SAP®

Tutorial: Hybrid App Package Development

SAP Mobile Platform 2.3

Contents

SAP Mobile Platform Tutorials

The SAP® tutorials demonstrate how to develop, deploy, and test mobile business objects, device applications, 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*

Getting Started with SAP Mobile Platform

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.

1. *Installing SAP Mobile Platform*

    Install SAP Mobile SDK and SAP Mobile Platform Runtime.

2. *Starting SAP Mobile Platform Services*

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

3. *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.

4. *Connecting to SAP Control Center*

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

5. *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.

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

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.

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 Database Mobile Business Object

Create a database mobile business object and deploy it to SAP Mobile Server.

Prerequisites

1. Complete the tasks in *Getting Started with Mobile Platform*.
2. (Optional) To use as a reference and to copy source code from when completing this tutorial, download the *Hybrid App Package* example project (source code only) from the SAP® Community Network: *http://scn.sap.com/docs/DOC-8803*.

Task

Develop the database mobile business object by:

1. *Creating the HybridApp101 Mobile Application Project*

   A mobile application project is the SAP Mobile WorkSpace container for the mobile business objects that form the business logic of mobile applications.

2. *Creating the TravelRequest Database Table*

   Create the database table to be used by the TravelRequest mobile business object.

3. *Creating the TravelRequest Mobile Business Object*

   Use the TravelRequest database table to create a mobile business object.

4. *Deploying the HybridApp101 Mobile Application Project*

   Deploy the project that contains the TravelRequest mobile business object to the server.

Creating the HybridApp101 Mobile Application Project

A mobile application project is the SAP Mobile WorkSpace container for the mobile business objects that form the business logic of mobile applications.

Create a mobile application project before creating its mobile business objects.

1. In SAP Mobile WorkSpace, click **File > New > Mobile Application Project**.
2. Enter `HybridApp101` as the project name.
3. Click **Finish**.

Creating the TravelRequest Database Table

Create the database table to be used by the TravelRequest mobile business object.

Create the database table from a SQL file:

1. In Enterprise Explorer, open the `Database Connections` folder, right-click `My Sample Database`, and choose **Connect**. If **Connect** is not available in the context menu, the database is already connected.

   See *Connecting to the Sample Database* in *Tutorial: Mobile Business Object Development* for more information.

2. From the main menu, select **File > New > Other.**
3. Select **SQL Development > SQL File**, and click **Next**.
4. Enter or select the values shown, then click **Finish**:

Field	Value
Parent folder	Select the **HybridApp101** project.
File name	Enter `TravelRequest`.
Database server type	Select **Sybase_ASA_12.x**.
Connection profile name	Select **My Sample Database**.
Database name	Select **sampledb**.
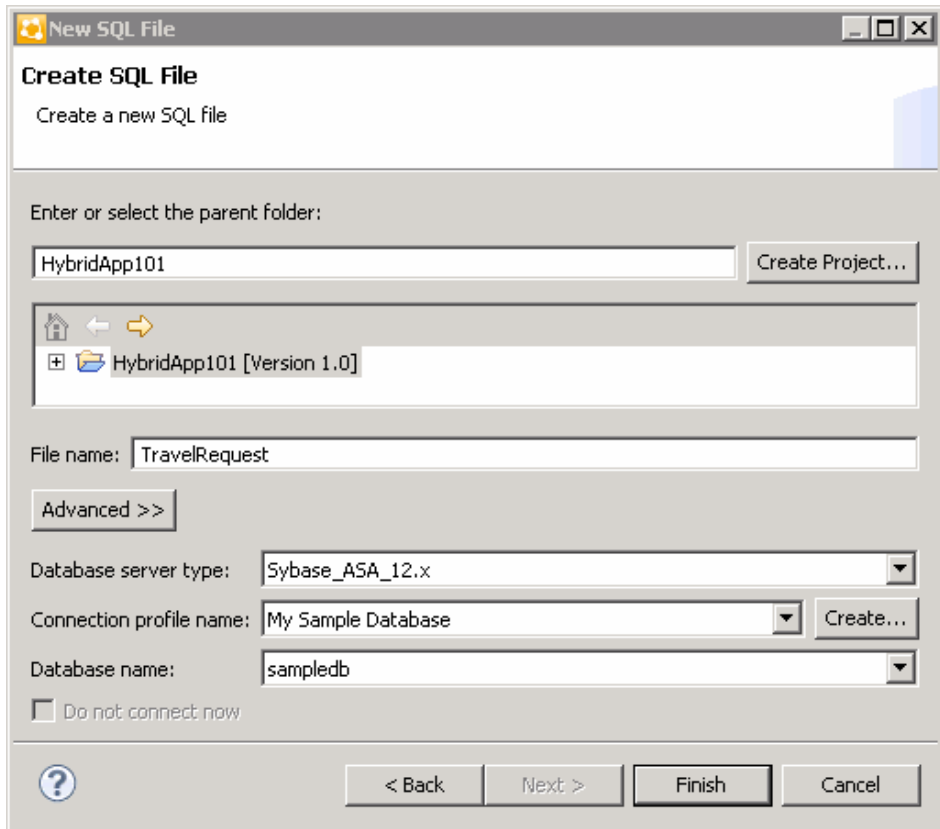
5. If the new file does not open in the SQL File Editor, open it by double-clicking
   `HybridApp101\TravelRequest.sql` in Workspace Navigator. In
   `TravelRequest.sql`, copy and paste this code:

```
CREATE TABLE TravelRequest (
  trvl_Id                       integer NOT NULL DEFAULT
autoincrement,
  trvl_Date                     date NULL,
  trvl_Loc                      varchar(20) NULL,
  est_Cost                      float NULL,
  purpose                       varchar(200) NULL,
  trvl_Status                   varchar(20) NULL,
  st_Cmmnt                      varchar(200) NULL,
  PRIMARY KEY CLUSTERED (trvl_Id)
);
```

6. Save `TravelRequest.sql` and close the editor.

7. In the WorkSpace Navigator, expand **HybridApp101**. Right-click **TravelRequest.sql**
   and choose **Execute SQL Files**.

   SAP Mobile WorkSpace creates a table called TravelRequest in the `sampledb` database.

---

8. In Enterprise Explorer, expand **sampledb**, expand the **Tables** folder, and verify that the new TravelRequest table is added.

    If the table is not visible, right-click the **Tables** folder and choose **Refresh**.

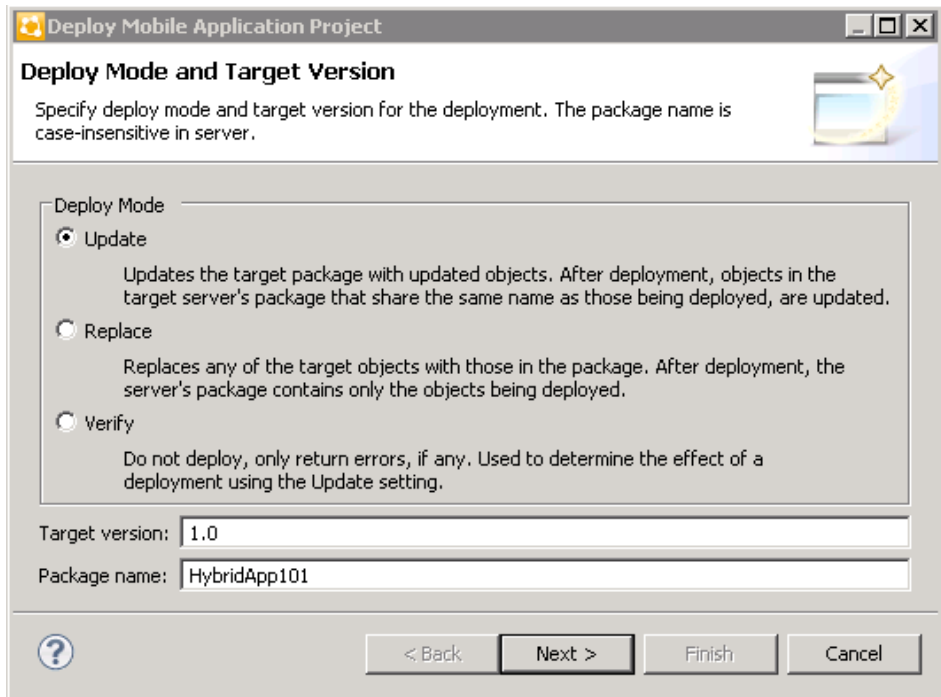# Creating the TravelRequest Mobile Business Object

Use the TravelRequest database table to create a mobile business object.

1. In Enterprise Explorer, open the `Database Connections\My Sample Database\sampledb\Tables` folder.
2. Drag the TravelRequest table from the Enterprise Explorer onto the HybridApp101 Mobile Application Diagram.
3. In the Quick Create wizard, accept the default settings and click **OK**.
    An MBO is created in the Mobile Application Diagram.
4. Save the diagram.

Deploying the HybridApp101 Mobile Application Project

Deploy the project that contains the TravelRequest mobile business object to the server.

1. Right-click in the HybridApp101 Mobile Application Diagram and choose **Deploy Project**.
2. In the first page of the Deploy Mobile Application Project wizard, accept the default options and click **Next**.

3. In the Contents page, select the **TravelRequest** mobile business object, and click **Next**.

4. In the Package User-defined Classes page, click **Next**.

   Note: The Package User-Defined Classes window appears by default in the advanced developer profile.

5. In the Target Server page, select **My Mobile Server** in the list of available servers. If the **Next** button is not enabled, click **Connect** to connect to the server. Then click **Next**.

6. In the Assign package to applications page, click **Next**.

7. In the Server Connection Mapping page, click **My Sample Database** under **Connection profile**. In the same row, choose **sampledb** under **Server connection**.
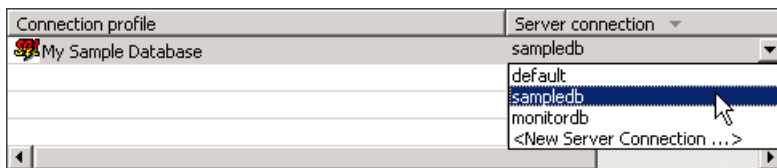
8. Click **Finish**, and click **OK** to dismiss the Deployment Status window.

9. In Enterprise Explorer, open `Mobile Servers\My Mobile Server\Domains \default\Packages\hybridapp101:1.0\Mobile Business Objects`, and verify that the folder contains the deployed `TravelRequest` MBO.

Developing a Hybrid App Package

Define the logic and screens for your Hybrid App, generate a package to run on specific device platforms, and deploy it to the server.

This tutorial has been developed using SAP Mobile Platform 2.2 SP02 and this device-specific software:

* Mac OS X 10.7.4, iOS SDK 6.0, Xcode 4.5.1, and the iOS 6.0 simulator
* Android 4.0.3 SDK and simulator
* BlackBerry 6.0 JDE and MDS-CS simulator
* Windows XP ActiveSync 4.5, Windows Device Emulator (Standalone Release), and Windows Mobile 6.5.3 Professional Emulator

If you use a different version of SAP Mobile Platform, some steps may vary.

1. *Registering Application Connections in SAP Control Center*

    Make the application available on specific devices.

2. *Developing a Hybrid App Screenflow*

    Use the Hybrid App Designer to create the screenflows for the TravelRequest mobile business object.

3. *Generating Code for a Hybrid App Package*

    Create a Hybrid App package, customize it, and deploy it to SAP Mobile Server to make it available to device clients.

