Deploying Database Web Services

Purpose

This module describes how to deploy a call out Web service to OC4J.

Topics

This module will discuss the following:

Viewing Screenshots

Overview

Back to List

Database Web Services Overview

Web services enable programmatic access to remote content and application functionality on the Web using industry standard mechanisms, without any dependency on the platform, location, and implementation. The interfaces and binding of Web services are capable of being described by standard XML. Web services are services deployed on the middle-tier application servers. Oracle9 *i* AS Web services focus on exposing the J2EE components as Web services. However, there is an increasing need to access stored procedures as well as data and metadata through the Web services interface. Database Web services focuses on exposing functionality in the database as Web services and consuming external Web services from the database. Database Web services represent a database-centric vision of Web services and work in two directions: exposing database resources as Web services, and consuming external Web services from the database

Prerequisites

Back to Topic List

Before starting this module, you should have:

☐ **1.** Completed the Configuring Linux for the Installation of Oracle Database 10g lesson

☐ **2.** Completed the Installing the Oracle Database 10g on Linux lesson

☐ **3.** Completed the Installing Oracle9i JDeveloper on Linux lesson.

☐ **4.** Download and unzip dbws.zip into your working directory (i.e. /home/oracle/wkdir)

Understanding the Purchase Order Application

Back to List

This application you will deploy in this lesson illustrates calling a Web service from within an Oracle database. It shows calling-out Web services from Java stored procedure. The application demonstrates a purchasing model. JSP pages are used to design the front end for this application. These pages display the product catalog and allow you to browse through the catalog. You can purchase products of your interest. To purchase products, you have to provide login information. The authentication is performed by the Web service. This application uses a Credit Agency Web service that maintains a repository of customers and their credit card information. This Web service exposes two methods:

- To authorize the customer
- To bill the amount of purchase to the customer's credit card account.

How does the application work?

1. The products catalog is displayed and you can browse through the catalog.

2. To purchase any product, you have to enter the customer id.

3. The Credit Agency Web service maintains a repository of customers and their credit card information. Based on the login information supplied in step 2 the Web service authenticates you for buying products of your interest.

