

Extension Services



Order Entry Sample

Contents

Samples	1	Running in WebSphere Studio	3
Order Entry	1	Running on a device.	4
Creating the project	2	Using the application	6
Launching the server application	3		

Samples

Order Entry

Extension Services provides a sample three-tier application that demonstrates some of the components and capabilities of Service Management Framework (SMF) and Extension Services. This sample, called Order Entry, uses the Web Container, Servlet 2.3, Configuration Admin services from SMF, and MQ Everyplace[®], JDBC, and DB2[®] Everyplace provided by Extension Services.

This sample program illustrates the following scenario. A delivery employee drives a soda truck to three different customers: SpeedyMarket, Marine Mart, and Quick Stop. As he travels to each of these locations to offload his stock, he determines how many bottles and cans of soda to order for the next week and enters that information into his device. The order is stored within the local DB2 Everyplace database, and also queued for transmission to the server using MQ Everyplace. The order will be transmitted to the server only when a network connection exists and the server is ready to receive, otherwise it will be held on the device. Once the order is sent from the device to the backend server, it is then confirmed with a message back to the employee's device.

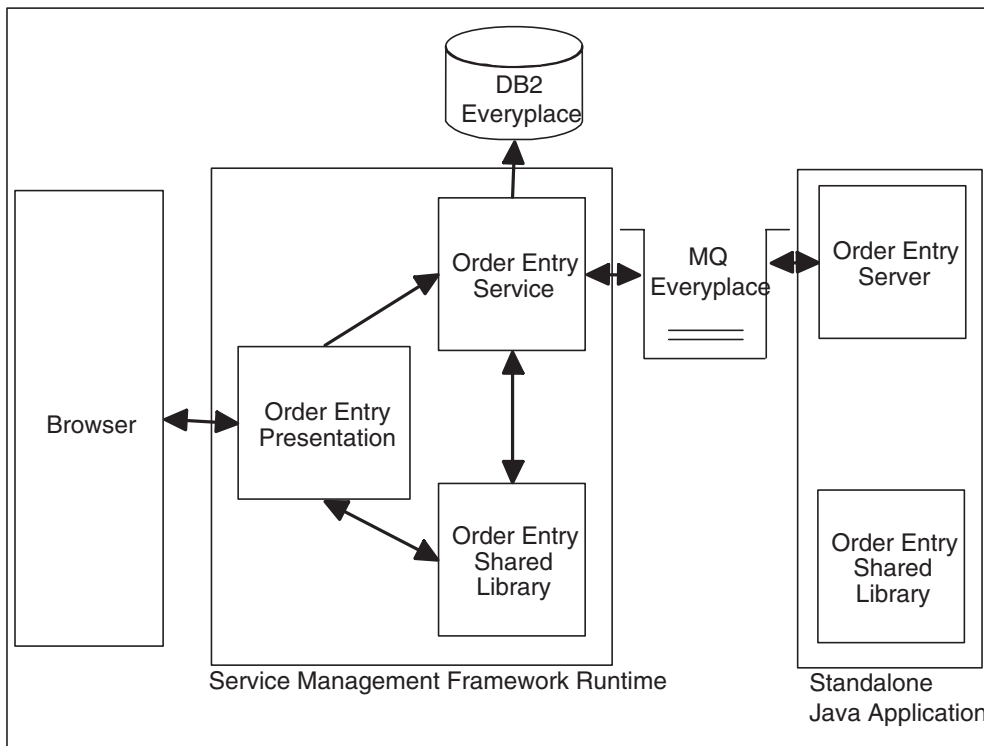


Figure 1. Order Entry sample diagram

This sample contains two projects with four components. The first project, Order Entry Service Sample, contains a core library bundle, a service bundle, and the server-side application. The second project, Order Entry Web Application, contains the presentation Web application.

In this scenario, the Order Entry client processes input from the user and generates requests to the Order Entry server, which represents an inventory management system. A DB2 Everyplace database stores the requests and sends the requests through MQ Everyplace to the Order Entry server. When the Order Entry server receives a message, it confirms the order and sends a response to the client. When the Order Entry service receives the response message, it updates the status of the order in the client database.

Note: If the Order Entry server is not running when an order is submitted, MQ Everyplace retains the order in the queues until the server becomes available.

Creating the project

Perform the following steps to create the Order Entry Service and Order Entry Web Application projects:

1. Select **File > New > Example... > Extension Services > Order Entry Web Application**. Click **Next**.

2. Click **Finish**.

Note: If prompted to switch to the SMF perspective, click **OK**.

