   Check the JDK system requirements for the BlackBerry environment you are using: *http://us.blackberry.com/developers/javaappdev/*.

   If you have Sybase installed, you can use the existing JDK to set the JAVA_HOME variable: `JAVA_HOME=C:\Sybase\UnwiredPlatform\JDK1.6.0_31`.

   Note: For the Sybase 64-bit environment, you would still use the 32-bit JDK.

2. Add %JAVA_HOME%\bin to the system path:
   a) In System variables, select **Path** and click **Edit**.
   b) In the Edit System Variable dialog, add a JDK directory path.

      If you have Sybase installed, you can use the existing JDK to set the variable: `C:\Sybase\UnwiredPlatform\JDK1.6.0_31\bin`.
3. Restart the system for the environment variables to take effect.

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.

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 a `RIM` 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 select the default configuration and continue without any changes.
5. In the Select Mobile Business Objects page, select the **Customer MBO**, then click **Next**.
6. In the Configure Options page, specify these values and click **Finish**.

Option	Description
Language	**Java**
Platform	**Java ME for BlackBerry**
Unwired server	**My Unwired Server**
Server domain	Accept default value.
Page size	Accept default value of 1024.
Package	(Optional) 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	You do not need to choose one for this tutorial. You see a warning at the top of the page: `The third-party class 'net.rim.device.api.system.ApplicationDescriptor' cannot be found.` You can ignore this warning; it does not affect code generation.
Generate JavaDoc	Unselect.

   You can ignore the error icons that appear in WorkSpace Navigator under `\Generated Code\src\SUP101`, and the error in the Problems view: `net cannot be resolved`.

   The error occurs because the package Java file called by `SUP101DB.jar` is not yet in the project build path. The errors resolve later, when you build the application in the BlackBerry Java Plug-in for Eclipse.
7. In the Success dialog, click **OK**.
   In the Generated Code directory, you see a `\src\SUP101` folder.

Creating the BlackBerry Project

Create a new BlackBerry SUP101Sample project in the BlackBerry Java Plug-in for Eclipse. Add library resources to the project.

Prerequisites

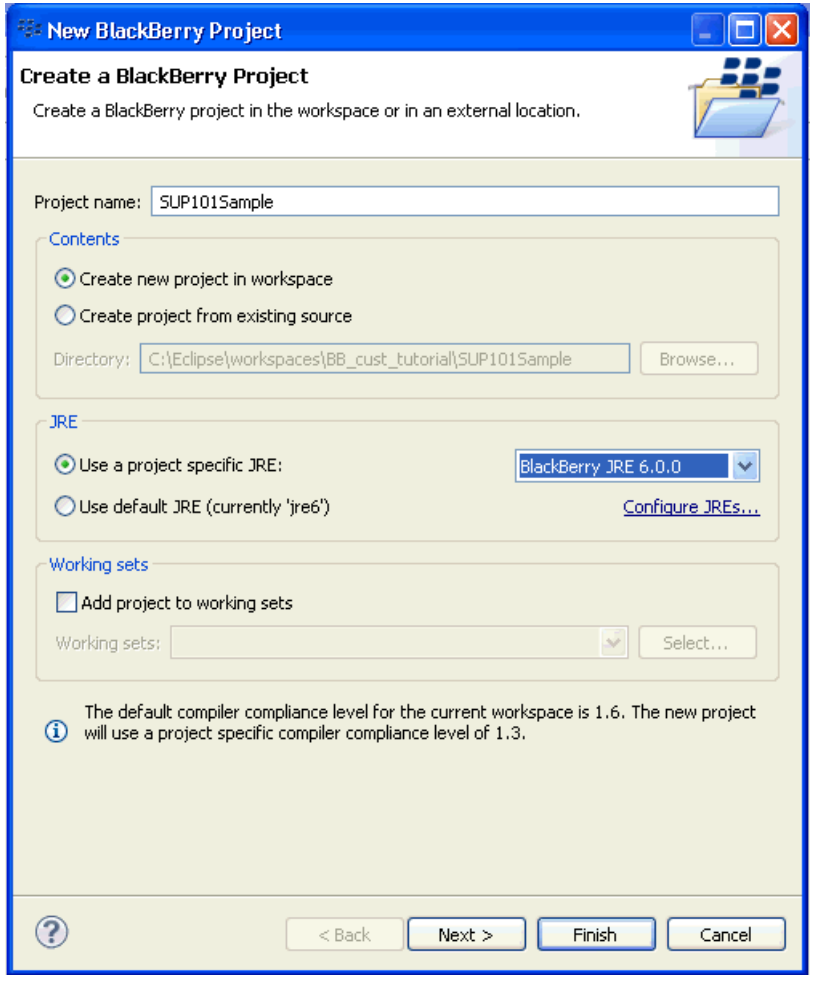
Install the BlackBerry Java Plug-in for Eclipse. Review the *Supported Hardware and Software* requirements to ensure you have the correct version.