4. *Testing the Hybrid App on the iPhone Simulator*

    Build a sample Hybrid Web Container in the Xcode IDE, and run the Travel Request application on an iPhone simulator.

5. *Testing the Hybrid App on the Android Emulator*

    Install and configure the Android simulator software and use it to run the Travel Request application.

6. *Testing the Hybrid App on the BlackBerry Simulator*

    Install and configure the BlackBerry simulator, and use it to run the Travel Request application.

7. *Testing the Hybrid App on the Windows Mobile Emulator*

    Install and configure the Windows Mobile synchronization software and emulator, and use it to run the Travel Request application.

8. *Verifying the Data on the Back-end Database*

    After submitting a travel request in an emulator, verify that the information is updated in the database.

Registering Application Connections in SAP Control Center

Make the application available on specific devices.

1. Log in to SAP Control Center using the supAdmin user name and the current password.
2. In SAP Control Center, select **View > Select > Mobile Server Cluster Management View**.
3. Click **Applications** in the left pane. In the right pane, click **Application Connections**.
4. Register an application connection for each device that you will test in the tutorial:
   a) Click **Register**.
   b) In the Register Application Connection window, specify:
      - User – enter the name of the user who will activate and register the Hybrid App. For this tutorial, where several different emulators are tested, enter a unique user name for each device.
      - Template – select the **HWC** template.
      - Application ID – accept the default value.
      - Security configuration – accept the default value.
      - Logical role – accept the default blank field.
      - Domain – accept the default blank field.
      - Activation code length – accept the default value.
      - Activation expiration (hours) – accept the default value.
      - Specify activation code – select this option and enter a value, such as 123. If you do not specify an activation code, one is generated automatically. The user must enter the activation code when logging in to the application. The value can contain letters A – Z (uppercase or lowercase), numbers 0 – 9, or a combination of both.
   c) Click **OK**.

   SAP Control Center adds the connection to the Application Connections table. The following example shows three registered connections, each for a different device. The connections do not show a Device Type or Device ID yet because the devices are not currently connected: these fields are automatically loaded when the specified user accesses the connection.

Developing a Hybrid App Screenflow

Use the Hybrid App Designer to create the screenflows for the TravelRequest mobile business object.

Prerequisites
Complete *Deploying the HybridApp101 Mobile Application Project* on page 12.
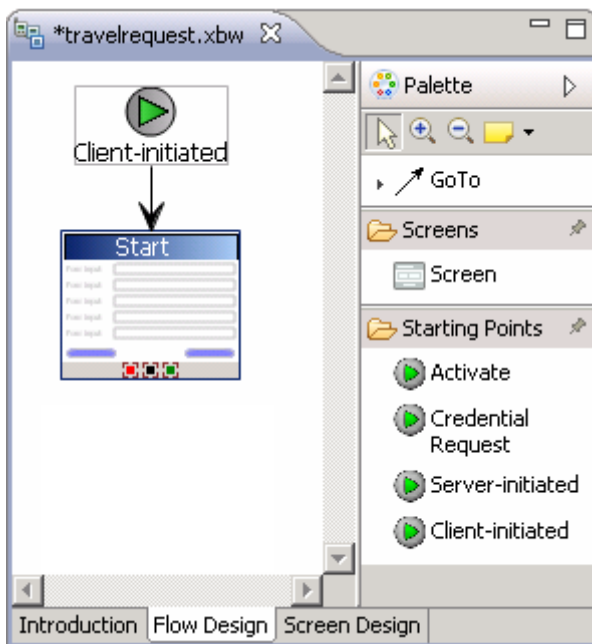
See also
• *Generating Code for a Hybrid App Package* on page 28

Creating a New Screenflow

Use the Hybrid App Designer wizard to create the initial screen.

1. In the Hybrid App Designer, click **File > New > Hybrid App Designer**.
2. Select the HybridApp101 folder and enter travelrequest.xbw as the Hybrid App name.
3. Click **Next**.
4. In the Starting Points page, select **Can be started, on demand, from the client**, and click **Finish**.
5. In the Hybrid App Designer, click the **Flow Design** tab.
   In the screenflow, a Client-initiated screen is connected to a Start screen.
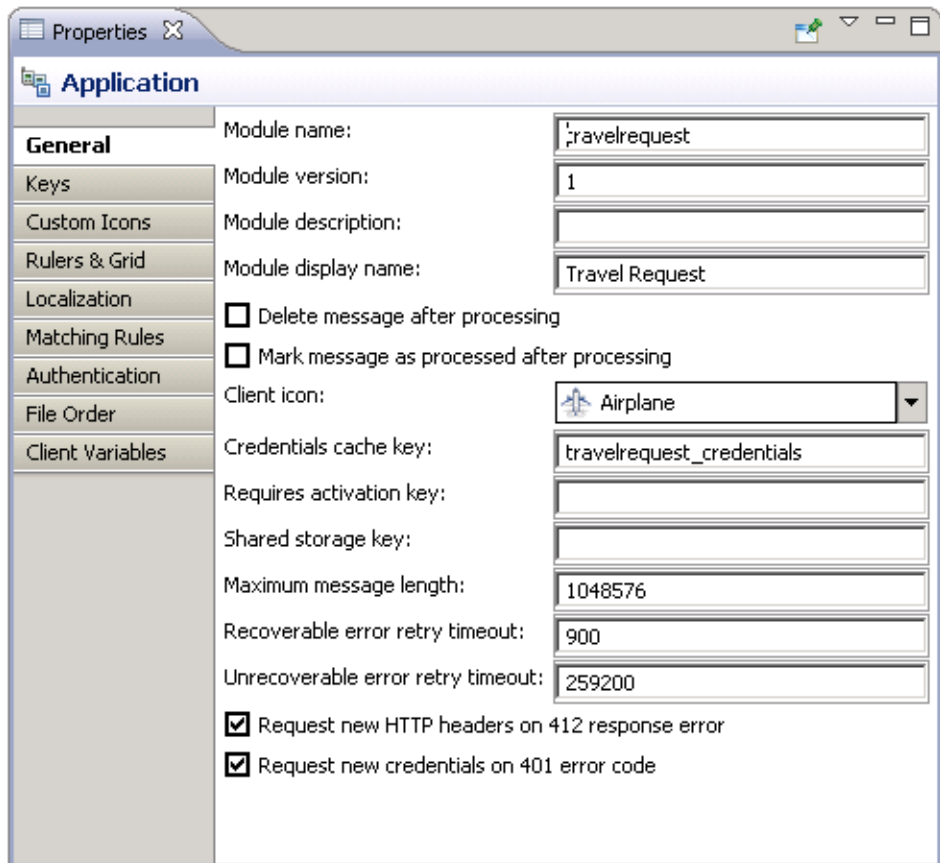
See also
- *Customizing the Module Display Name and Client Icon* on page 18
- *Adding the Travel Request Screen* on page 19
- *Modifying Menu Text and Adding a Key* on page 21
- *Making Screen Content Collapsible* on page 25

Customizing the Module Display Name and Client Icon

Change the generated Hybrid App name and icon.

You can change the default appearance of the generated Hybrid App. For this tutorial, change the Hybrid App name to make it more readable on the mobile device, and choose a different client icon.

1. In the **travelrequest.xbw** Flow Design, click a blank part of the application.
2. In the General page of the Application Properties view, change the value of **Module display name** (not **Module name**) to `Travel Request`.
3. In Client icon, change the default image from `New Hybrid App` to `Airplane`.
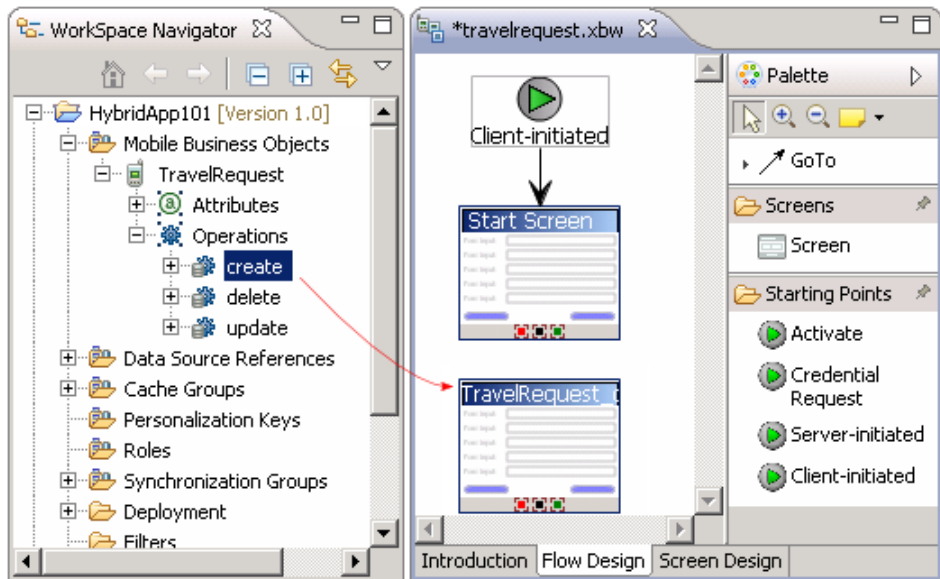
See also

Adding the Travel Request Screen

Add a screen for Travel Request input.

1. In Workspace Navigator, open `HybridApp101\Mobile Business Objects \TravelRequest\Operations`.
2. To add the TravelRequest_create screen, drag a **create** operation from the Navigator onto the Flow Design.

---

3. Change the TravelRequest screen title to make it more readable.

   a) In the Flow Design, click the TravelRequest screen to display its properties.

   b) In the Properties view General page, change the Name property from
      TravelRequestcreate to `Travel Request`.
      The key name automatically changes to Travel_Request.

4. Add a GoTo connection from the Start Screen to the Travel Request screen:

   a) In the Palette, click **GoTo**.

   b) Click the **Start** Screen, and while holding the mouse down, drag the connection to the
      Travel Request screen.

   A line with an arrow connects the two screens.
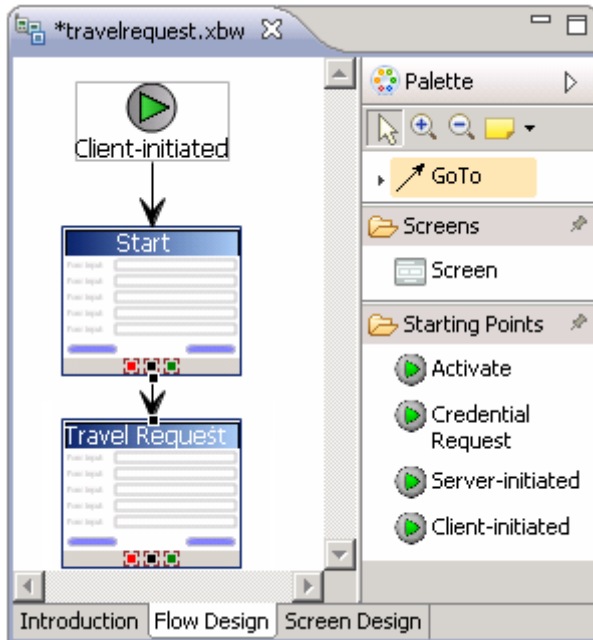
5. Double-click the **Travel Request** screen to open its screen design.

6. In the Screen Design page, modify the label of the Trvl Status edit box to make it more readable:

   a) In the Travel Request screen, select **Trvl Status**.

   b) In the Properties view, change the Label to Status:.

   c) In the Travel Request screen, select the **St Cmmnt** edit box.

   d) In the Properties view, change the Label to Comment:.

