Tutorial: BlackBerry Object API Application Development

SAP Mobile Platform 2.3 SP02

DOCUMENT ID: DC01940-01-0232-01
LAST REVISED: May 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

SAP Mobile Platform Tutorials

The SAP® 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 SAP Mobile 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 SAP Mobile Platform (On-Premise)

Install and learn about SAP Mobile Platform and its associated components.

Complete the following tasks for all tutorials, but you need to perform them only once.

Installing SAP Mobile Platform

Install SAP Mobile SDK and SAP Mobile Platform Runtime.

Before starting this tutorial, install all the requisite SAP Mobile Platform components. See the SAP Mobile Platform documentation at *http://sybooks.sybase.com/sybooks/sybooks.xhtml?id=1289&c=firsttab&a=0&p=categories*:

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

1. Install these SAP Mobile Platform Runtime components:
   - Data Tier (included with single-server installation)
   - SAP Mobile Server
2. Install SAP Mobile SDK, which includes:
   - Development support for native Object API and OData SDK applications, as well as HTML5/JS Hybrid Apps.
   - SAP Mobile WorkSpace, the Eclipse-based development environment for MBOs and Hybrid Apps.

Starting SAP Mobile Platform Services

Start SAP Mobile Server, SAP Control Center, the sample database, the cache database (CDB), and other essential services.

The way in which you start SAP Mobile Platform Services depends on the options you selected during installation. You may need to manually start SAP Mobile Platform Services. Select **Start** > **(All) Programs** > **SAP** > **Mobile Platform** > **Start SAP Mobile Platform Services**.
The following services will be started:

- SAP Control Center *<Version>*
- SAP Mobile Platform Cache DB

- SAP Mobile Platform SampleDB
- SAP Mobile Server

SAP Mobile Platform Services enable you to access the SAP Mobile Platform runtime components and resources.

Note: The SAP Mobile Platform installer creates the Windows service (SAP Mobile Platform Sample DB) that runs the sampledb server only when you install SAP Mobile Server with a Personal or Enterprise Development license. If you installed SAP Mobile Server with an Enterprise Server (production) license, you must create this service using the `sampledb.bat` command line utility. See *Create or Remove the Windows Service for sampledb Server (sampledb) Utility* in *System Administration* for more information about using this command line utility.

Starting SAP Mobile WorkSpace

Start the development environment, where you can create mobile business objects (MBOs), create connection profiles and manage SAP Mobile Server connections, develop Hybrid Apps, and generate Object API code.

Select **Start > (All) Programs > SAP > Mobile Platform > Mobile WorkSpace 2.3**.

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

Next

To read more about SAP Mobile WorkSpace concepts and tasks, select **Help > Help Contents**.

Connecting to SAP Control Center

Open SAP Control Center to manage SAP Mobile Server and its components.

From SAP 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 SAP Mobile Server, click **Help > Help Contents**.

1. Select **Start** > **(All) Programs** > **SAP** > **SAP Control Center**.

   Note: If SAP Control Center does not launch, make sure that the SAP Control Center service is started in the Windows Services dialog.

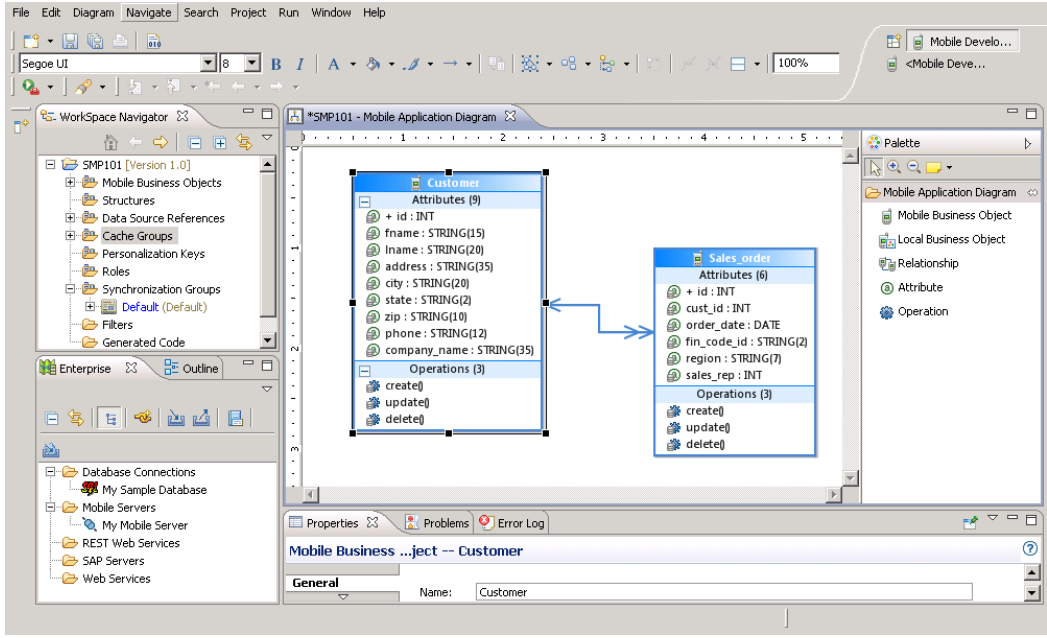
2. Log in by entering the credentials set during installation.