Task

1. Start the BlackBerry Java Plug-in for Eclipse.

   Note: When the Eclipse Workspace Launcher prompts you to select a workspace, do not choose one that you are using with Unwired WorkSpace. If necessary, select or create a new workspace.

2. Depending on the BlackBerry Plugin version:
   - Select **File > New > BlackBerry Project**, or
   - Select **File > New > BlackBerry > BlackBerry Project**.
3. In the New BlackBerry Project wizard, use these values and click **Next**.
   - Project name – enter `SUP101Sample`.
   - Use a project specific JRE – select the BlackBerry OS version that was automatically installed with the plugin.

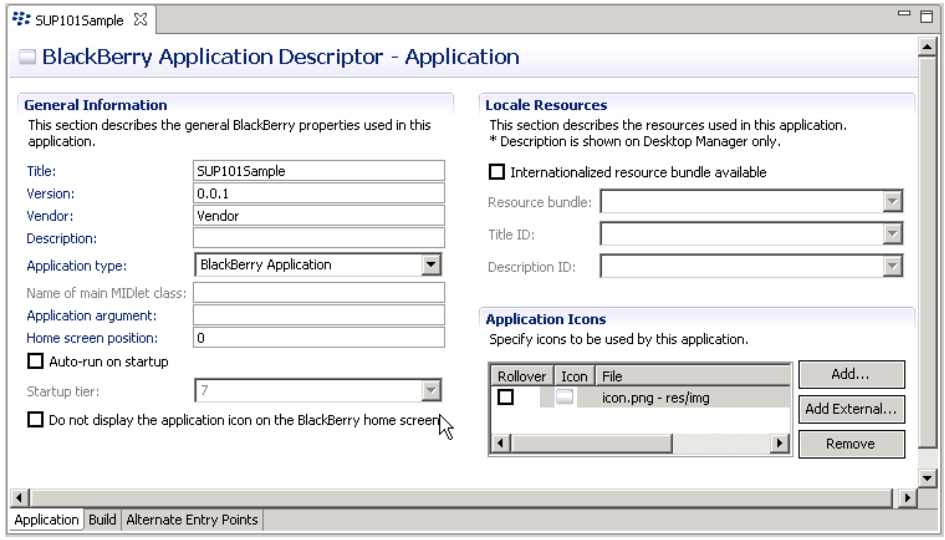


4. In the Java Settings page, modify the build path to point to the correct location for the `sup_client2.jar` and `UltraLiteJ12.jar` files for the project:
   a) Click the **Libraries** tab.
   b) Click **Add External JARs**.
   c) Browse to `<UnwiredPlatform_InstallDir>\UnwiredPlatform\MobileSDK213\ObjectAPI\BB`.
   d) Select the two JAR files, then click **Open**.
   e) Click **Finish**.

Configuring BlackBerry Application Properties

Modify the BlackBerry application description to define the general BlackBerry properties used in an application.