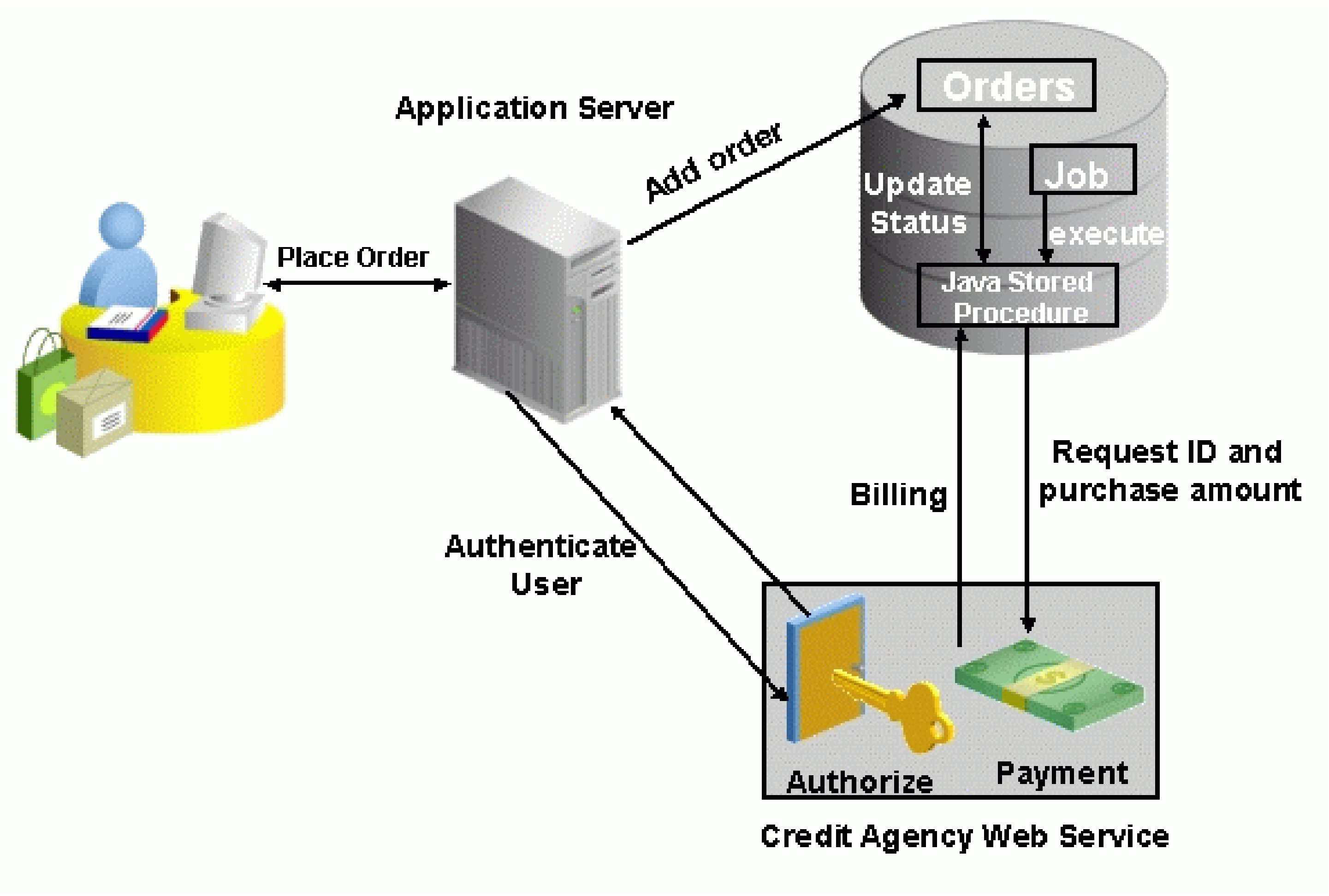
   a. If you are not a valid customer, the Web service returns 'INVALID_CUSTOMER.'
   b. If you are a valid customer, the Web service returns 'VALID_CUSTOMER.' The information about the products, that you have chosen, will be stored in the database. The status of the order will be stored as 'PENDING' and processed later.

4. A batch job runs every 6 minutes to process the 'PENDING' orders.

5. This job invokes the Credit Agency Web service to bill your account. The Web service is invoked with the customer information and the amount of purchase.

   a. If the amount in your account is less than the bill amount, then the Web service will returns a string 'INSUFFICIENT_FUNDS' and your account is restored back.
   b. If the amount in your account is sufficient, then the account is charged and a message 'SHIPPED' is returned.
   c. If the Web service does not recognize you, then it returns a string 'INVALID_CUSTOMER.'

6. Depending on the response from the Web service, the batch job updates the order status with the string returned by the Web service.



Files in the Application

WSClientSample

build.xml	Project build file for generating deployment file using ANT. You will use this file only if you are deploying the application to standalone OC4J
WSClientSample.jws	Oracle9 *i* JDeveloper workspace file
WSClientSample.jpr	Oracle9 *i* JDeveloper project file

WSClientSample/config

application.xml	Configuration file for the Application Server
WSCreate.sql	The SQL script file used to create a user and the tables required by the application
WSProcedure.sql	The SQL script used to publish the Java stored procedure to the database and create a batch job for invoking the Java stored procedure
Connection.properties	The properties file that has details of the database connection

WSClientSample/src/oracle/otnsamples/wsservices

CreditAgencyService.java	Class that implements a stateless Java Web service
ICreditAgencyService.java	The Web service interface that lists the methods exposed as Web service
CreditAgencyService.wsdl	The WSDL file for the Web service
ConnectionManager.java	A Java bean used for managing database connection