   SAP Control Center gives you access to the SAP Mobile Platform administration features that you are authorized to use.

Learning SAP Mobile WorkSpace Basics

SAP Mobile WorkSpace features are well integrated in the Eclipse IDE. If you are unfamiliar with Eclipse, you can quickly learn the basic layout of SAP Mobile WorkSpace and the location of online help.

- To access the online help, select **Help** > **Help Contents**. Some documents are for SAP Mobile WorkSpace, 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** in the upper right corner of the page.
  - Reopen the Welcome page by selecting **Help** > **Welcome**.
  - To learn about tasks you must perform, select the **Development Process** icon.
- In SAP Mobile 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 SAP Mobile Server.
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.

Window	Description
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.
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 object API code containing mobile business object (MBO) references for the BlackBerry platform, develop a BlackBerry device application using that code and sample files, and test the application's functionality on a simulator.

Prerequisites

Note: This tutorial was created using SAP Mobile Platform 2.3, BlackBerry Java SDK 7.1.0.10, BlackBerry Java Plug-in (core) 2.0.0, and run on an BlackBerry SDK 9900 device simulator. If you use a different version, some steps may vary.

1. Complete the tasks in *Getting Started with Mobile Platform*.
2. Either:
   - create the MBO project by completing *Tutorial: Mobile Business Object Development*, or
   - download and deploy the MBO SMP101 example project (complete project files) from the SAP® Community Network: *http://scn.sap.com/docs/DOC-8803*.

   Note: If you upgrade SAP Mobile SDK after completing the tutorial, you can convert the project to the current SDK by importing the earlier project into the SAP Mobile WorkSpace and then accepting the confirmation prompt.
3. (Optional) To use as a reference and copy source code when completing this tutorial, download the BlackBerry SMP 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 BlackBerry Java Plug-in for Eclipse.
   See the *Supported Hardware and Software* guide at *http://sybooks.sybase.com/sybooks/sybooks.xhtml?id=1289&c=firsttab&a=0&p=categories*. Select the appropriate version of the SAP Mobile Platform document set.

Task

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

Installing the BlackBerry Java Plug-in for Eclipse

Install the supported version of BlackBerry Java Plug-in in the SAP Mobile WorkSpace Eclipse environment.

See *RIM BlackBerry Versions for Object API* in *Supported Hardware and Software* at *http://sybooks.sybase.com/sybooks/sybooks.xhtml?id=1289&c=firsttab&a=0&p=categories*. Select the appropriate version of the SAP Mobile Platform document set.

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

1. Confirm that your system meets the requirements at *https://developer.blackberry.com/java/download/eclipse/*.
2. Start SAP Mobile WorkSpace, then select **Help > Install New Software**.
3. In the Available Software window, click **Add**.
4. In the Add Repository dialog, enter `BlackBerry Plugin` for the name and `http://www.blackberry.com/developers/jar/win/java` for the location. Click **OK**.
5. In the Available Software dialog, select **BlackBerry Java Plug-in (core)** and the appropriate version of the **BlackBerry Java SDK**, for example, 7.1.0.10, then click **Next**.
6. Review the items to be installed, then click **Next** again.
7. Accept the license agreements, then click **Finish**.

   Note: If you get a security warning about the authenticity or validity of the software, click **OK**.
8. When the installation completes, restart SAP Mobile WorkSpace.
9. Click **Finish**.

Generating Java Object API Code

Use the Generate Code wizard to generate object API code for the SMP101 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 Mobile 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 SAP Mobile WorkSpace, open the **SMP101** mobile application project.

   In WorkSpace Navigator, right-click the **SMP101** folder and select **Open in Diagram Editor**.
2. In WorkSpace Navigator, expand **SMP101**. Under **Generated Code** add a folder named `BlackBerry`.

   The `Generated Code` directory was created automatically during the MBO tutorial.