1. In Package Explorer, expand the **SUP101Sample** project.
2. Open the `BlackBerry_App_Descriptor.xml` file.
3. Select the **Application** tab.
4. In the Title, enter `SUP101Sample`.
5. (Optional) Enter a version, vendor, and description.
   You can review the other options in the Application Descriptor window, such as general attributes, resources, and build settings of the project. For this tutorial, leave all other settings unchanged.



6. Click **File** > **Save**.

Copying Unwired Platform Files to the Sample Project

Copy Unwired Platform BlackBerry runtime files and generated object API code to the SUP101Sample project. You had created the code previously in the Generate Code wizard.

1. In Windows Explorer, browse to `<UnwiredPlatform_InstallDir>\UnwiredPlatform\MobileSDK213\ObjectAPI\BB`.
2. Copy these files: `CommonClientLib.cod`, `MessagingClientLib.cod`, `MocaClientLib.cod`, `sup_client2.cod`, and `UltraLiteJ12.cod`.

3. Paste the copied COD files into the BlackBerry Java Plug-in simulator directory, located in `<BlackBerry_Eclipse_InstallDir>\plugins\net.rim.ejde.componentpack<version>\components\simulator`.
4. In Windows Explorer, go to the workspace folder you set for Sybase Unwired WorkSpace, and copy the generated code files.
   For example, go to `C:\Documents and Settings\user1\SUP-wksp\SUP101\Generated Code\src` and copy the `SUP101` folder.
5. In the BlackBerry Application Development workspace, go to the `SUP101Sample` project and paste the folder into the `src` directory.
   For example, go to `C:\Documents and Settings\user1\BB-wksp\SUP101Sample\src` and paste the `SUP101` folder.
   In Package Explorer, under the `SUP101Sample/src` folder, you see the `SUP101` and `SUP101.intrnl` folders.

   If the copied folders do not appear, click **Refresh** or press **F5**.

Creating the User Interface

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

Prerequisites

Obtain text files from the `SUP_BB_Custom_Dev_Tutorial_code.zip` file. The text files contain code snippets that you need to build the user interface. You will create these classes for the interface: `CustomerSample`, `CustomerSampleScreen`, `CustomerList`, `CustomerDBCallback`, and `ObjectAPIUtil`.

- If you are viewing this guide online from the Sybase Product Documention Web site, click *SUP_BB_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*. Select **Sybase Unwired Platform**. Select the appropriate version, navigate to this topic in the tutorial, then click the link for the ZIP file to save the files locally.