7. Press **Ctrl+S** to save your screen design.

See also

- *Creating a New Screenflow* on page 17
- *Customizing the Module Display Name and Client Icon* on page 18
- *Modifying Menu Text and Adding a Key* on page 21
- *Making Screen Content Collapsible* on page 25

Modifying Menu Text and Adding a Key

Change a menu item label and add introductory text for the start screen using a key.

1. Open the Start screen design.
   For example, in the tool bar, choose **Start** in the screen selection control:
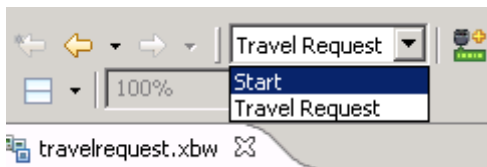
2. In the Screen Design, shorten the menu text to fit better on the mobile device screen.

   a) In the menu, click **Open Travel Request**.

   b) In the Menu Item Properties view, change **Name** from Open Travel Request to Next.

3. Add an HtmlView control for the Start screen.

   The HtmlView control defines basic HTML markup for the screen. It is a container to which you can add other controls from the Palette, as well as your own custom HTML code.

   a) In the Palette, select the **HtmlView** control.

   b) Click in the Start screen to add the control:
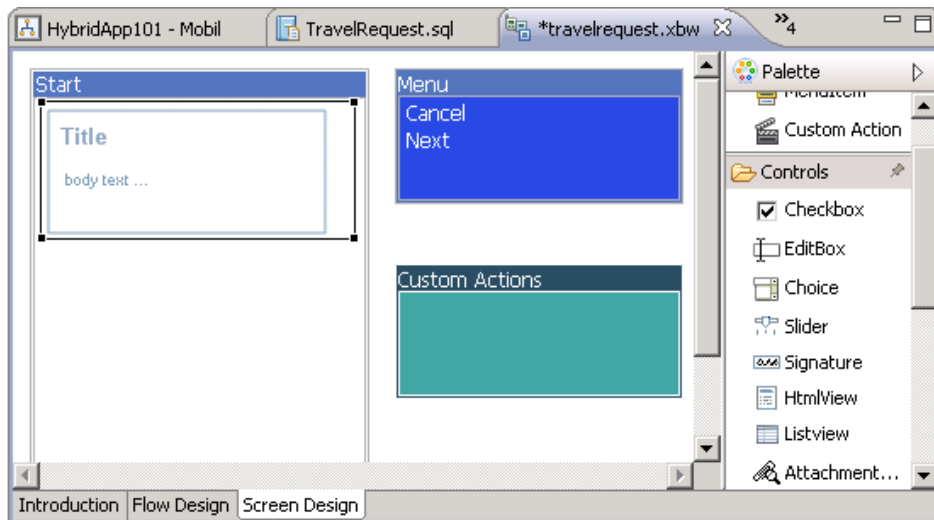
4. Click the **Flow Design** tab.

5. Select the **Client-initiated** starting point.

   The Properties view shows all the keys in the create method of the TravelRequest mobile business object, which were automatically created when you dragged and dropped the create operation on the screen. If the Properties view does not open, right-click the **Client-initiated** screen and choose **Show Properties View**.
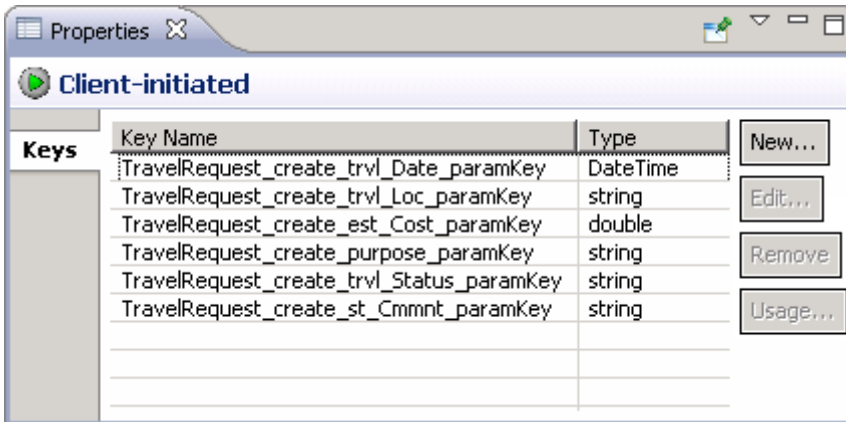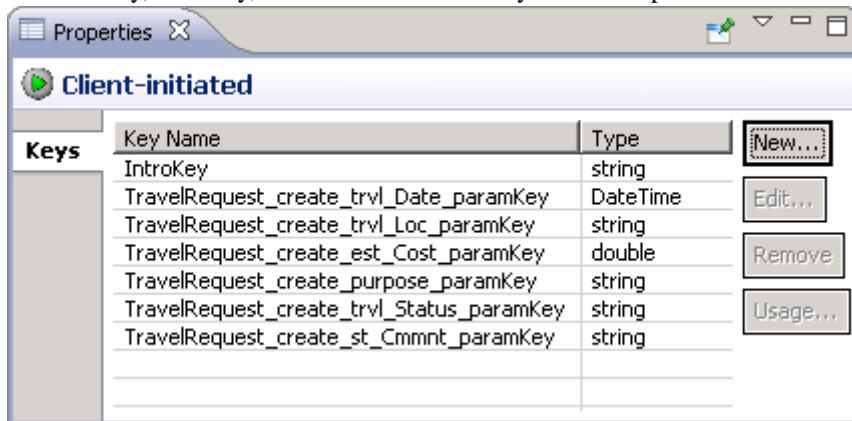
6. Create a new key:
   a) In the Client-initiated Properties view, click **New**.
   b) Enter these values:
      - Name – `IntroKey`
      - Type – `string`

   **Note:** Only string keys can be associated with an HtmlView control.

   c) Click **OK**.

   The new key, IntroKey, is added to the list of keys in the Properties view.

7. Define the default value for the new key:
   a) In the Flow Design, double-click the Start screen to open its screen design.
   b) In the Start Screen, select the **HtmlView** control to display its properties.
   c) In HTMLView Properties, under Input Data Binding, choose **IntroKey**.
   d) In Default value, enter this message code:
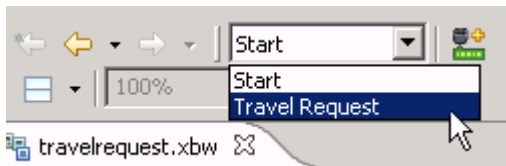
```
<H2>Travel Request App</H2>This application enables vacation
requests to be sent for approval.<br>
```
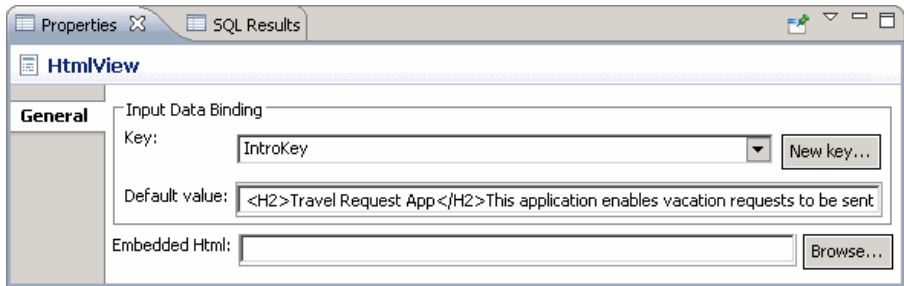
8. View the current parameter settings of the Travel Request screen:

   a) Switch to the Travel Request screen design.

      For example, choose **Travel Request** in the screen selection control:

   b) In the menu, select the **Create** operation to display its properties.

   c) In Properties, click **Parameter Mappings**.

      The parameter mapping table shows the mobile business object parameters for the operation and their keys.
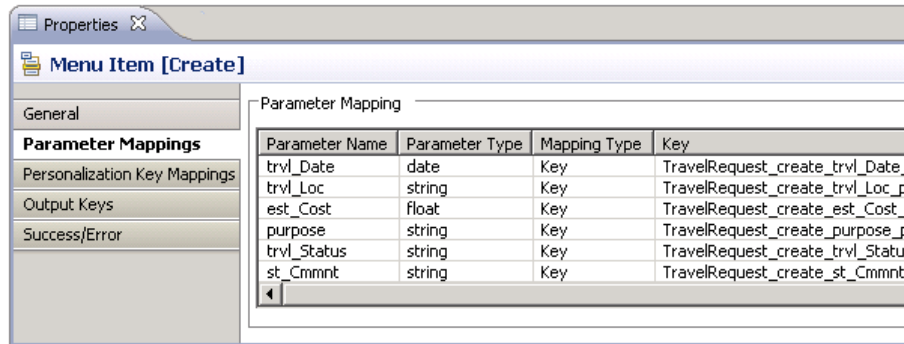
See also

- *Creating a New Screenflow* on page 17
- *Customizing the Module Display Name and Client Icon* on page 18
- *Adding the Travel Request Screen* on page 19
- *Making Screen Content Collapsible* on page 25

Making Screen Content Collapsible

Enable users to expand and collapse screen elements.

By default, content in a Collapsible control is initially hidden. The application user can expand the control and show the content by clicking the plus sign icon beside it. To hide the content again, the user clicks the same icon (now changed to a minus sign). Collapsible controls can be useful for simplifying the initial screen, while giving the user control over what content to show.

An Accordion control enables you to associate two or more Collapsible controls. By default, only one Collapsible control in the group is expanded at a time: if the user expands one Collapsible control, all the others in the Accordion control are collapsed.

Note: Accordion and Collapsible controls are not supported in Windows Mobile devices: their content is static, not collapsible.