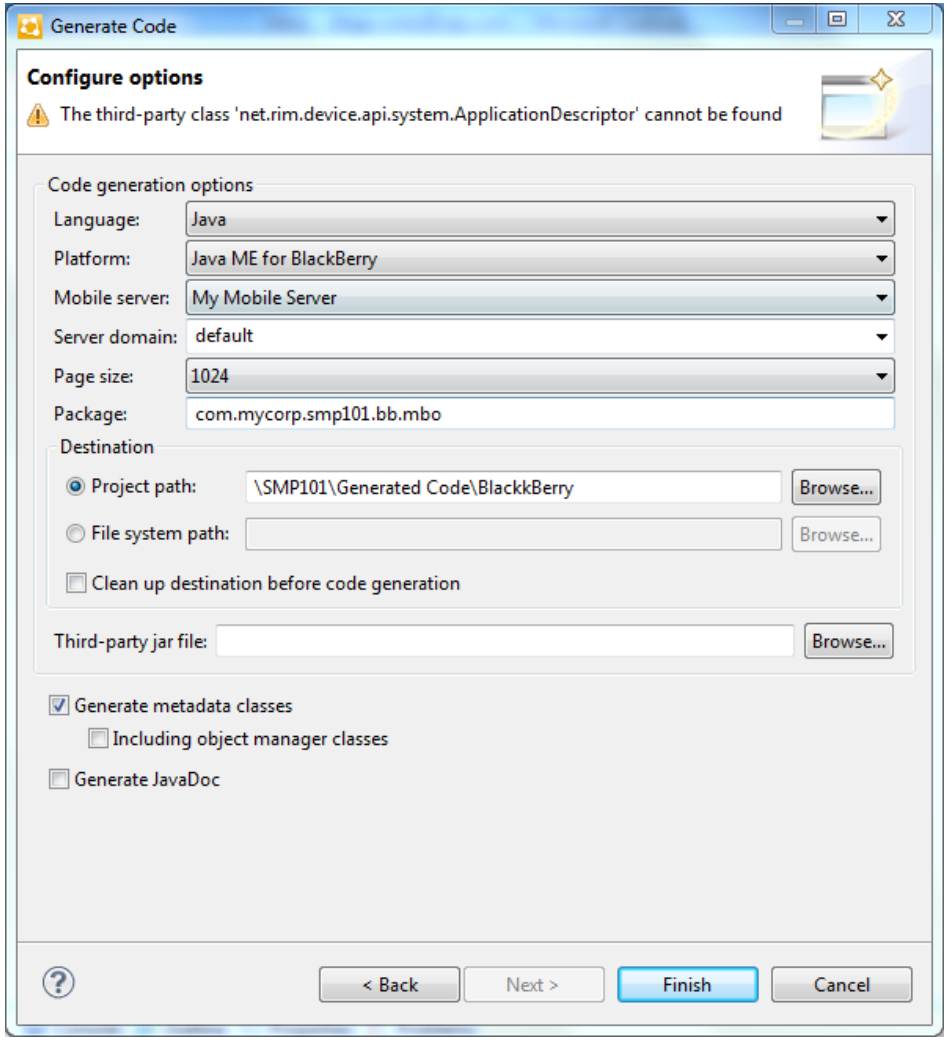
3. Right-click anywhere in the SMP101 - Mobile Application Diagram and select **Generate Code**.
4. In the Generate Code wizard, click **Next** to select the default configuration and continue without any changes.
5. In the Select Mobile Business Objects window, select the **Customer** MBO, then click **Next**.

   Ignore any warning about unresolved mobile business object dependencies. The warning appears because you selected the customer data only.
6. In the Configure Options window, specify these values and click **Finish**.

Option	Description
Language	Select **Java**.
Platform	Select **Java ME for BlackBerry**.
Mobile server	Select **My Mobile Server** (or the server to which you deployed the SMP101 project, if you used another).
Server domain	Accept default value.
Page size	Accept default value of 1024. The page size should be larger than the sum of all attribute lengths for any MBO that is included with all the MBOs selected, and must be valid for the database. If the page size is not set, the default page size is 4KB at runtime.
Package	Enter `com.mycorp.smp101.bb.mbo`. Bundling groups of related types into packages makes types easier to find and use, avoids naming conflicts, and controls access.
Project path	Enter `\SMP101\Generated Code \BlackBerry`.
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 dependent third-party class 'net.rim.device.api.system.ApplicationDescriptor' cannot be found.` You can ignore this warning; it does not affect code generation.

Option	Description
Generate Metadata Classes	Select for this tutorial to generate metadata for the attributes and operations of each generated client object.
Generate JavaDoc	Unselect for this tutorial. You are not creating automatic documentation from comments in the code.

Note: For more about the configuration options, see *Developer Guide: BlackBerry Object API Applications*.



You can ignore the error icons that appear in WorkSpace Navigator under `SMP101\Generated Code\BlackBerry\src\`, and the error in the Problems view: `net cannot be resolved`.

The error occurs because the package Java file called by `SMP101DB.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 `\BlackBerry\src\` folder.