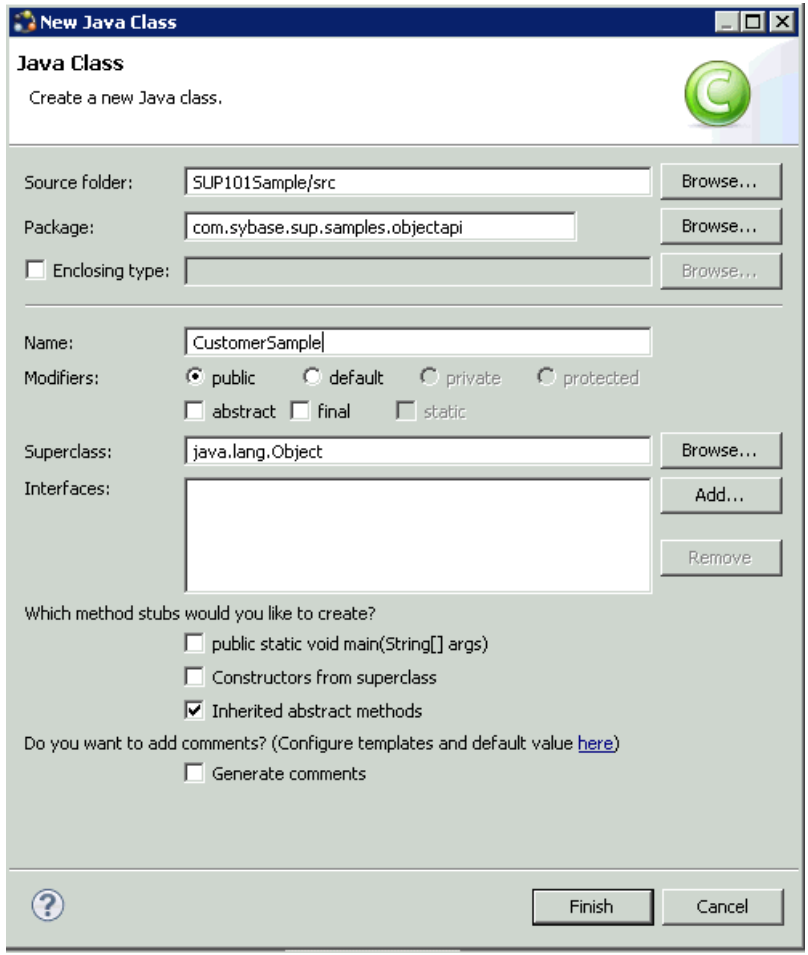
Task

1. In Package Explorer, right-click **SUP101Sample**, and select **New** > **Package**.
2. In the New Java Package wizard, use these values and click **Finish**.
   - Source folder – verify `SUP101Sample/src` appears.
   - Name – enter `com.sybase.sup.samples.objectapi`.

3. In Package Explorer, you will create one new Java class for the application in the `com.sybase.sup.samples.objectapi` package, and then copy the remaining four classes in the ZIP file into the package. Here is the list of files in the ZIP:
   - `CustomerDBCallback.java` – implements the CallbackHandler to demonstrate how to track changed entities in the onSynchronize callback method.
   - `CustomerList.java` – populates the customer list.
   - `CustomerSample.java` – creates the main customer application.
   - `CustomerSampleScreen.java` – creates the customer screen.
   - `ObjectAPIUtil.java` – governs how the application initializes and synchronizes data in the background.

   a) Right-click the **com.sybase.sup.samples.objectapi** package and select **New > Class**.

   b) In the Java Class page, for the name, enter one of the Java class names, for example: `CustomerSample`, and click **Finish**.



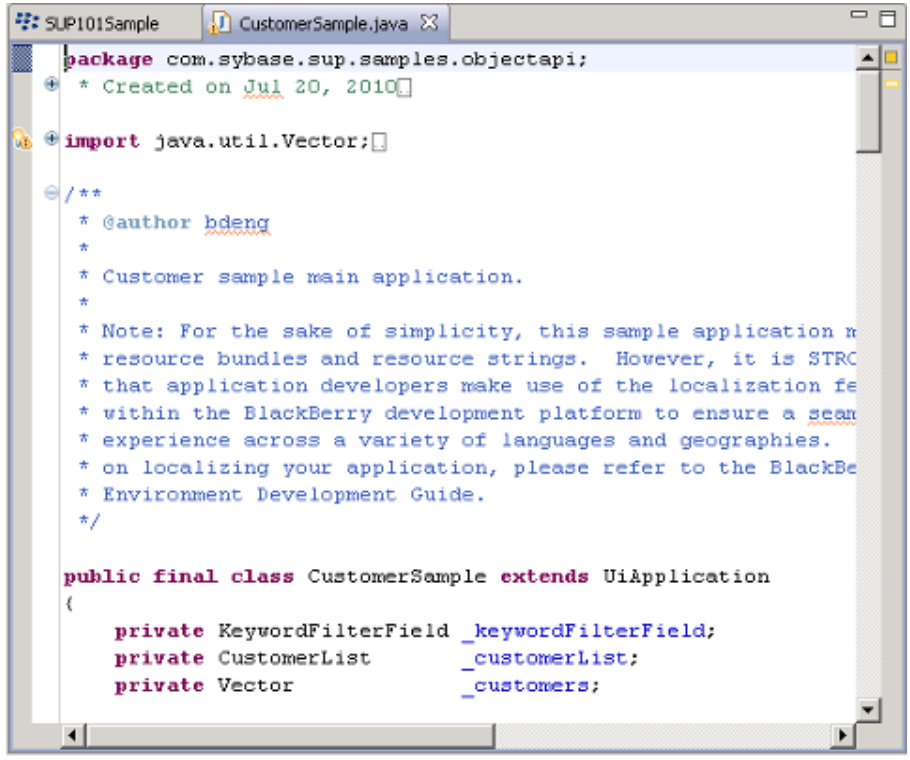
In the SUP101Sample, under the `src\com.sybase.sup.samples.objectapi` package, you see the new Java class: `CustomerSample.java`.