1. In the Designer, open the Travel Request screen.
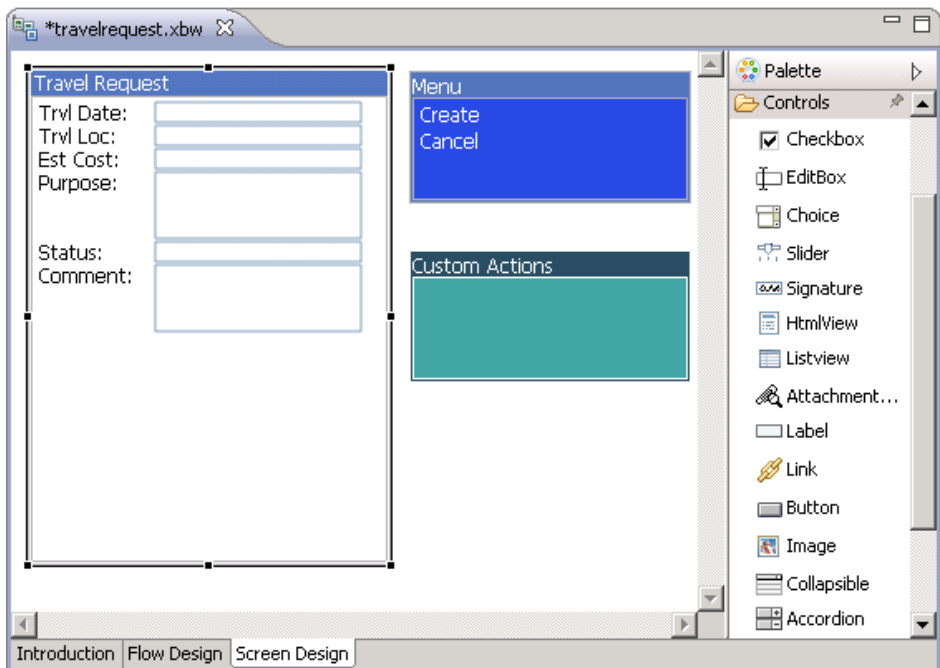
2. In the Screen Design Palette, click **Accordion**. Then click in a blank area of the Travel Request screen to add the control.
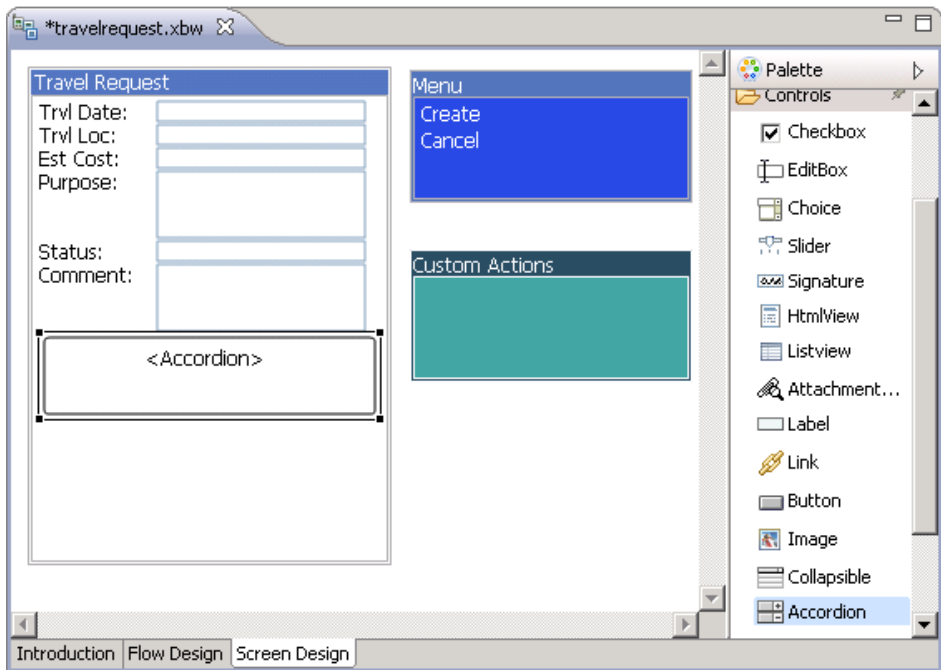
3. Add a Collapsible control to the Accordion:
   a) In the Screen Design Palette, click **Collapsible**.
   b) Click in the Accordion control.
   c) In Collapsible properties, enter `Basic Information` as the Label.
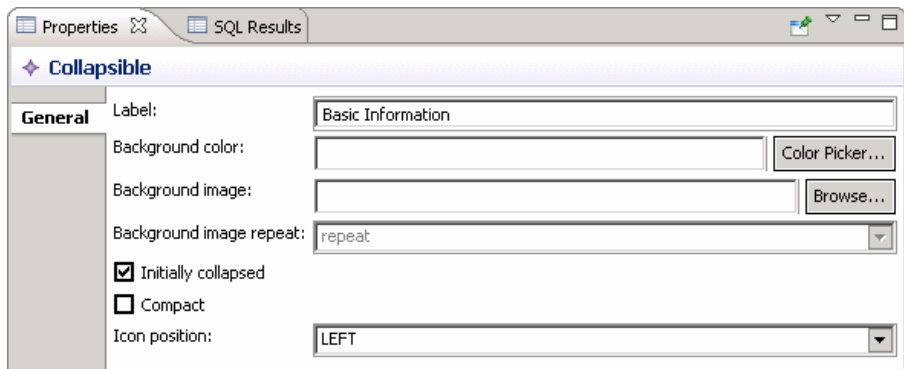
4. Repeat the previous step to add a second Collapsible control below the first one, specifying the label `More Information`.

   Because both Collapsible controls are in the same Accordion control, the application user can expand either one, but not both at the same time.
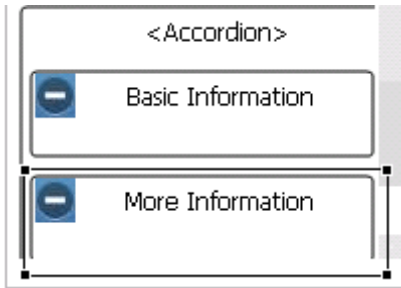
5. Move the **Trvl Date**, **Trvl Loc**, and **Est Cost** controls from the main part of the screen to the **Basic Information** control.

   For example, select the three controls and press **Ctrl+X** to cut them, then select the **Basic Information** control and press **Ctrl+V** to paste them.

6. Repeat the previous step to move the **Purpose**, **Status**, and **Comment** controls to the **More Information** Collapsible.
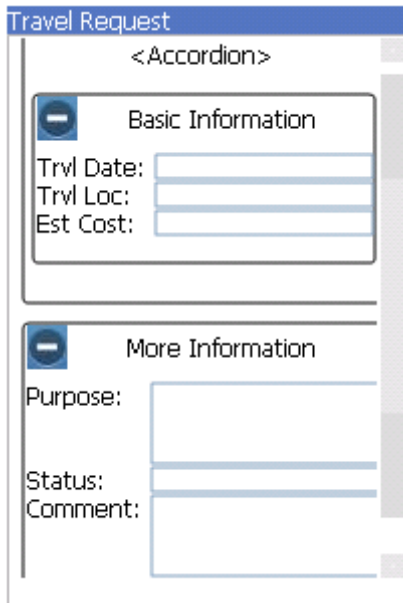
7. Press **Ctrl+S** to save the screen design.

See also

- *Creating a New Screenflow* on page 17
- *Customizing the Module Display Name and Client Icon* on page 18
- *Adding the Travel Request Screen* on page 19
- *Modifying Menu Text and Adding a Key* on page 21

# Generating Code for a Hybrid App Package

Create a Hybrid App package, customize it, and deploy it to SAP Mobile Server to make it available to device clients.

Use the Hybrid App Package Generation wizard to generate files for the Hybrid App package, deploy the generated package files to the server, and assign the package to one or more devices. Then perform a simple customization by adding a header image to the generated package.

1. In SAP Mobile WorkSpace, if the Hybrid App Designer is not open, expand `HybridApp101` in the WorkSpace Navigator and double-click **travelrequest.xbw**.

2. In the SAP Mobile WorkSpace toolbar, click the 🖳 Generate Hybrid App button.

3. Specify these options:

Option	Description
Mobile Server profile	Accept the default profile, `My Mobile Server`, to associate with the package.
Generate package	Accept the default selection of this option, **Update generated code**, and **Generate into the project**.
Deploy to a Mobile Server as a replacement	Accept the default selection of this option.
Assign hybrid app to user(s)	Select this option and click **Get Users** to assign each of the users for whom you registered connections (see *Registering Application Connections in SAP Control Center*). Use commas to separate users.
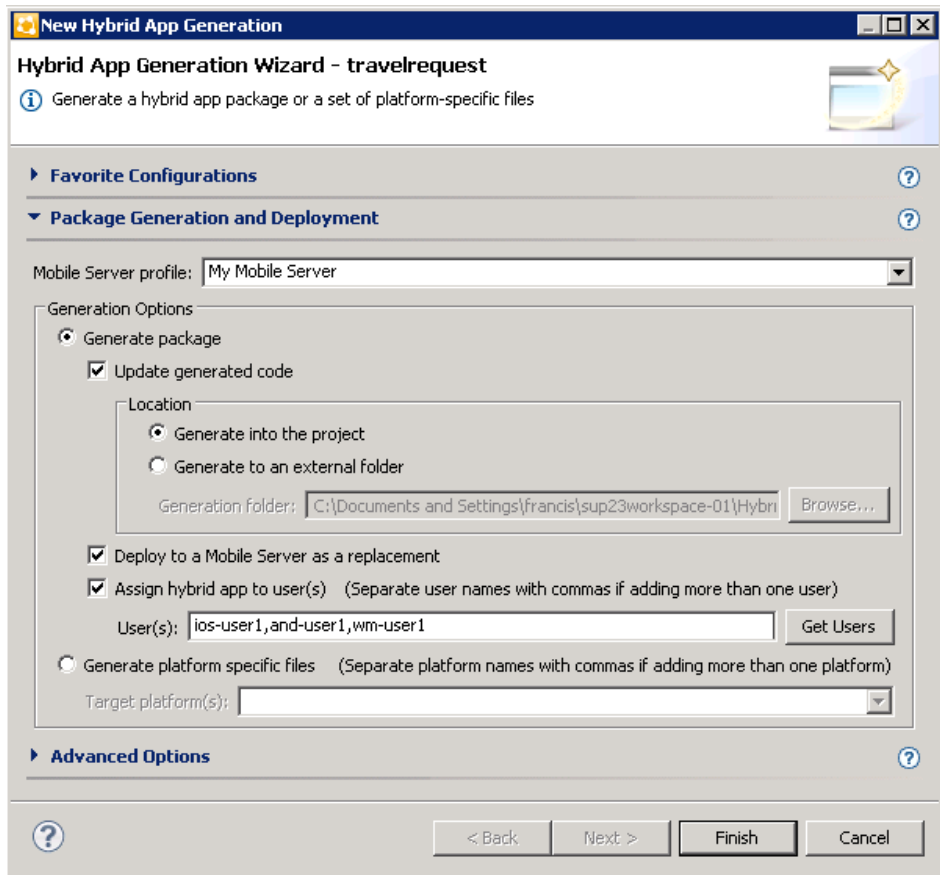
4. Click **Finish**.

   The WorkSpace assigns the TravelRequest application to the users and generates the files for the Hybrid App package in a zip archive on the server. The files are visible in your project in the Workspace Navigator.
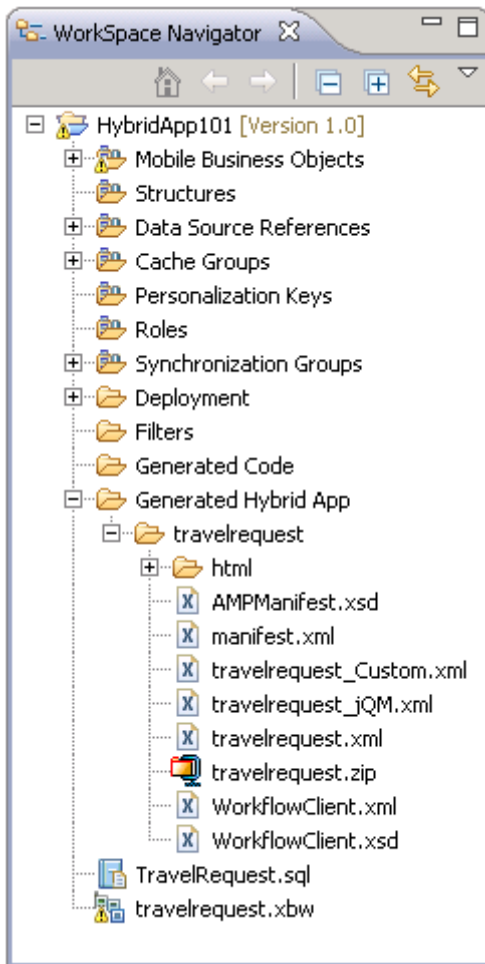
The Hybrid App package is added to Enterprise Explorer under the `Mobile Servers\My Mobile Server\Hybrid Apps` folder.

If these changes do not appear, press **F5** to refresh the display.