Creating the BlackBerry Project

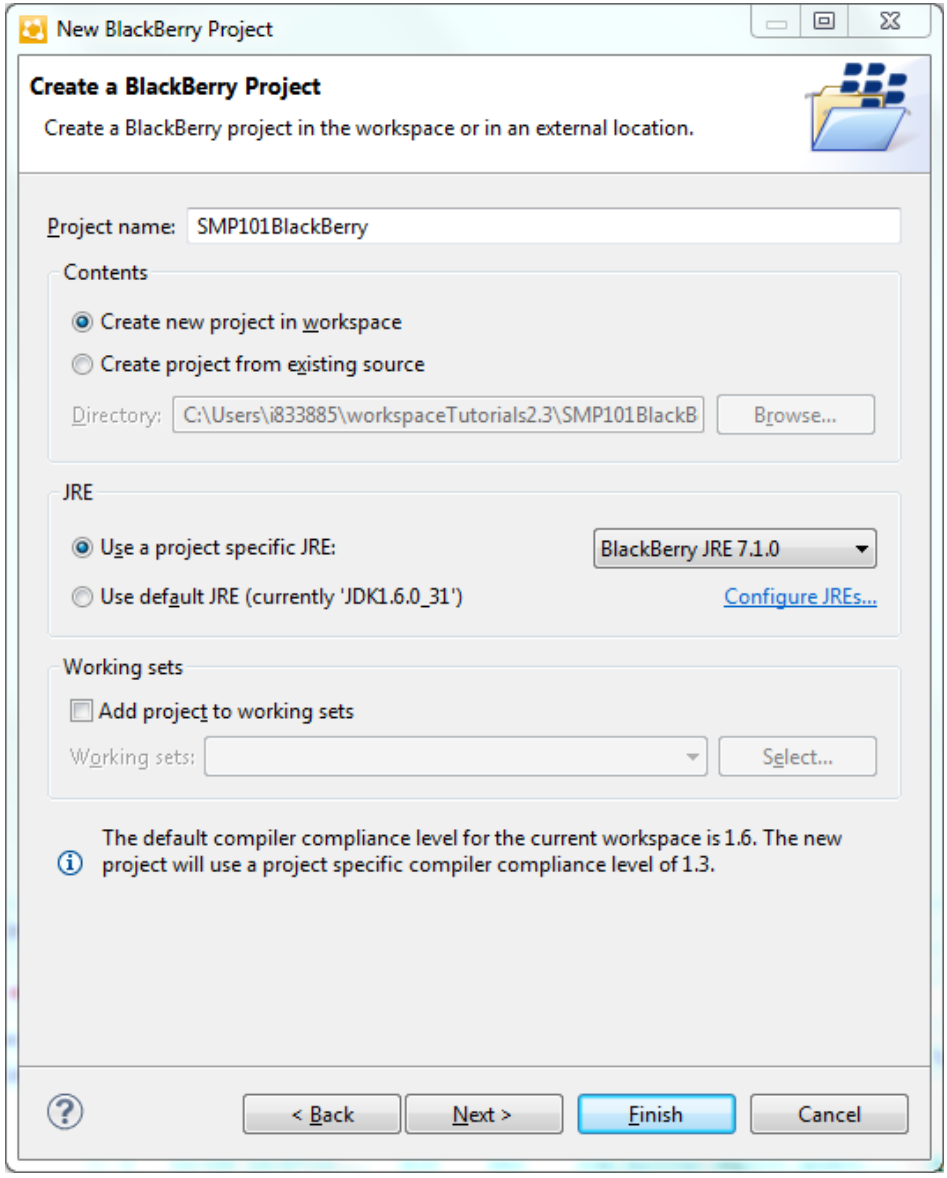
Create a new BlackBerry project in SAP Mobile WorkSpace. Add library resources to the project and set other application properties.

Prerequisites

To help create your project, and in a subsequent topic, build the user interface, download the SMP101 BlackBerry 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 SAP Mobile WorkSpace.
2. Select **File** > **New** > **Project**.
3. In the New Project window, select **BlackBerry Project** and click **Next**.
4. In the Create a BlackBerry Project window, enter `SMP101BlackBerry` for the project name and click **Next**.