WSClientSample/src/oracle/otnsamples/wsclient

CreditAgencyServiceClient.java	Stored procedure that invokes the Web service for validation
CreditAgencyServiceStub.java	Client side stub file, for the Web service, that is generated by Oracle9 *i* JDeveloper
StoresBean.java	The Java bean used by the JSP to handle database operations
OrdersInfo.java	Value object used to pass order information between the JSP and the Java bean
ProductsInfo.java	Value object used to pass product information between the JSP and the Java bean

WSClientSample/webroot

Products.jsp	Displays the product catalog, and accepts orders
ViewOrders.jsp	Retrieves order status for a specified order ID
ShipOrders.jsp	Retrieves order information from the database for the shipping service
ErrorHandler.jsp	Error page for JSP pages
WEB-INF/web.xml	Configuration file for the Web container

Installing the Purchase Application

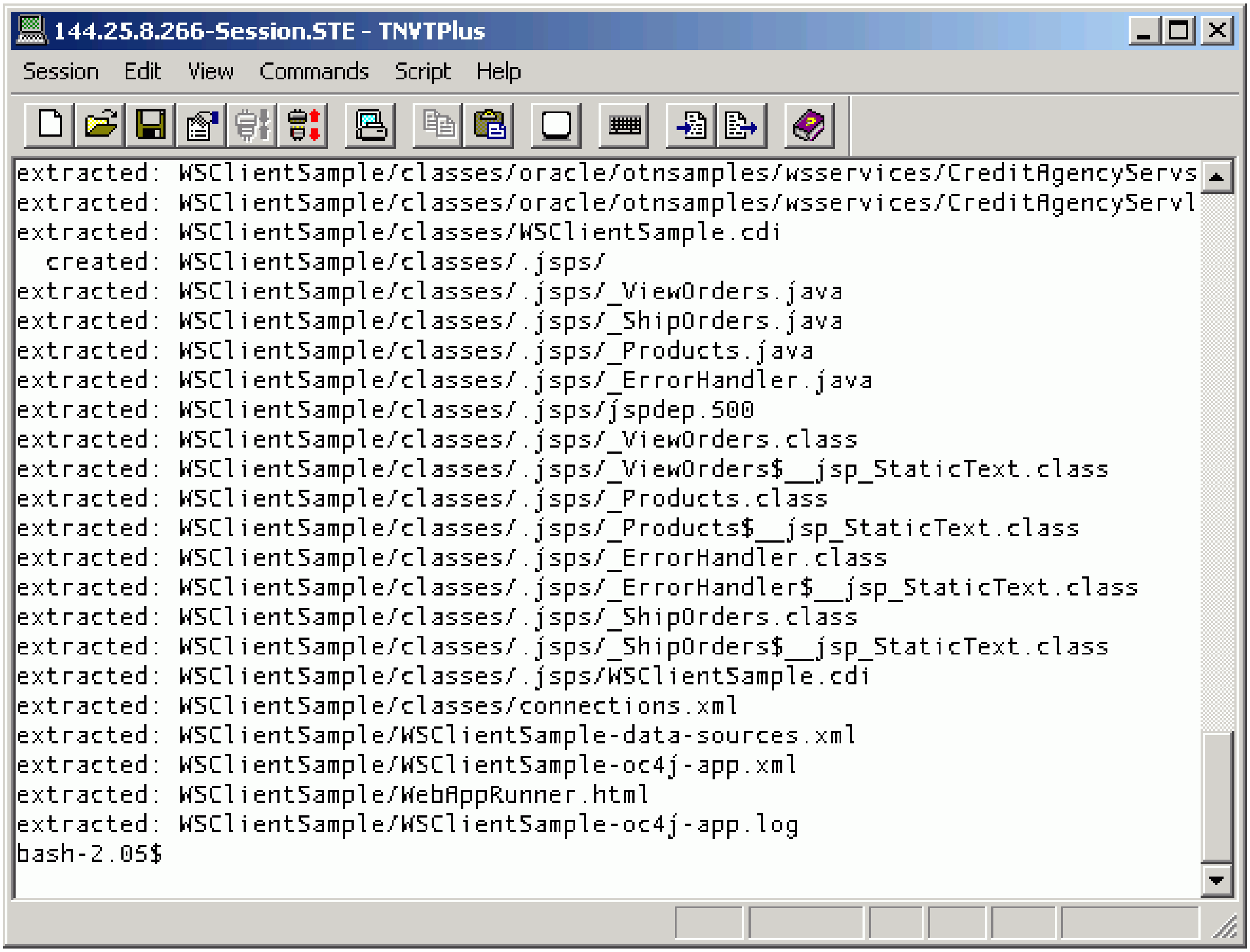
Back to List

To install the Purchase application, perform the following steps:

☐ **1.** Open a terminal window and execute the following commands:

```
cd /home/oracle/wkdir
```

```
jar xvf dbwebservice.jar
```



```
extracted: WSClientSample/classes/oracle/otnsamples/wsservices/CreditAgencyServs