See also
*   *Developing a Hybrid App Screenflow* on page 17
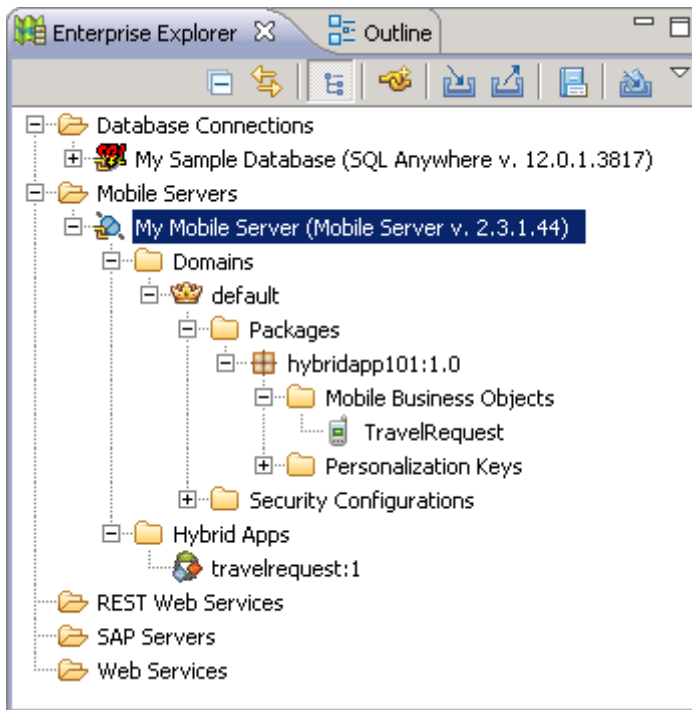*   *Testing the Hybrid App on the iPhone Simulator* on page 32

Customizing Generated Code

Modify Custom.js to change a generated Hybrid App.

Until now, this tutorial has shown how to customize your Hybrid App in the Designer, by modifying the properties of screens, menus, and controls. Your generated code reflects those changes. This topic demonstrates how to customize the generated code directly, by editing a special file, Custom.js.

Custom.js is generated the first time you generate the Hybrid App package. It contains commonly used functions from the SAP Mobile Platform API and instructions for modifying them. Any modifications you make in Custom.js override the default behavior of those functions. However, unlike other generated files, Custom.js is not overwritten the next time you regenerate your package. In this way, your custom code is preserved.

The customization in this tutorial adds your own image, such as a company logo, to the top of the application screen.

1. To add the header image:

   a) Copy the image that you want to use in your header to `Generated Hybrid App \travelrequest\html\images` in your project folder. Name the copy "header.png".
   For example:

   b) In Workspace Navigator, open the file, `Generated Hybrid App \travelrequest\html\js\Custom.js` for editing.

   c) In `Custom.js`, find the `customAfterHybridAppLoad` method. At the bottom of the function, add this code to display the header image:

```
hwc.customAfterHybridAppLoad = function() {
.
.
.
  hwc.log("customAfterHybridAppLoad", "DEBUG", false );
  hwc.log("Application " + hwc.getVersionURLParam(), "DEBUG",
false );
  hwc.log("Platform: isIOS=" + hwc.isIOS() + ", isAndroid=" +
hwc.isAndroid() + ",
  isBB=" + hwc.isBlackBerry() + ", isWin=" + hwc.isWindows() +
",
   isWinMobile=" + hwc.isWindowsMobile(), "DEBUG", false );

  var screenKey = getCurrentScreen();
  var form = document.forms[screenKey + "Form"];
  if (form) {
    var topOfFormElem = document.getElementById("topOf" +
screenKey + "Form");
    topOfFormElem.innerHTML = '<img alt="header" src="images/
header.png">';
  }
}
```

   Note: For the `src` value, use the name of the image file that you saved in Step a.

   d) Save your changes to `Custom.js`.

2. Run the Hybrid App Generation wizard again.

Testing the Hybrid App on the iPhone Simulator

Build a sample Hybrid Web Container in the Xcode IDE, and run the Travel Request application on an iPhone simulator.

1. *Building the iOS Hybrid Web Container Using the Provided Source Code*
   Build a sample Hybrid Web Container.
2. *Configuring iPhone Connection Settings*
   Configure settings for the Hybrid App.
3. *Running the Hybrid App on the iPhone Simulator*
   Run the Travel Request application on the iPhone simulator.

See also
- *Generating Code for a Hybrid App Package* on page 28
- *Testing the Hybrid App on the Android Emulator* on page 36

Building the iOS Hybrid Web Container Using the Provided Source Code

Build a sample Hybrid Web Container.

Prerequisites

- Register the device in SAP Control Center.
- Have access to a Mac with a supported version of Xcode and the iOS SDK.

See *Supported Hardware and Software* for the most current version information for mobile device platforms and third-party development environments.

Task

1. On your Mac, connect to the Microsoft Windows machine where SAP Mobile Platform is installed:
   a) In the Apple menu, click **Go > Connect to Server**.
   b) Enter the name or IP address of the machine.
      For example, smb://*machine DNS name* or smb://*IP Address*.
2. Copy the iOS_HWC_*version*.tar.gz archive from *SMP_HOME* \MobileSDK*version*\HybridApp\Containers\iOS\ to a location on your Mac.

   In the archive file name, *version* is the current SAP Mobile Server version number. For example, iOS_HWC_2.3.0.tar.gz.
3. Unpack iOS_HWC_*version*.tar.gz.
   The extraction creates a HybridWebContainer directory.
4. In the HybridWebContainer directory, double-click **HWC.xcodeproj** to open it in the Xcode IDE.
5. If you are building for a device, you must add provisioning profiles to the project to be able to sign the executable.

---

a) In Xcode, click the **HWC** project and select the HWC target.
b) Select the **Build Settings** tab.
c) Under the Code Signing section, add code-signing identities for each configuration (Debug, Release, or Distribution) you want to build, depending on how you will deploy the app.

   When you build the Hybrid Web Container using your provisioning profile, you are creating your own version of the application. You can reuse the bundle ID that is distributed with the HWC template project, but you cannot upgrade your custom-built application through the normal means.

   The reason for this is because on iOS the Keychain is used to store information and Keychain rights depend on the provisioning profile used to sign your application. Therefore, you should consistently use the same provisioning profile across different versions of your application. Follow the instructions in *Using Multiple Hybrid Web Containers on the Same iOS Device* when you build the HWC template source.

6. In Xcode, click **Product > Build** to build the project.

Configuring iPhone Connection Settings

Configure settings for the Hybrid App.

1. In the iPhone simulator Settings window, click **HWC**.
2. Enter these connection settings:
   - **Server name** – machine and domain of the host server where the mobile application project is deployed, or the input server IP address.
   - **Server port** – accept the default port number, 5001.
   - **Farm ID** – accept the default value, 0.
   - **Protocol** – accept the default protocol, HTTP.
   - **Registration** – accept the default type, Manual.
   - **User name** – user that you registered for this device in SAP Control Center.
   - **Activation code** – the user activation code you specified in SAP Control Center.
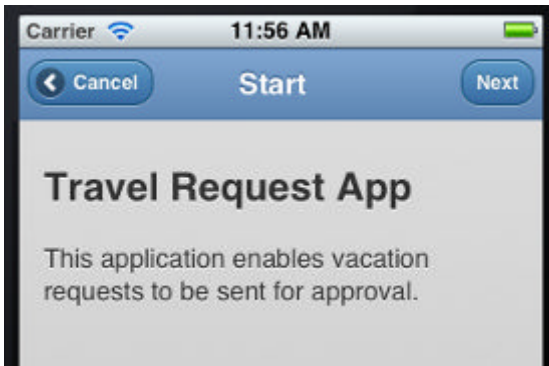
Running the Hybrid App on the iPhone Simulator

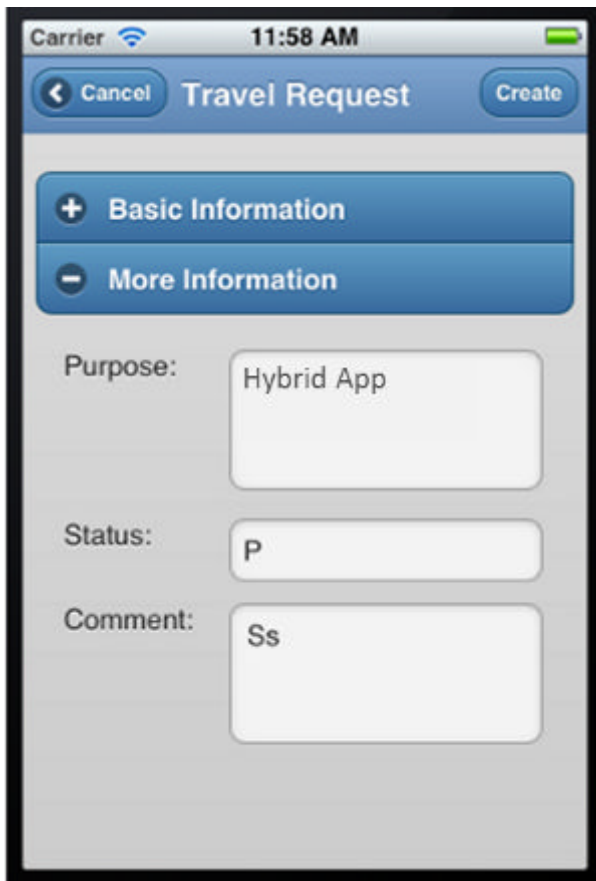Run the Travel Request application on the iPhone simulator.

1. Click **HWC**.
2. In Hybrid Apps, click **Travel Request**.

3. In the introduction page, click **Open Travel Request**.

4. Fill out the information for the travel request, and click **Create**.

The simulator closes the Hybrid App. The TravelRequest database table is updated on the back end.

Testing the Hybrid App on the Android Emulator

Install and configure the Android simulator software and use it to run the Travel Request application.

1. *Configuring the Android Emulator*

   Configure an Android emulator for testing a Hybrid App package.
2. *Configuring Connection Settings on the Android Emulator*

   Configure connection settings on the Android emulator.
3. *Running the Hybrid App on the Android Emulator*

   Run the Travel Request application on the Android emulator.

See also
*   *Testing the Hybrid App on the iPhone Simulator* on page 32
*   *Testing the Hybrid App on the BlackBerry Simulator* on page 45

Configuring the Android Emulator

Configure an Android emulator for testing a Hybrid App package.

Note: The steps or interface may be different depending on the Android SDK version you are using.

1. Install the Android SDK.
    a) Go to *http://developer.android.com/sdk/*.
    b) Download the Android SDK (for example, `installer_r21-windows.exe`).

        Note: Do not download the larger SDK starter package (ADT Bundle for Windows). The starter package includes not only the SDK but also the ADT plug-in for Eclipse and a more recent platform than the one shown in this tutorial.

    c) In Windows Explorer, double-click the downloaded installer to run it.

        Note where the SDK is installed on your system, for example,

        `C:\Program Files\Android\android-sdk`.