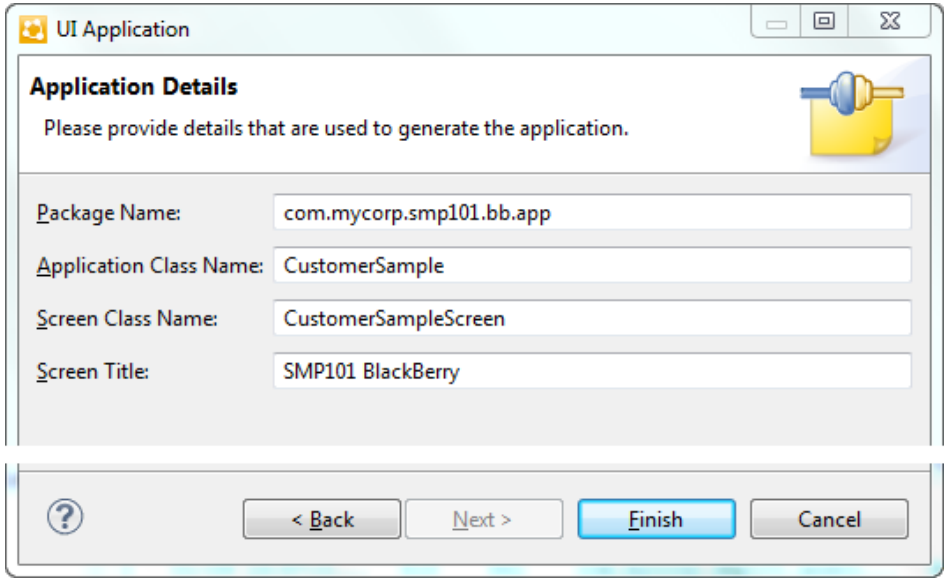
5. In the Java Settings page, modify the build path to the `sup_client2.jar` and `UltraLiteJ12.jar` files:
   a) Click the **Libraries** tab.
   b) Click **Add External JARs**.
   c) Browse to *SMP_HOME*`\MobileSDK23\ObjectAPI\BB`.
   d) Select the two JAR files, then click **Open**.

e) Click the **Order and Export** tab.
f) Make sure the `sup_client2.jar` is selected.

Select only `sup_client2.jar` in this tab because it is a third-party JAR file that does not have a `cod` file. You add the `UltraLiteJ12.cod` file in *Copying SAP Mobile Platform Files to the Sample Project*.

The `sup_client2.jar` file is for the ObjectAPI client, and `UltraLiteJ12.jar` is the client database library.

6. Click **Next**.
7. In the Templates window, accept the default BlackBerry Application and click **Next**.
8. In the Application Details window, enter these values and click **Finish**.
   - Package Name – `com.mycorp.smp101.bb.app`
   - Application Class Name – `CustomerSample`
   - Screen Class Name – `CustomerSampleScreen`
   - Screen Title – `SMP101 BlackBerry`



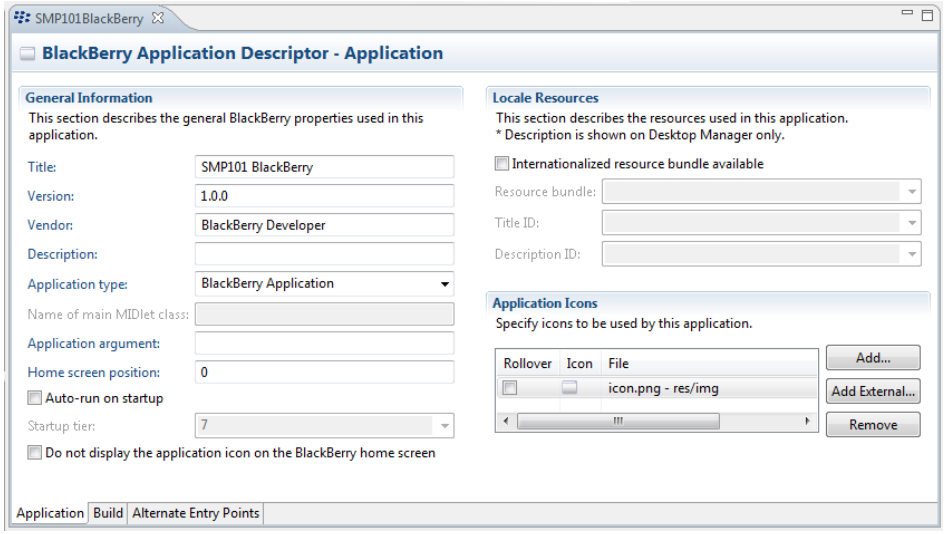
9. If prompted to use a different perspective, click **No**.

Configuring BlackBerry Application Properties

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

1. In WorkSpace Navigator, expand the **SMP101BlackBerry** project.
2. Double-click the `BlackBerry_App_Descriptor.xml` file.

3. Select the **Application** tab.
4. In the Title, enter `SMP101 BlackBerry`.
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. Select **File > Save**.

Copying SAP Mobile Platform Files to the Sample Project

Copy SAP Mobile Platform BlackBerry runtime files and the object API code you generated using the Generate Code wizard to the SMP101BlackBerry project.

1. In Windows Explorer, browse to *SMP_HOME*`\MobileSDK23\ObjectAPI\BB`.
2. Copy the `UltraLiteJ12.cod` file.

   The client database library is required.
3. Paste the copied COD file into the BlackBerry Java Plug-in simulator directory, located in *SMP_HOME*`\MobileSDK23\Eclipse\plugins\net.rim.ejde.componentpack`*<version>*`\components\simulator`.
4. Generate the application key file `SMP.key` using `BlackBerrySigningAuthorityTool_1.0.exe`, and copy it to the project root folder.