extracted: WSClientSample/classes/oracle/otnsamples/wsservices/CreditAgencyServl
extracted: WSClientSample/classes/WSClientSample.cdi
  created: WSClientSample/classes/.jsps/
extracted: WSClientSample/classes/.jsps/_ViewOrders.java
extracted: WSClientSample/classes/.jsps/_ShipOrders.java
extracted: WSClientSample/classes/.jsps/_Products.java
extracted: WSClientSample/classes/.jsps/_ErrorHandler.java
extracted: WSClientSample/classes/.jsps/jspdep.500
extracted: WSClientSample/classes/.jsps/_ViewOrders.class
extracted: WSClientSample/classes/.jsps/_ViewOrders$__jsp_StaticText.class
extracted: WSClientSample/classes/.jsps/_Products.class
extracted: WSClientSample/classes/.jsps/_Products$__jsp_StaticText.class
extracted: WSClientSample/classes/.jsps/_ErrorHandler.class
extracted: WSClientSample/classes/.jsps/_ErrorHandler$__jsp_StaticText.class
extracted: WSClientSample/classes/.jsps/_ShipOrders.class
extracted: WSClientSample/classes/.jsps/_ShipOrders$__jsp_StaticText.class
extracted: WSClientSample/classes/.jsps/WSClientSample.cdi
extracted: WSClientSample/classes/connections.xml
extracted: WSClientSample/WSClientSample-data-sources.xml
extracted: WSClientSample/WSClientSample-oc4j-app.xml
extracted: WSClientSample/WebAppRunner.html
extracted: WSClientSample/WSClientSample-oc4j-app.log
bash-2.05$
```