Launching the server application

The Order Entry server is a standalone Java application. Complete the following steps to set up a launch configuration for the program that starts the Order Entry server:

1. In the SMF perspective, select **Run > Run...** to display the Launch Configurations window.
2. Select **Java Application**. Click **New**. Provide a name for the application.
3. If the Project field is empty, select **browse** and choose **Order Entry Service Project**.
4. Click **Search** next to the Main class field.
5. In the Choose Main Type window, select **Server** and click **OK**.
6. Select the **JRE** tab.
7. Select an applicable J9 JVM to run this application. Choose either **J9 2.2.0** or **Extension Services Generated JRE (JCL Foundation)**.
8. Click **Run**.

Notes:

1. A J9 console window might display after you click Finish. If a J9 console window is displayed, minimize the window and return to WebSphere Studio. If you close the window, the server terminates and the sample application will not function correctly.
2. If an Error panel displays with the message "Library JCL Foundation 1.0 [j2me/2.2.0/foundation10] is not available on VM vmname", then a J9 JVM is not being used to run the application. Return to the Launch Configuration window for this application and select a J9 JVM.

Running in WebSphere Studio

Setting up

The Order Entry sample requires the use of native libraries which WebSphere Studio must be able to locate. If you have not already completed the steps in the *Getting Started* section of this guide, perform the following steps to set up the environment for the Order Entry sample:

1. For Windows, add the following entry to the end of your PATH environment variable:

```
WSDD_HOME\wsdd5.0\technologies\smf\client\nativeLib
```

For Linux, add the following entry to the end of your LD_LIBRARY_PATH environment variable:

```
WSDD_HOME/wsdd5.0/technologies/smf/client/nativeLib
```

Where WSDD_HOME is the location where you installed WebSphere Studio Device Developer 5.7.

2. Restart WebSphere Studio.

Deploying the application

Perform the following steps to set up the Extension Services Local Server:

1. Select **File > New > Other**. The New window is displayed.

2. Select **Server** in the left pane and **Server** and **Server Configuration** in the right pane. Click **Next**. The Create a New Server and Server Configuration window is displayed.
3. Specify a name of OrderEntry in the Server field. In the Server type field, select **Extension Services > Local Server**. Accept the default values for the other fields. Click **Next**.

Note: If you have not already created a Servers project, you might be prompted to create one. Answer **Yes** to create a new Servers project.

4. Accept all default values and click **Next**.
5. Select **Extension Services: jclFoundation** as the Platform Profile.
6. Click **Finish**.
7. In the Server perspective, open the Servers window and verify that the status of your server is Stopped.
8. Right-click your server in the Server window and select **Start**.

Perform the following steps to submit the Order Entry Service bundles to the Extension Services Local Server:

1. In the SMF perspective, right click on the **Order Entry Service** project and select **SMF > Submit Bundle...**
2. In the resulting Target Submission window, select **Submit Jar**.
3. Select **Admin@localhost:8080/smf** in the Export Targets box.
4. Click **Finish** to submit the bundle to the bundle server.
5. Switch to the Bundle Server view and open **Admin@localhost:8080/smf > Bundles**. Verify that bundles named OrderEntryCommon and OrderEntryService are listed.

Launching the client application

Complete the following instructions to launch the Order Entry Web Application sample:

1. Change to the Server perspective.
2. In the Server Configuration window, right-click on the server that you created in and select **Add and remove projects...**
3. Select **Order Entry Web Application** from the Available projects and click **Add**. Click **Finish**.
4. In the Servers window, right-click on **OrderEntry** and select **Restart** to start the bundle server and the SMF runtime and to load the Order Entry Web Application into the SMF Runtime.
5. In the Navigator window, right-click **Order Entry Web Application > Run on Server** to launch a browser window that loads Order Entry Web Application.

For instructions on how to access and use the Order Entry sample application, see “Using the application” on page 6.

Running on a device

Deploying the application

Perform the following steps to submit the Order Entry sample application bundles to the filesystem:

1. In the SMF perspective, right click on the **Order Entry Service** project and select **SMF > Submit Bundle...**

2. In the resulting Target Submission window, select **Submit Jar** and click **Add Directory...**
3. Select a folder in which to submit the Order Entry sample application service bundles.
4. Select the newly added directory in the Export Targets box.
5. Click **Finish** to submit the bundle to the filesystem.
6. Right click on the **Order Entry Web Application** project and select **Export...**
7. Choose **WAB file** and click **Next**.
8. Provide the same directory location where you submitted the Order Entry sample application service bundles, and name the file `OrderEntry.wab`. Click **Finish**.
9. Verify that the bundles are placed into the specified directory. The bundles will be named `OrderEntryCommon+5_7_0.jar`, `OrderEntryService+5_7_0.jar`, and `OrderEntry.wab`.