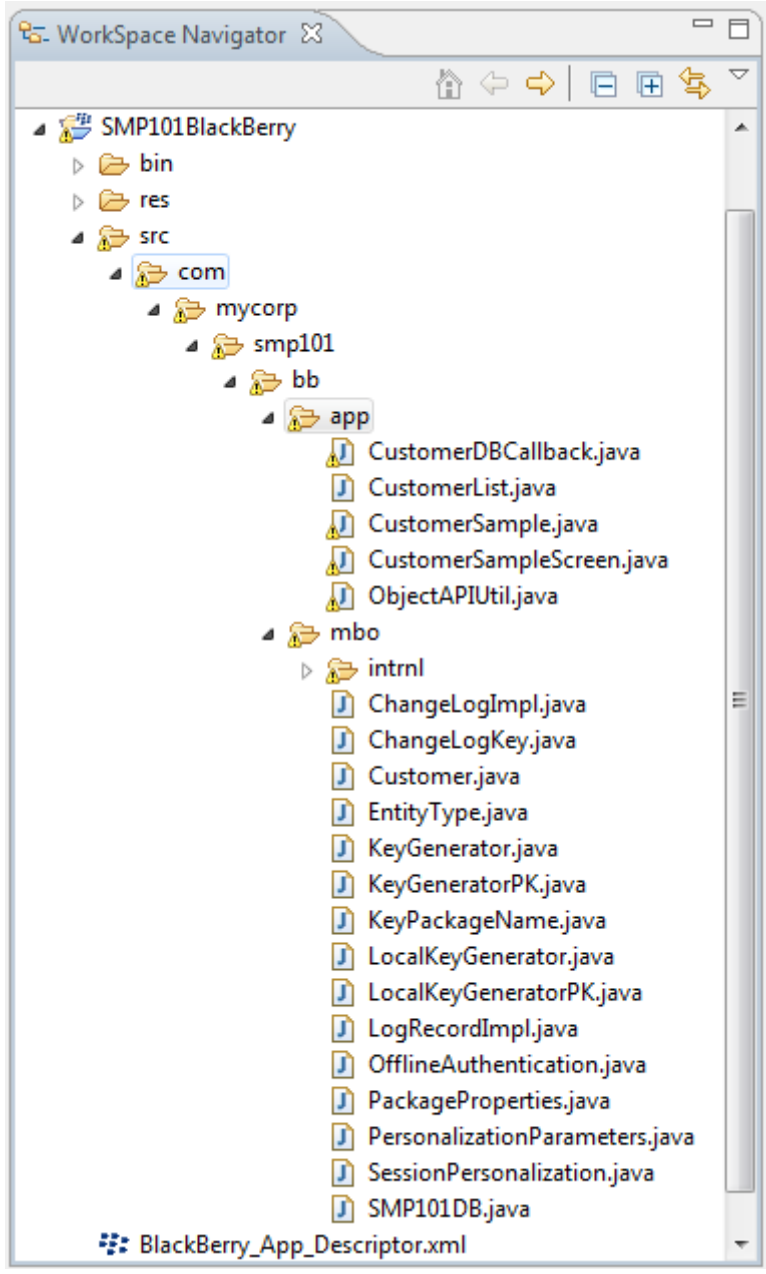
   See BlackBerry Password Based Code Signing Authority at: *http://supportforums.blackberry.com/t5/Java-Development/Protect-persistent-objects-from-access-by-unauthorized/ta-p/524282*.

5. In WorkSpace Navigator, select the **SMP101** project and copy the `com` folder under `\Generated Code\BlackBerry\src`.
6. In WorkSpace Navigator, go to the **SMP101BlackBerry** project and paste the `com` folder in to the `src` directory. Select **Yes to All** to copy over existing folders.

Creating the User Interface

Copy the Java code files, which provide the functionality and layout of the user interface, from the SMP101 BlackBerry Project example project archive to the SMP101 BlackBerry project.

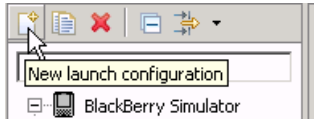
1. In Windows Explorer, browse to the directory where you saved the SMP101 BlackBerry Project example project file.
2. Copy these Java files:
   - `CustomerDBCallback.java` – implements the CallbackHandler to demonstrate how to track changed entities in the onSynchronize callback method.
   - `CustomerList.java` – 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.
3. In WorkSpace Navigator, go to **SMP101BlackBerry** and expand `\src\com\mycorp\smp101\bb\app`, then paste the copied Java files, copying over any existing files.
4. Modify the host name or IP address in the `ObjectAPIUtil.java` file to point to the SAP Mobile Server.
   a) In WorkSpace Navigator, expand the **SMP101BlackBerry** project.
   b) Under the `\src\com\mycorp\smp101\bb\app` folder, double-click the `ObjectAPIUtil.java` file.
   c) Modify the host name or IP address, and verify the username and password are valid.