Setting up the database

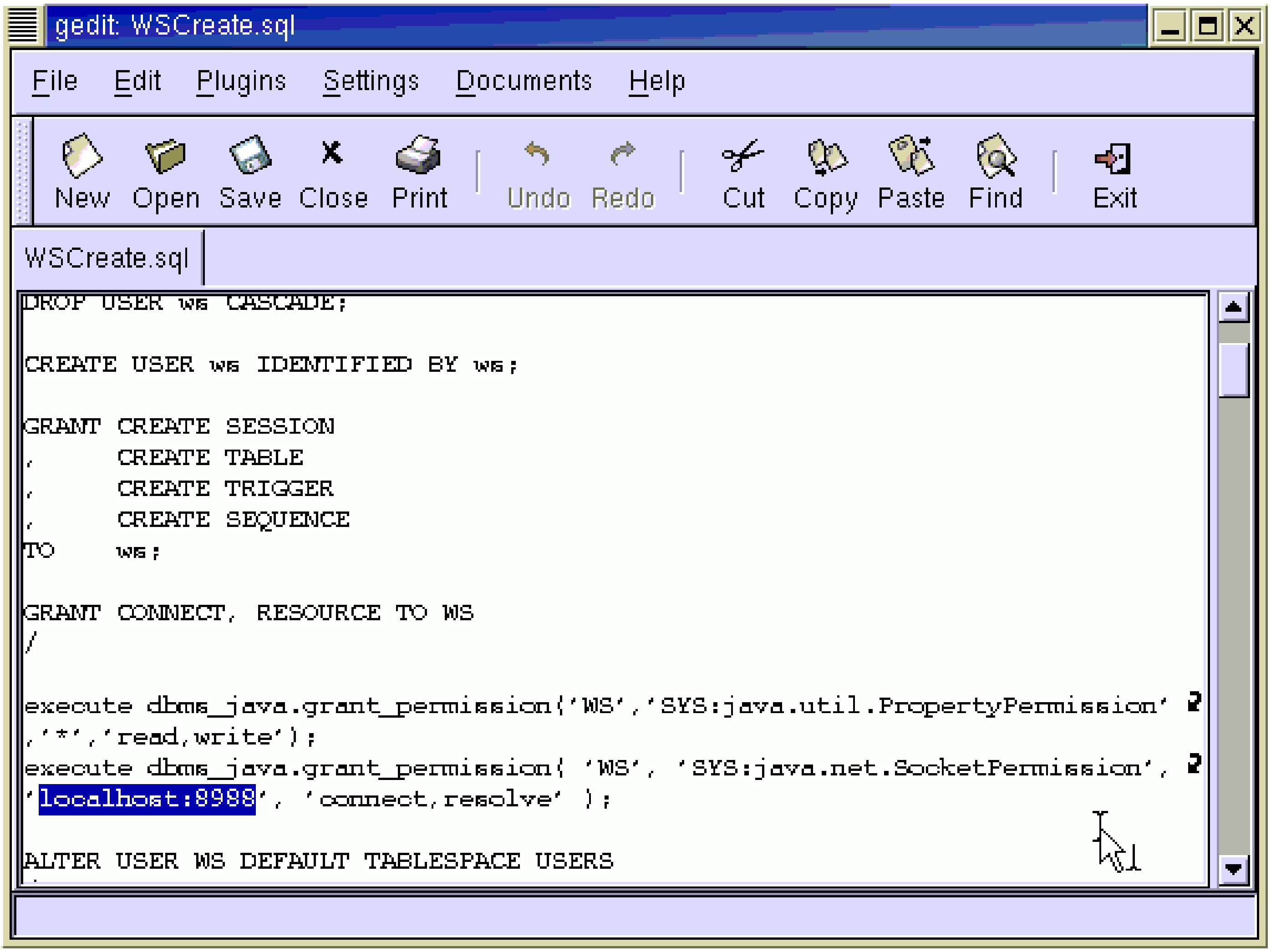
Back to List

Perform the following to setup the database:

☐ **1.** Open this file with an editor: **`/home/oracle/wkdir/WSClientSample/config/WSCreate.sql`**

Verify that the following line specifies **localhost:8988,** correct it if necessary:

```
execute dbms_java.grant_permission('WS','SYS:java.net.SocketPermission',
  '
localhost:8988
','connect,resolve');
```