Creating a Platform Builder project

Perform the following steps to create a Platform Builder project that contains the Order Entry application bundles. This project can be used to build a target platform for a specific device type that runs the Order Entry application.

1. Select **File > New > Project > Extension Services > Platform Builder**. Click **Next**.
2. Enter `OrderEntryPlatformBuilder` for the project name. Click **Next**.
3. On the Platform Options page, select the desired device type. Select the desired output format and specify an output location. If the output location is left blank, the build output will be placed in the `OrderEntryPlatformBuilder/output/` directory. Click **Next**.
4. On the J9 Options page, select the **jclFoundation** class library and the desired package format.
5. On the Startup Options page, leave all options with their default values. Click **Next**.
6. On the Platform Languages page, select the desired platform language support. Click **Next**.
7. On the Custom Bundles page, click **Add...** and select the `OrderEntryCommon+5_7_0.jar`, `OrderEntryService+5_7_0.jar`, and `OrderEntry.wab` files from the filesystem location where they were deployed. Click **OK**.
8. Click the **Compute** button to determine the dependencies for the Calendar sample application bundle. A Resolve Dependencies dialog will appear. Select the following bundles from the list and then click **Resolve...**
 - DB2 Everyplace Client
 - JDBC 2.0
 - Java Servlet 2.3 API
 - MQ Everyplace
 - OSGi Service Interfaces and Classes
 - OSGi Service Tracker
 - Web Application
 - Web Container

9. After recalculating the platform dependencies, another Resolve Dependencies dialog will appear. Click **Select All** and then click **Resolve...** A dialog will appear stating that all dependencies have been resolved. Click **OK**. Click **Next**.
10. On the System Bundles page, the system bundles that were added from the dependency resolution are listed. Click **Next**.
11. On the Summary page, make sure that the **Build platform** checkbox is selected. Review the summary and click **Finish**. The wizard will create the `OrderEntryPlatformBuilder` project in the workspace and will immediately launch a build to generate a target platform.
12. When the build is complete, verify that the target platform package is placed in the previously specified output directory.

Launching the client application

To run the target platform on your device, perform the following steps:

1. Install the target platform onto your device. This procedure may differ by device, depending on which output format was selected:
 - If the output format was `zip`, `tar` or `tar.gz`, unpackage the platform package into a location on the device filesystem
 - If the output format was `ipk`, copy the `ipk` file to the device and install with the appropriate `iPKG` tools
2. Execute the platform startup script. The startup script is named `StartSMF` and is located in the `smf` directory of the target platform. Running this script will launch the Extension Services platform.

Note: If running on a Linux device (Red Hat PC or Zaurus), be sure to run the `StartSMF` script as the root user, as the platform will be attempting to access port 80. See the Additional Information appendix in this guide for more information.

For instructions on how to access and use the Order Entry sample application, see “Using the application.”

Using the application

After you run the Order Entry application, you should be able to complete the following tasks:

- Accessing the current orders
- Displaying the customer status
- Placing an order

When you run the Order Entry application, assume the following:

- Web Container uses port 80
- The Order Entry Server runs on port 8082
- Extension Services clears the local database every time you start the Order Entry service

The Order Entry application provides a configuration page that you can use to adjust configuration parameters. Go to `http://localhost/OrderEntry/OrderEntryConfig` to access the configuration page. You can also update the same parameters with the SMF Administrator utility. For more information about the SMF Administrator utility, refer to the Service Management Framework documentation.

Use the following instructions to run the Order Entry sample application:

1. Verify that both the client and server applications are running.
2. If you are running the client on the same device on which you are running the server, skip to step 7.
3. If you are running the client and server on different devices, you will need to configure the client to communicate with the server. From a browser on the device, go to <http://localhost/OrderEntry/OrderEntryConfig>.
4. Enter the host name or IP address for the workstation running the server application.
5. Enter a new name for the client device.
6. Click **OK**.
7. Open a browser and go to <http://localhost/OrderEntry> to access the Order Entry sample application. The main splash screen is displayed.
8. On the main splash screen, click **Start** to begin the application.
9. The Select a Customer screen is displayed. Select a customer from the list and click **OK**.
10. The Current Orders screen lists the orders that have been placed and the order status for the customer you selected. If this is the first time you have run the demonstration, there are no orders to display.
11. Click **New Order** to place a new order.
12. The New Order screen is displayed. Select the number of cans and bottles to order from the lists and click **Submit**.
13. The Current Orders screen displays the order you created. The screen indicates the status of the order as Queued. Assuming the connection between the client and the server is functioning properly, the status changes to Confirmed when you refresh the page.

Note: You might have to refresh the screen more than once before the status changes to Confirmed.
14. To return to the Select a Customer screen, click **Customers**. From there, you can select other customers and place additional orders.