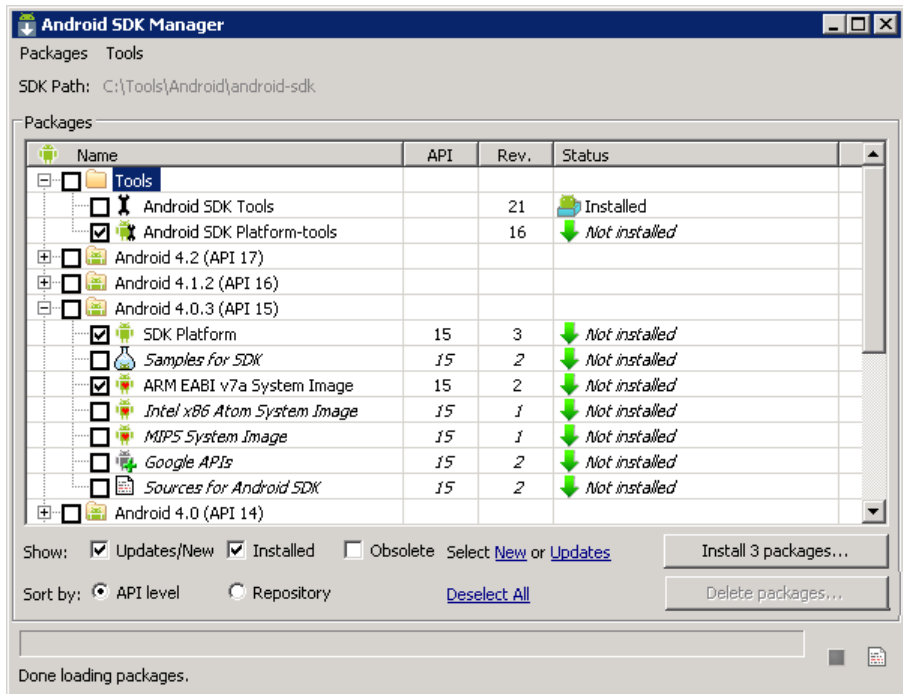2. Install the SDK platform tools:
    a) Run the Android SDK Manager, *android-sdk*`\SDK Manager.exe`.
    b) In the Android SDK Manager, expand Tools and select **Android SDK Platform-tools**.

        Android SDK Tools should already be installed.

    c) Expand **Android 4.0.3 (API 15)** and select these packages:
        *   **SDK Platform**.
        *   **ARM EABI v7a System Image**.
    d) Click the **Install *n* packages** button.

e) In Choose Packages to Install, select **Accept All**, then click **Install**. Close the log window when done.

f) Close the Android SDK Manager.

3. Run the Android Virtual Device Manager, *android-sdk*\AVD Manager.exe.

4. Configure and start an Android emulator instance.

a) In the AVD Manager, click **New**.

b) In the Create new Android Virtual Device window, enter an AVD name and select a supported Android device for this instance.

For example:

Create new Android Virtual Device (AVD)   ☒

AVD Name:	Android_4.0.3
Device:	Nexus S (4.0", 480 × 800: hdpi) ▼
Target:	Android 4.0.3 - API Level 15 ▼
CPU/ABI:	ARM (armeabi-v7a) ▼
Keyboard:	☑ Hardware keyboard present
Skin:	☑ Display a skin with hardware controls
Front Camera:	None ▼
Back Camera:	None ▼

Memory Options:
RAM: 343   VM Heap: 32

Internal Storage:
200   MiB ▼

SD Card:
◉ Size: _____ MiB ▼
○ File: _____ Browse...

Emulation Options: ☐ Snapshot   ☐ Use Host GPU

☐ Override the existing AVD with the same name

[OK]   [Cancel]

c) Click **OK** to add the instance to the AVD Manager.
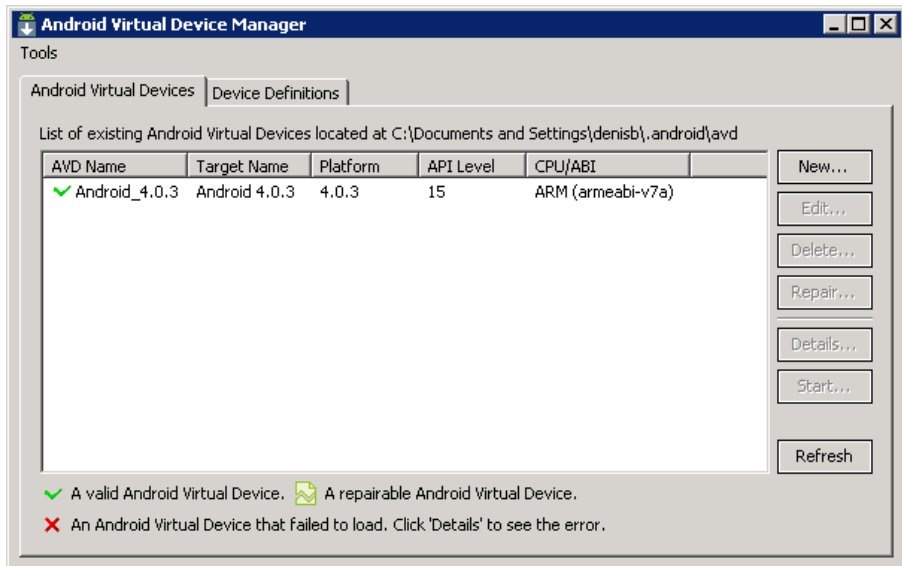
d) Select the new virtual device and click **Start**.

e) In Launch Options, click **Launch** to open the Android emulator screen.

5. Install the Hybrid Web Container on the emulator instance:

   a) With the Android emulator running, open a command prompt window.

   b) Run *android-sdk*\platform-tools install *SMP_HOME*
   \MobileSDK*version*\HybridApp\Containers\Android
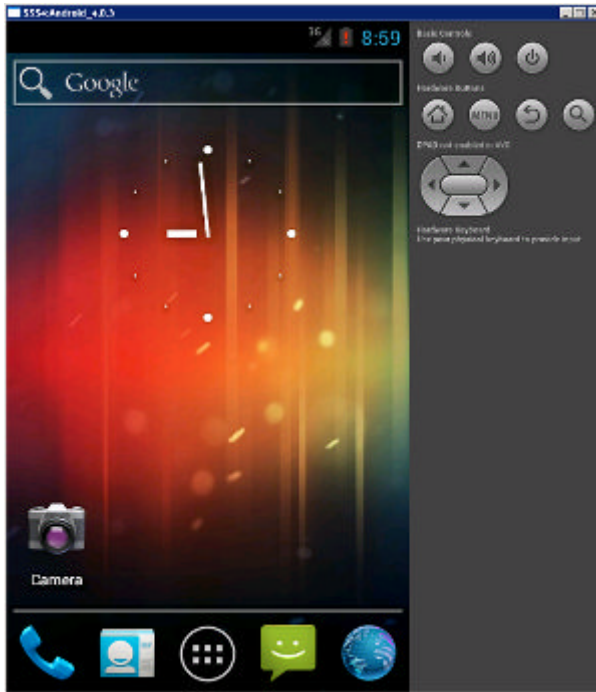   \HybridWebContainer.apk.

   For example:

```
C:\Android\android-sdk\platform-tools\adb install ^
C:\SAP\MobilePlatform\MobileSDKversion\HybridApp\Containers
\Android\HybridWebContainer.apk
```
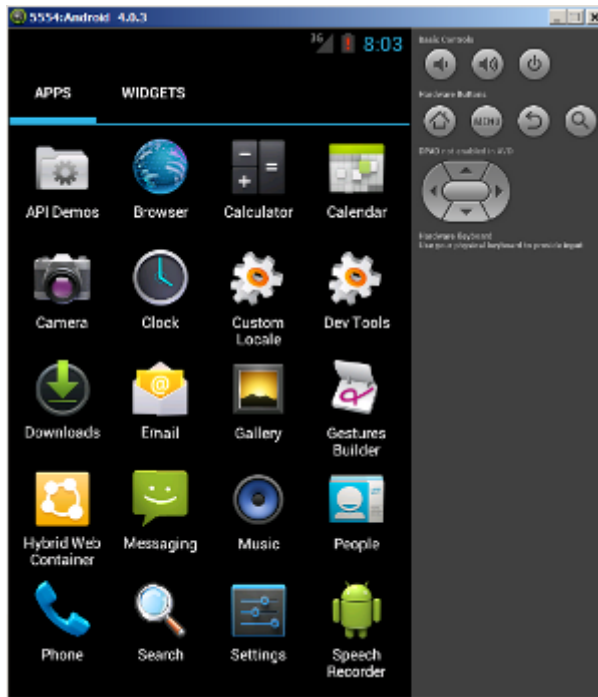
Configuring Connection Settings on the Android Emulator

Configure connection settings on the Android emulator.

1. In the emulator home page, click the launcher icon.

2. In the launcher, find and click the **Hybrid Web Container** icon.

3. Enter a new password and click **OK**.

4. In Settings, enter the connection settings for your application.

- **Registration** – accept the default type, Manual.
- **Server name** – IP address of the host server where the mobile application project is deployed.
- **Port** – accept the default port number, 5001.
- **Farm ID** – enter 0.
- **User name** – user that you registered for this device in SAP Control Center.
- **Activation code** – the user activation code you specified in SAP Control Center.
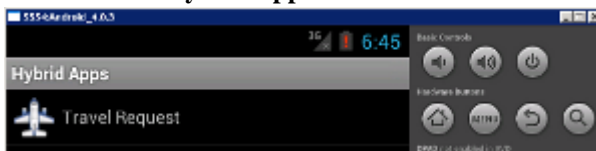- **Protocol** – accept the default protocol, HTTP.
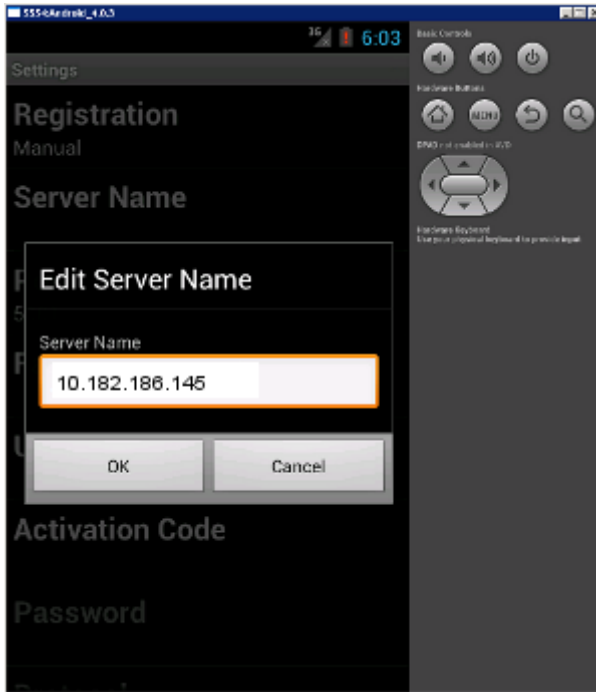
For example:

5. Click **Menu**. In the menu, click **Save** to save your settings.

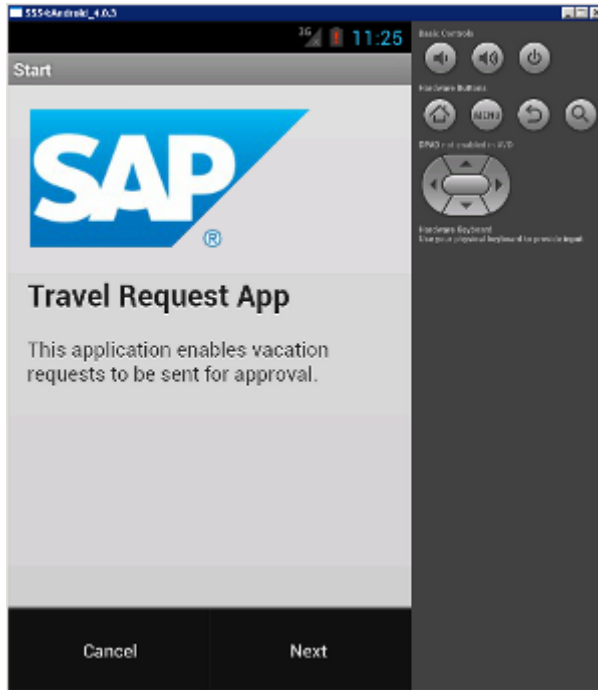## Running the Hybrid App on the Android Emulator

Run the Travel Request application on the Android emulator.