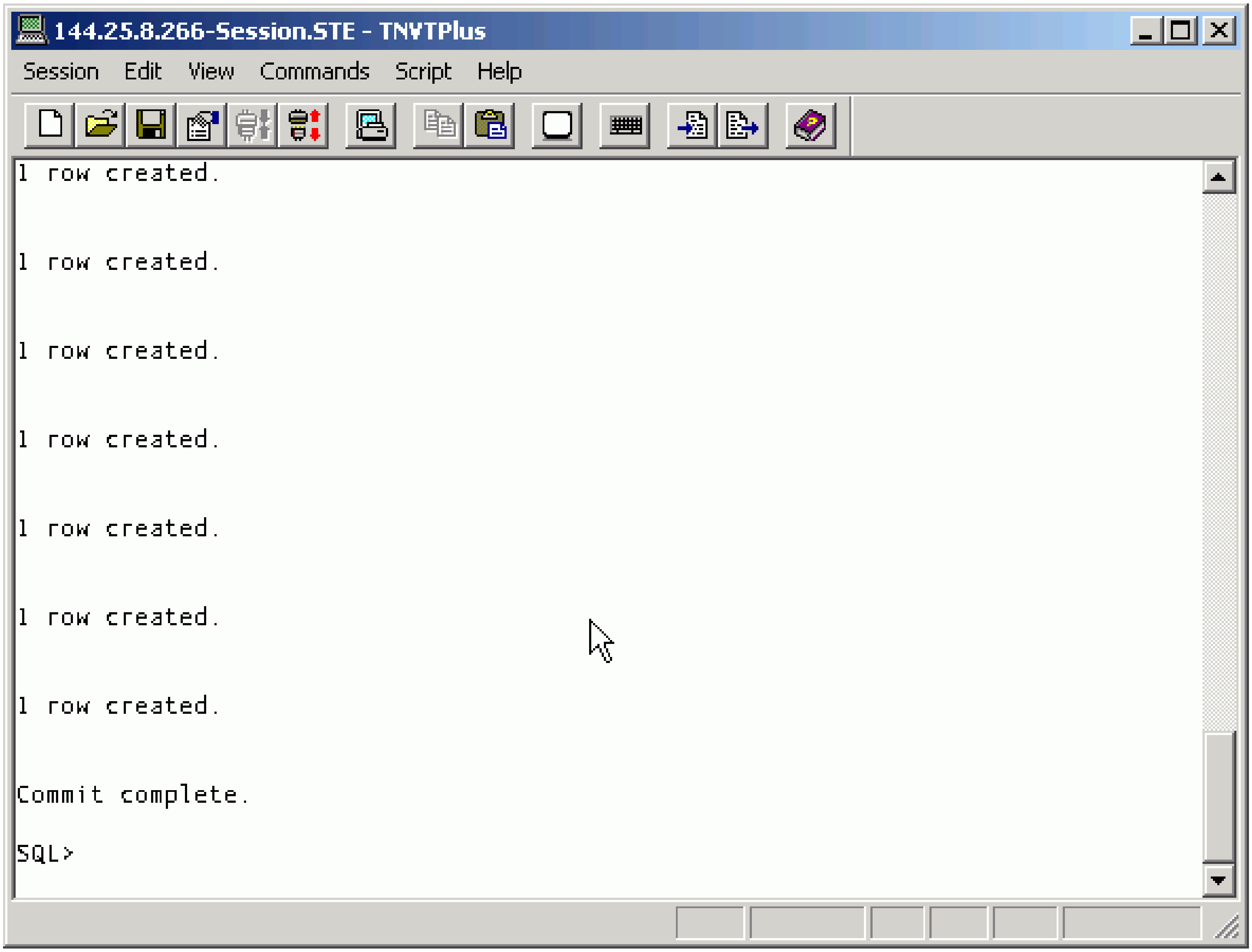


```
DROP USER ws CASCADE;

CREATE USER ws IDENTIFIED BY ws;

GRANT CREATE SESSION
,     CREATE TABLE
,     CREATE TRIGGER
,     CREATE SEQUENCE
TO    ws;

GRANT CONNECT, RESOURCE TO WS
/

execute dbms_java.grant_permission('WS','SYS:java.util.PropertyPermission'
,'*','read,write');
execute dbms_java.grant_permission( 'WS', 'SYS:java.net.SocketPermission',
'localhost:8988', 'connect,resolve' );

ALTER USER WS DEFAULT TABLESPACE USERS
```

☐ **2.** From your terminal window, execute the following commands:

```
cd /home/oracle/wkdir/WSClientSample/config
sqlplus /nolog
connect / as sysdba
@WSCreate
exit
```