c) Browse to the directory where you saved the ZIP file.

d) Copy the content of the Java text file with the same name: `CustomerSample.java`.

e) In Package Explorer, open the new Java class with the same name: `CustomerSample.java`.

f) Paste the copied text code into the new Java class: `CustomerSample.java`. Copy over any existing content.



g) Select **File** > **Save**.
The package contains errors until you create the other classes. Ignore these errors and continue.

h) For the remaining four classes, copy the Java files from the ZIP archive to the `SUP101Sample\src\com.sybase.sup.samples.objectapi` package.
The compilation errors in the Java files resolve when you finish creating the classes.

4. Modify the host name or IP address in the `ObjectAPIUtil.java` file to point to the Sybase Unwired Server.

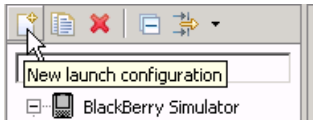
a) In Package Explorer, expand the **SUP101Sample** project.

b) Under the `\src\com.sybase.sup.samples.objectapi` package, double-click the `ObjectAPIUtil.java` file.
c) Modify the host name or IP address, and **Save**.

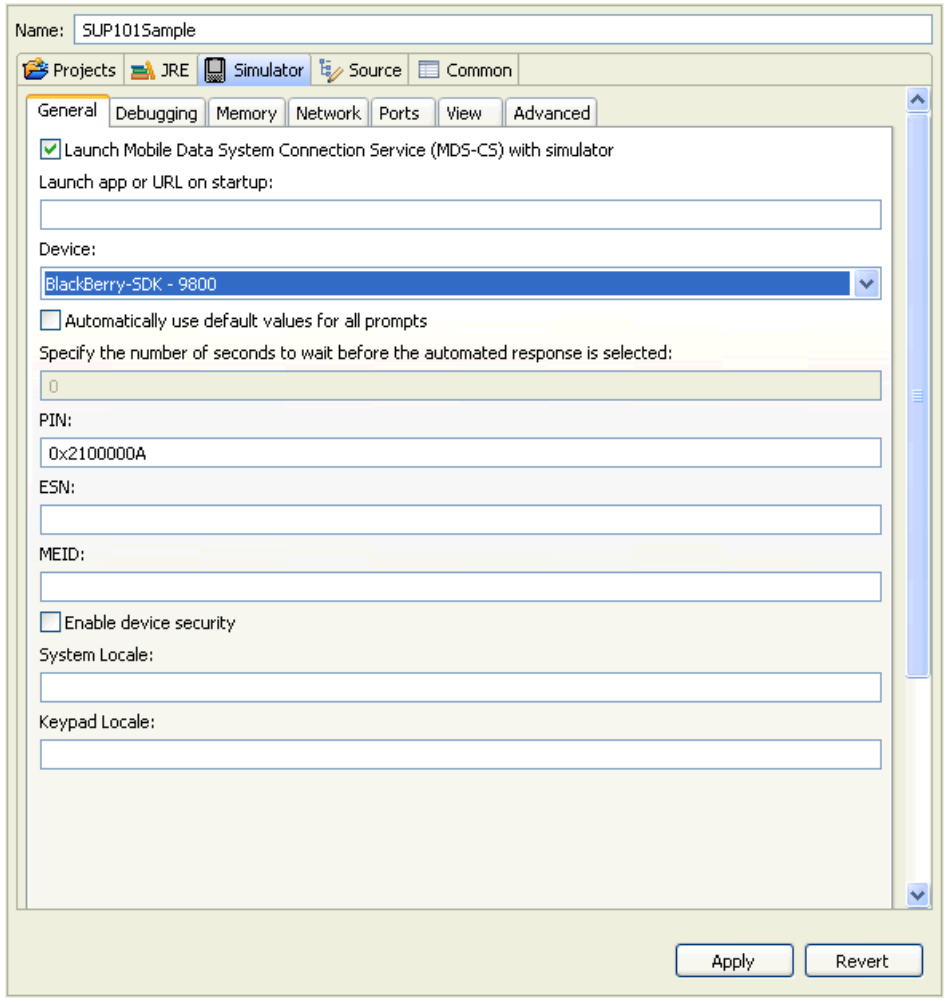
Creating a Launch Configuration for the Project