WorkSpace Navigator
SMP101BlackBerry
bin
res
src
com
mycorp
smp101
bb
app
CustomerDBCallback.java
CustomerList.java
CustomerSample.java
CustomerSampleScreen.java
ObjectAPIUtil.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
SessionPersonalization.java
SMP101DB.java
BlackBerry_App_Descriptor.xml

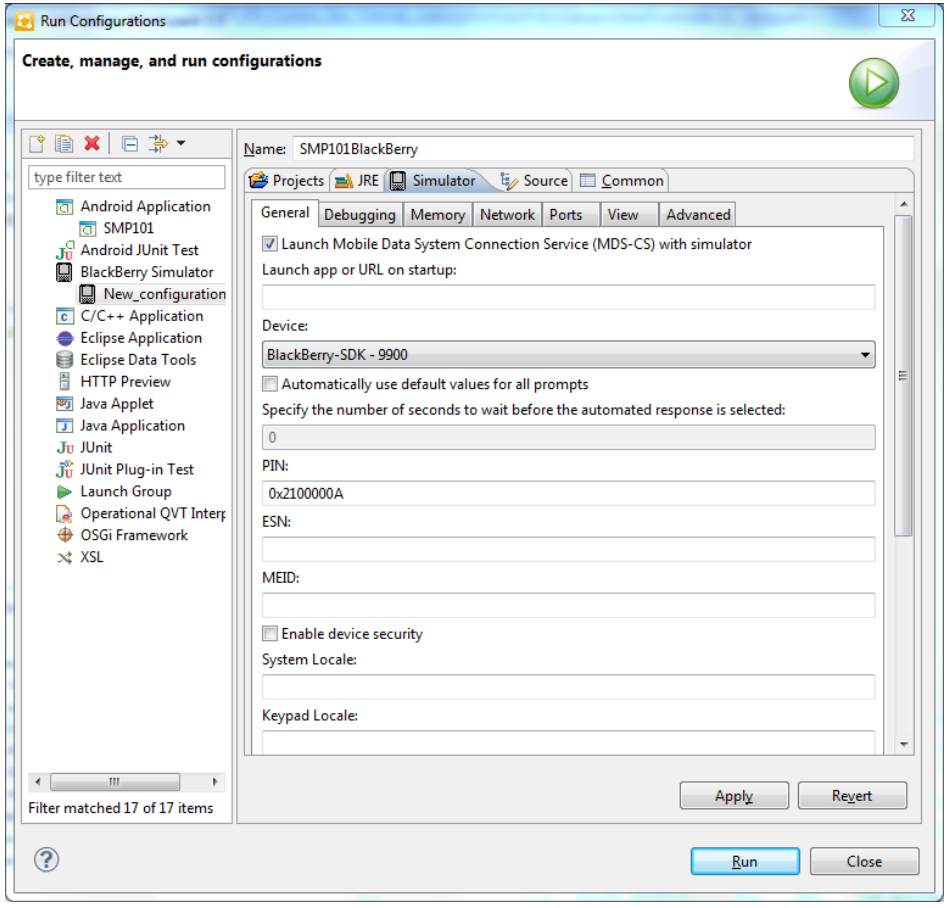
Creating a Launch Configuration for the Project

Create a new launch configuration for the SMP101BlackBerry project. The configuration specifies how the application launches, and defines the target BlackBerry platform.

1. In WorkSpace Navigator, right-click the **SMP101BlackBerry** project, and select **Run As > Run Configurations**.
2. Select **BlackBerry Simulator** and the **New launch configuration** icon.



3. In the Projects tab, select the **SMP101BlackBerry** project.
4. In the Name field, enter `SMP101BlackBerry`.
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 **BlackBerry-SDK - *Version*** as the device.



If you are running the BlackBerry JDE 7.0 with the BlackBerry Java Plug-in v1.5 instead of the BlackBerry plug-in for Eclipse, you must launch the Mobile Data System (MDS) manually for the BlackBerry simulator to run by following the sub-task below. Otherwise, proceed to step 7.

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, which has been fixed in BlackBerry JDE 7.1, that the simulator cannot launch MDS.

a) From a command prompt, go to `<Eclipse_InstallDir>\plugins\net.rim.ejde.componentpack7.0.0_7.0.0.33\components\MDS`.