1. Open the Hybrid Web Container screen.
   For example, from the Settings screen, click the **Back** button.
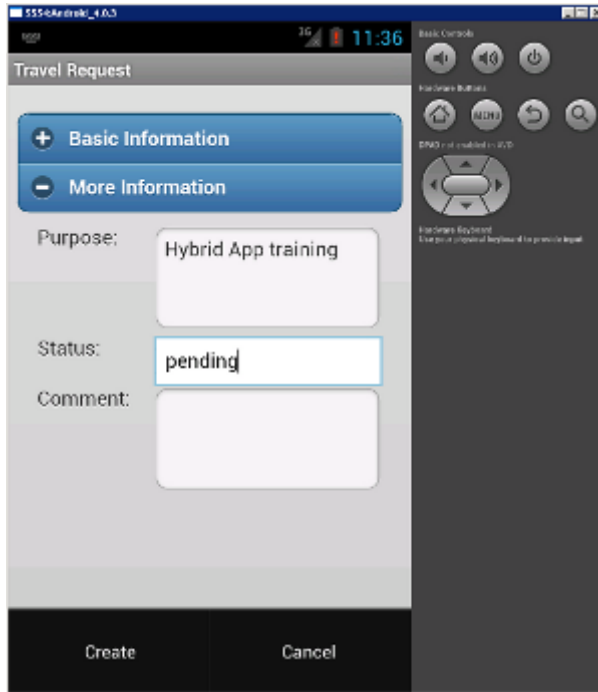
2. Click **Menu > Hybrid Apps**.

3. Click **Travel Request** to open the Hybrid App start screen.

---

4. Click **Menu > Next** to open the Travel Request screen.

5. Enter your travel request information.
6. Click **Menu > Create**.

The Hybrid App closes and the `TravelRequest` database table on the server is updated.

Testing the Hybrid App on the BlackBerry Simulator

Install and configure the BlackBerry simulator, and use it to run the Travel Request application.

1. *Downloading the BlackBerry JDE*

   To generate and distribute BlackBerry device applications, download the BlackBerry JDE and its prerequisites from the BlackBerry Web site.

2. *Configuring the Hybrid App in the BlackBerry Simulator*

   Start the BlackBerry simulator and configure it to connect to the Hybrid App.

3. *Running the Hybrid App on the BlackBerry Simulator*

   Run the Travel Request application on the BlackBerry simulator.

See also

- *Testing the Hybrid App on the Android Emulator* on page 36
- *Testing the Hybrid App on the Windows Mobile Emulator* on page 51

Downloading the BlackBerry JDE

To generate and distribute BlackBerry device applications, download the BlackBerry JDE and its prerequisites from the BlackBerry Web site.

Prerequisites

- The BlackBerry MDS software requires the 32-bit JDK to be installed, even for 64-bit operating systems.
- A registered BlackBerry developer account to download the JDE.

Task

Go to the BlackBerry Web site to download and install the BlackBerry JDE.
The MDS-CS simulator is installed with the BlackBerry JDE.

Configuring the Hybrid App in the BlackBerry Simulator

Start the BlackBerry simulator and configure it to connect to the Hybrid App.

1. Under your BlackBerry device installation folder, run **clean.bat** to remove any previously installed Hybrid Web Container.
   For example, in Windows Explorer, open `C:\Program Files\Research In Motion\BlackBerry JDE 6.0.0\simulator` and double-click `clean.bat`.

2. Start the Mobile Data System Connection Service (MDS-CS): click **Start > Programs > Research In Motion > BlackBerry JDE 6.0.0 > MDS-CS**.
   MCS-CS runs in a command prompt window and waits for a BlackBerry device to send data.

3. In your Windows file system, copy the SAP Mobile Platform Hybrid App client files to your BlackBerry simulator.
   For example, copy `C:\SAP\MobilePlatform\MobileSDKversion\HybridApp\Containers\BB\*.*` to `C:\Program Files\Research In Motion\BlackBerry JDE 6.0.0\simulator`.

4. Start the BlackBerry simulator.
   For example, click **Start > Programs > Research In Motion > BlackBerry JDE 6.0.0 > Device Simulator**.

5. Click through any introductory screens:

   - If the end-user license agreement opens, scroll to the end and click **OK** to accept it.
   - In the Setup screen, click **Escape** and then **Exit**.

   The main window opens:

6. Restart the simulator.

7. In the BlackBerry simulator, configure connection settings for the Hybrid App:

   a) In the main screen, click **All** to access the applications screen.

b) Click the **Hybrid Web Container** icon.

c) In the Messages screen, click **Menu > Settings**.

d) In the Connection screen, enter the connection settings.

- **Registration** – accept the default type, `Manual`.
- **Server name** – machine and domain of the host server where the mobile application project is deployed, or the input server IP address.
- **Server port** – accept the default port number, `5001`.
- **Farm ID** – accept the default value, `0`.
- **User name** – user that you registered for this device in SAP Control Center.
- **Activation code** – the user activation code you specified in SAP Control Center.
- **Protocol** – accept the default protocol, `HTTP`.

For example:

e) Click **Menu > Save**. Click **OK** if prompted to confirm.

## Running the Hybrid App on the BlackBerry Simulator

Run the Travel Request application on the BlackBerry simulator.

Note: This tutorial was developed using a supported version of the BlackBerry simulator. If you use a different supported version, the interface might differ in some details.

1. Start your BlackBerry simulator instance if it is not already running:
2. In the emulator, click **All**.
3. Click the **Hybrid Web Container** icon.
4. In the Messages screen, click **Menu > Hybrid Apps**.
5. In the Hybrid Apps screen, click **Travel Request**.
6. In the Start screen, click **Menu > Next**.
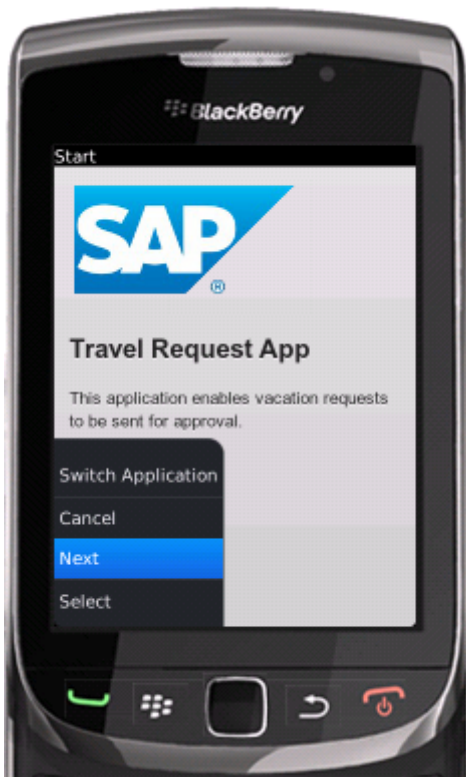
7. In the Travel Request screen, open each accordion control and enter your travel request information.

8. Click **Menu > Create** to submit your changes to the server.
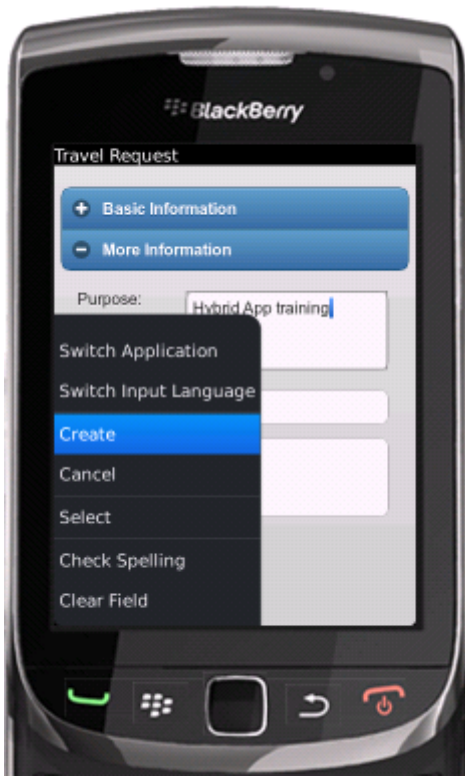
The Hybrid App closes and the `TravelRequest` database table on the server is updated.

Testing the Hybrid App on the Windows Mobile Emulator

Install and configure the Windows Mobile synchronization software and emulator, and use it to run the Travel Request application.

1. *Installing Microsoft Synchronization Software*

   Install and configure Microsoft synchronization software so you can deploy and run an application on a Windows Mobile emulator.

2. *Configuring the Windows Mobile Emulator*

   Configure the Windows Mobile emulator for Hybrid App package deployment.

3. *Configuring Connection Settings on Windows Mobile*

   Configure the connection settings on the Windows Mobile emulator.

4. *Running the Hybrid App on the Windows Mobile Emulator*

   Run the Travel Request application on the Windows Mobile emulator.

See also
*   *Testing the Hybrid App on the BlackBerry Simulator* on page 45
*   *Verifying the Data on the Back-end Database* on page 60

Installing Microsoft Synchronization Software

Install and configure Microsoft synchronization software so you can deploy and run an application on a Windows Mobile emulator.

Note: These instructions describe how to install Microsoft ActiveSync for Windows XP. If you are using Windows Vista, Windows 7, or Windows 2008, install Virtual PC 2007 SP1 and Windows Mobile Device Center to manage synchronization settings. Download the Windows Mobile Device Center from *http://www.microsoft.com/en-us/download/details.aspx?id=15* and follow Microsoft instructions for installing and using that software instead of this procedure.

1. Download Microsoft ActiveSync:
    a) In a Web browser, open the Windows Phone page at *http://www.microsoft.com/windowsmobile/en-us/help/synchronize/device-synch.mspx*.
    b) Follow the instructions to select and download the sync software for the system's operating system. Windows XP requires ActiveSync version 4.5.
    c) In the Windows Phone downloads page, click the **ActiveSync** button.
    d) Download the ActiveSync installation file and save it to your local system.
2. Run the downloaded installation file.
    For example, double-click **setup.msi** in Windows Explorer.
3. When the installation is complete, restart the system.
4. Start ActiveSync if it does not start automatically.
    For example, click **Start > Programs > Microsoft ActiveSync**.
5. Click **File > Connection Settings**.
6. Select **Allow connections to one of the following**, then select **DMA**.
7. Select **Work Network** for "This computer is connected to".
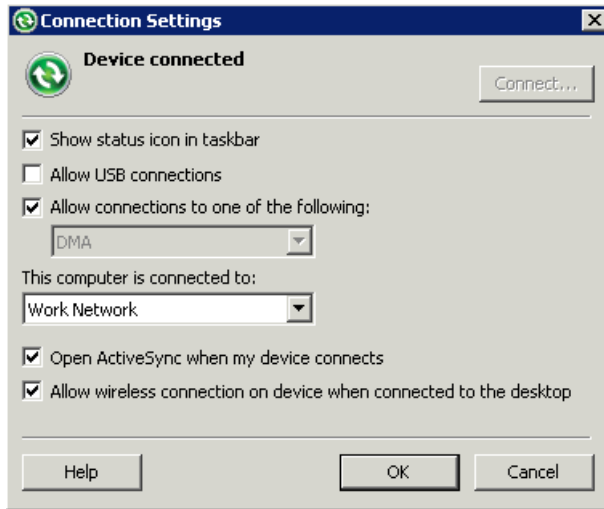
8. Click **OK**.