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

1. In Package Explorer, right-click the **SUP101Sample** project, and select **Run As > Run Configurations**.
2. Select **BlackBerry Simulator** and the **New launch configuration** icon.



3. In the Projects tab, select the **SUP101Sample** project.
4. For the name, enter `SUP101Sample`.
5. In the JRE tab, select the **Project JRE (BlackBerry JRE *Version*)**.
6. In the **Simulator** tab, select **Launch Mobile Data System Connection Service (MDS-CS) with simulator**. Then select the **BlackBerry-SDK - *Version*** as the device.



If you are running the BlackBerry JDE 7.0 with the BlackBerry Java Plug-in v1.5, you need to launch the Mobile Data System (MDS) manually for the BlackBerry simulator to run.

MDS is a framework that offers security, wireless connectivity, manageability options, and development methods. You need it to run the simulator. It is a known issue for BlackBerry that the simulator cannot launch MDS. This has been fixed in BlackBerry JDE 7.1.

a) From a command prompt, go to `<Eclipse_InstallDir>\plugins\net.rim.ejde.componentpack7.0.0_7.0.0.33\components\MDS`.
b) Run MDS manually, enter: `start run.bat`.
c) When the following message appears, ignore it and press any key to continue.

```
ERROR: Your java version does not meet the minimum
requirement of 1.6.0_20. Press any key to continue...
```

MDS starts and you can launch the BlackBerry simulator. Leave the command window open.

7. In the **Network** tab, verify that **Disable Registration** is checked, or the sample application cannot get data from Unwired Server.
8. Click **Apply**, then **Close**.