b) Start MDS by entering:

```
start run.bat
```

c) Press any key to bypass this message.

```
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 now launch the BlackBerry simulator. Leave the command window open.

7. In the **Network** tab, verify that **Disable Registration** is selected. If it is not, the sample application cannot get data from SAP Mobile Server.
8. Click **Apply**, then **Close**.

Testing the Device Application on the BlackBerry Simulator

Run the SMP101BlackBerry application on the BlackBerry simulator, and change customer information to update the interface.

1. In WorkSpace Navigator, right-click **SMP101BlackBerry** 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 you have run the simulator, cancel the Setup screen.

2. On the main window, click **All** to access the applications screen, then scroll until you see the SMP101BlackBerry application.

3. Click **SMP101 BlackBerry** to launch the application.

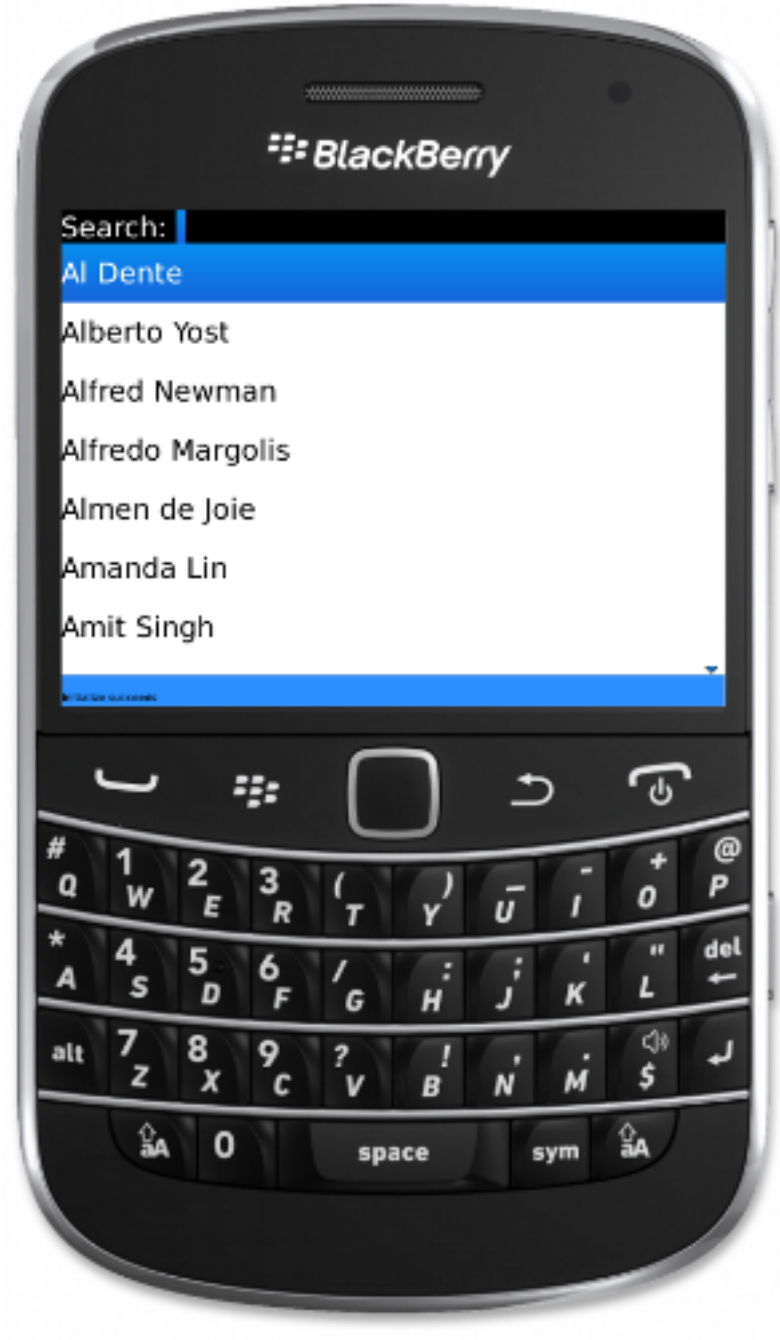
The application registers and synchronizes data from the server in the background.

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

```
SynchronizationGroup
sg=SMP101DB.getSynchronizationGroup("default");
sg.setEnableSIS(true);
sg.save();
```

After the application initializes, the device application shows the SMP101BlackBerry 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 SMP101DB.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. Select the customer to update.

5. Change the first name of the customer and click **submit**.

The **submit** button maps to the update operation of the customer mobile business object. The synchronize operation uses SMP101DB.beginSynchronize in the background so the user interface is not affected. When the application synchronizes, any pending operations are uploaded to SAP Mobile Server.

Any back-end changes initiate notifications from the server. The device application uses a ChangeLog API, specifically ObjectList changeLogs = SMP101DB.getChangeLogs(query);, to query those managed items and update the user interface if needed.

The customer list appears, and shows the name change you made.

6. Close the simulator to stop the SMP101BlackBerry application.

Learn More About SAP Mobile Platform

Once you have finished, try some of the other samples or tutorials, or refer to other development documents in the SAP Mobile 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 SAP Mobile 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 SAP Mobile 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: SAP Mobile Server Runtime > Management API* – customize and automate system administration features.

Javadoc and HeaderDoc are also available in the installation directory.

Index

T

U