Configuring the Windows Mobile Emulator

Configure the Windows Mobile emulator for Hybrid App package deployment.

Prerequisites

- Install Windows Mobile Professional emulator images to test and run the sample application. Emulator images are available individually, and some are installed with Visual Studio and Windows Mobile SDK. For example, the emulator used for this tutorial is included in the Windows Mobile 6.5.3 Developer Tool Kit, which you can download from the Microsoft Download Center (*http://www.microsoft.com/download*).
- Install the Microsoft Device Emulator, if it is not already installed (for example, as part of Visual Studio or the Windows Mobile SDK). Go to the Microsoft Download Center, search for "Device Emulator", and follow the links to download and install the Standalone Release of the Microsoft Device Emulator.
- Download the Microsoft .NET Compact Framework Redistributable (`NETCFSetupv35.msi`) from the Microsoft Download Center to your system.

  Note: Do not install the Redistributable on your system immediately; install it on the emulator later in this task.

Task

This tutorial uses examples based on one of the supported emulators. The screens and steps for configuring other supported emulators might be different. See *Supported Hardware and Software* for the most current version information for mobile device platforms and third-party development environments.

1. Start the synchronization software if it is not already running.
   For example, on Windows XP, start Microsoft ActiveSync. On Windows Vista, Windows 7, or Windows 2008, start the Windows Mobile Device Center.

2. Start the Device Emulator Manager and select an emulator to run.
   For example:

   a. Double-click **C:\Program Files\Microsoft Device Emulator \1.0\dvcemumanager.exe**.

   b. In the Device Emulator Manager, right-click the device you want to run and choose **Connect** to open the emulator.
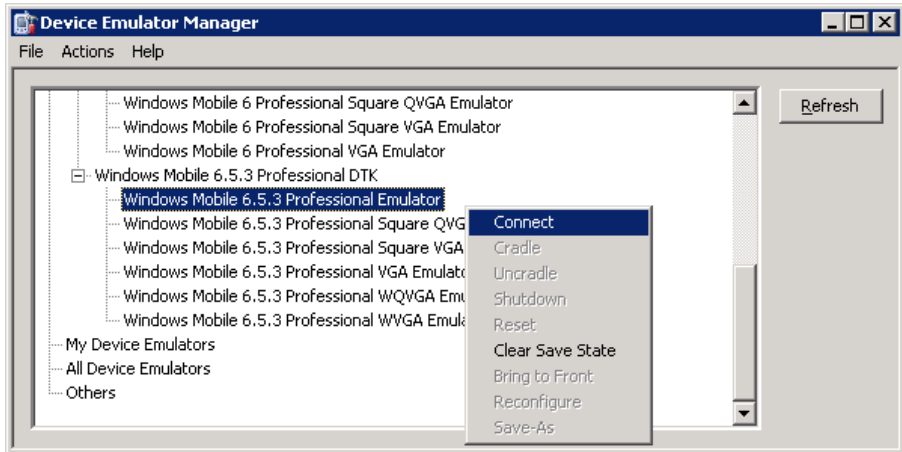
   c. In the Device Emulator Manager, right-click the device again and click **Cradle**.

3. The synchronization software runs and connects to your device. If the Pocket PC Sync Setup wizard opens, follow the instructions to configure synchronization. Accept all the defaults, or optionally, change the default computer name and the information types to synchronize. Click **Finish** to save your settings.
   ActiveSync completes the connection:
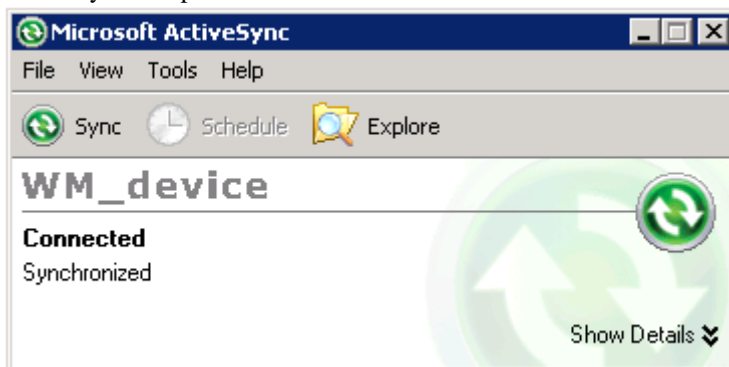
4. Run `NETCFSetupv35.msi` to install the Microsoft .NET Compact Framework Redistributable on your running emulator. Follow the setup wizard instructions, and click **Finish** to close the wizard when you are done.

> **Note:** Be sure to run `NETCFSetupv35.msi` while your emulator is running; otherwise the .NET Compact Framework Redistributable is not installed correctly.

5. In Windows Explorer, copy *SMP_Home*\MobileSDK*version*\HybridApp \Containers\WM\HybridWebContainer.cab file to a folder in the Mobile Device file system. If prompted by the File Conversion dialog to confirm the conversion of synchronized files, click **OK**.
For example:
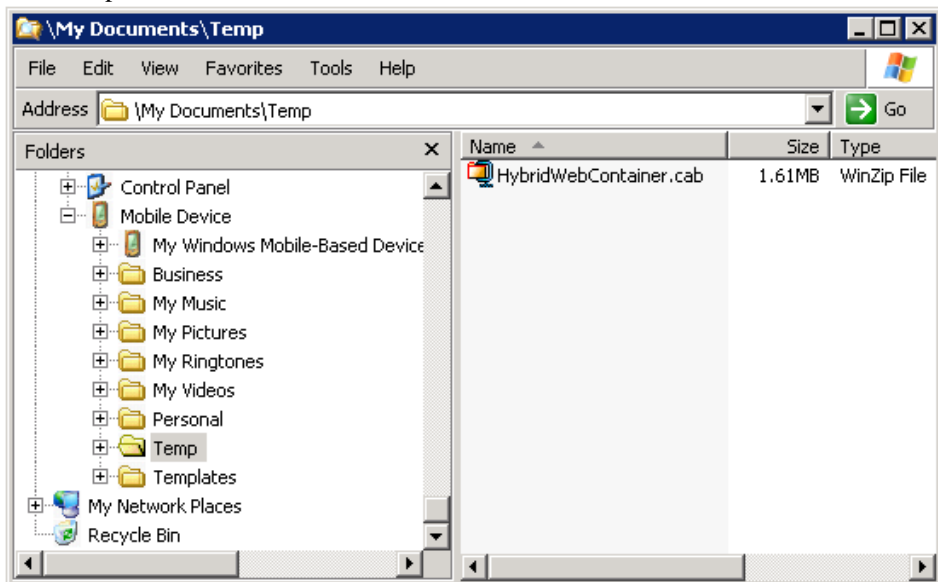
6. On the device emulator, open File Explorer and browse to the folder to which you copied the CAB file. Click the file once to install the Hybrid Web Container on your emulator.

Configuring Connection Settings on Windows Mobile

Configure the connection settings on the Windows Mobile emulator.

1. In the emulator, open the Start screen and click **Hybrid Web Container**.

2. In the Messages screen, click **Settings**.

3. In the Connection screen, enter the connection settings.

- **Server name** – machine and domain of the host server where the mobile application project is deployed, or the input server IP address.
- **Server port** – accept the default port number, 5001.
- **Farm ID** – accept the default value, 0.
- **User name** – user that you registered for this device in SAP Control Center.
- **Registration** – accept the default type, Manual.
- **Activation code** – the user activation code you specified in SAP Control Center.
- **Protocol** – accept the default protocol, HTTP.

For example:

4. Click **Save**.

5. Click **OK**.

> **Tip:** To view the device log at any time, return to the Settings screen, click **Menu > Show log**.

Running the Hybrid App on the Windows Mobile Emulator

Run the Travel Request application on the Windows Mobile emulator.

1. In the emulator, open the Start screen and click **Hybrid Web Container**.

2. In the Messages screen, click **Hybrid Apps**.

3. In Hybrid Apps, click **Travel Request**:

4. In the Start screen, click **Next**.

5. Enter your travel request information.
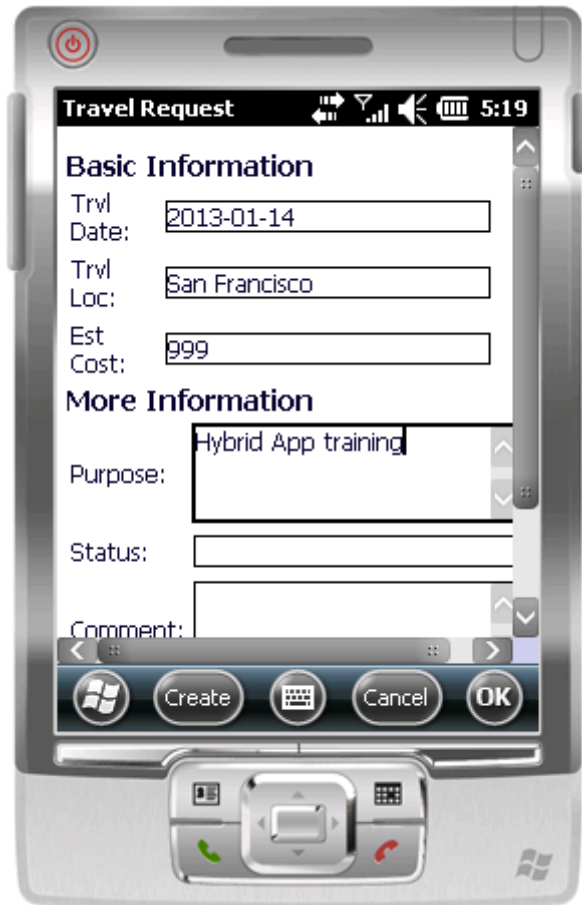
6. Click **Create**.

The emulator closes the Hybrid App and updates the TravelRequest database table on the back end.

Verifying the Data on the Back-end Database

After submitting a travel request in an emulator, verify that the information is updated in the database.

1. In SAP Mobile WorkSpace, expand the sample database in Enterprise Explorer.

2. In the Tables folder, right-click **TravelRequest** and select **Edit Data**.
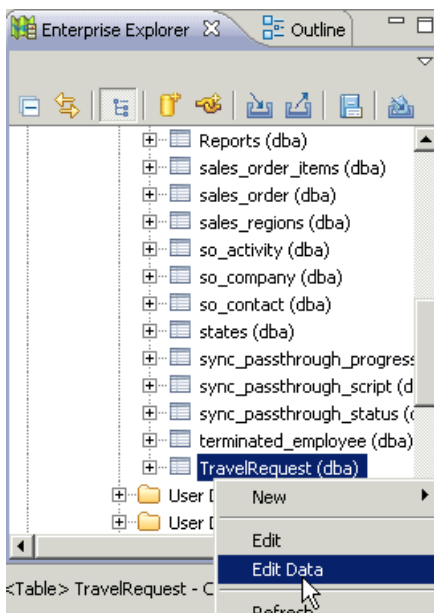
3. In the Table Data Filter dialog, accept the default settings and click **OK**.
The TravelRequest table opens, displaying the new row.

See also

*   *Testing the Hybrid App on the Windows Mobile Emulator* on page 51

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*

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