Testing the Device Application on the BlackBerry Simulator

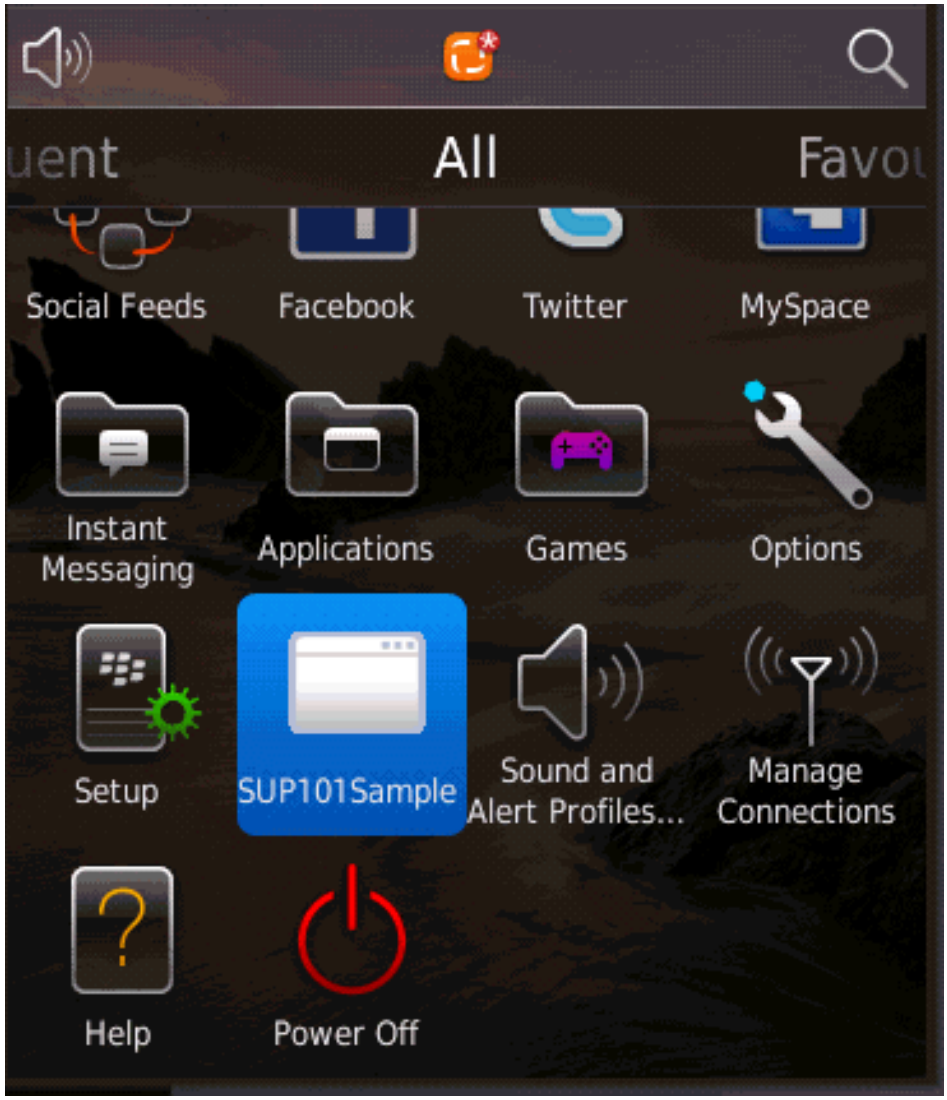
Run the SUP101Sample application on the BlackBerry simulator, and change customer information to update the interface.

Depending on the simulator you are using, the screens may vary.

1. In Package Explorer, right-click the **SUP101Sample** project and select **Run As** > **BlackBerry Simulator**.
   It may take several minutes for the BlackBerry simulator's Setup screen to appear. If this is the first time running the simulator, cancel the Setup screen.


BlackBerry
Setup
3g
Setup
Select the feature you want to set up.
Email Accounts (1)
Instant Messaging
Social Networking
Wi-Fi
Personalization
Select the feature you want to customize.
Ring Tones
Wallpaper
Languages
Fonts
Date and Time
Keyboard
SIM Contacts Sync
Password
space
sym

2. On the main window, click **All** to access the applications screen, then scroll until you see the SUP101Sample application.



3. Click **SUP101Sample** to launch the application.

   The application registers and synchronizes data from the server in the background.

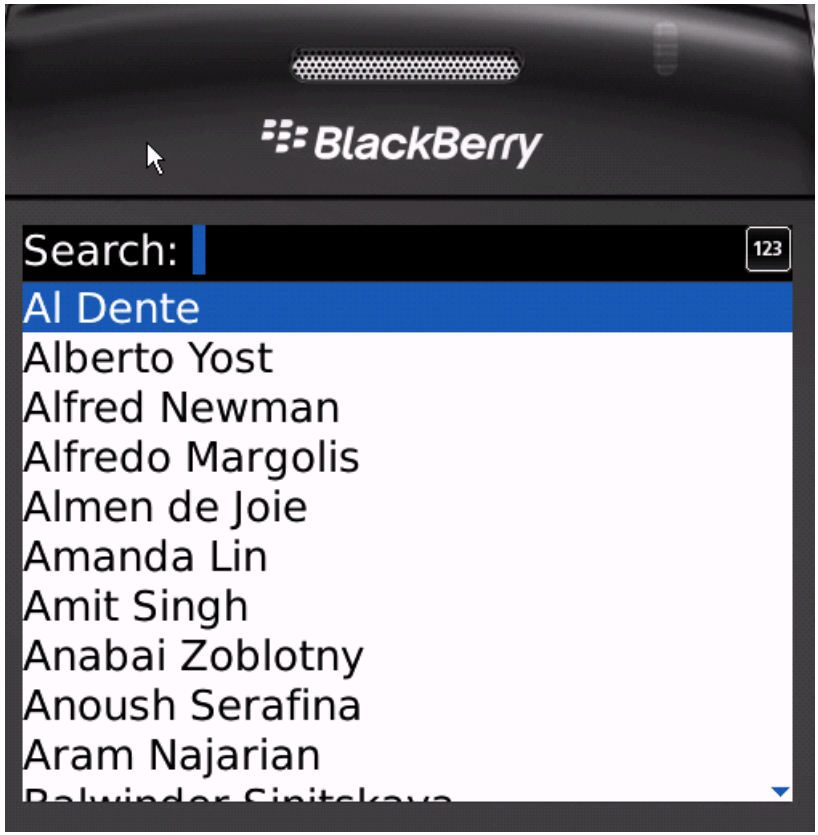
   In the initialization process, the system enables the operation to target change notifications using:

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

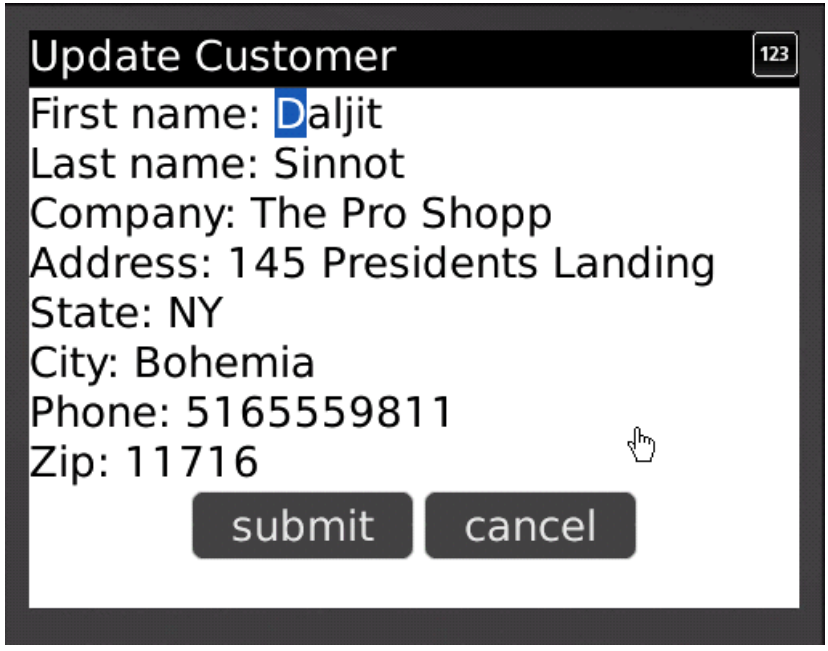
```
sg.save();
```

After the application initializes, the device application shows the SUP101Sample application with a list of customer data. You can scroll through the customer list to see more data, search, and make changes. The data loads from the database on demand.

When the application queries the customer list, it uses a named query findAll. If the customer list is long, 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.



4. Focus on the customer list and in the **Search** bar, enter `dal`.
   The customer list is filtered and only customers with a first or last name beginning with "dal" are shown, in this case, Daljit Sinnot.
5. Select the customer, **Daljit Sinnot**, and click the trackball.
   The Update Customer screen for Daljit Sinnot appears.



6. Change the first name of the customer to `abc` and click **submit**.

   The **submit** button maps to the update operation of the customer mobile business object. The synchronize operation uses SUP101DB.beginSynchronize and occurs in the background so the user interface is not affected. When the application synchronizes, any pending operations are uploaded to Unwired Server.

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

   The customer list appears with the name change you made.

7. Close the simulator 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