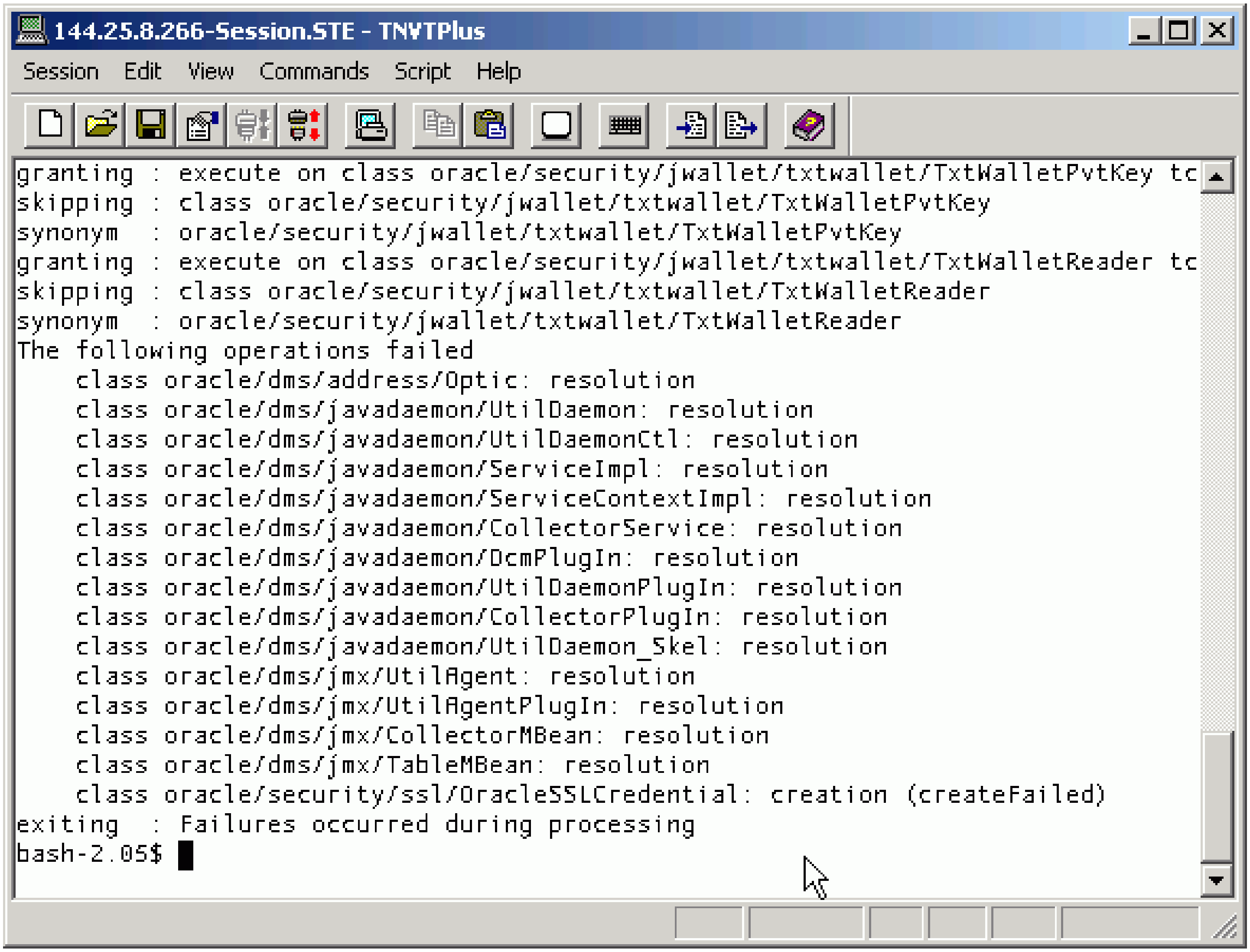
This script will prompt you to enter password for user `system` and the tns name. Enter **oracle** and **orcl** respectively.



```
1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


1 row created.


Commit complete.

SQL>
```

☐ **3.** The Web service client is a Java stored procedure. This client application will require some jar files to be present in the database to invoke the external Web service. From your terminal window, run the following commands:

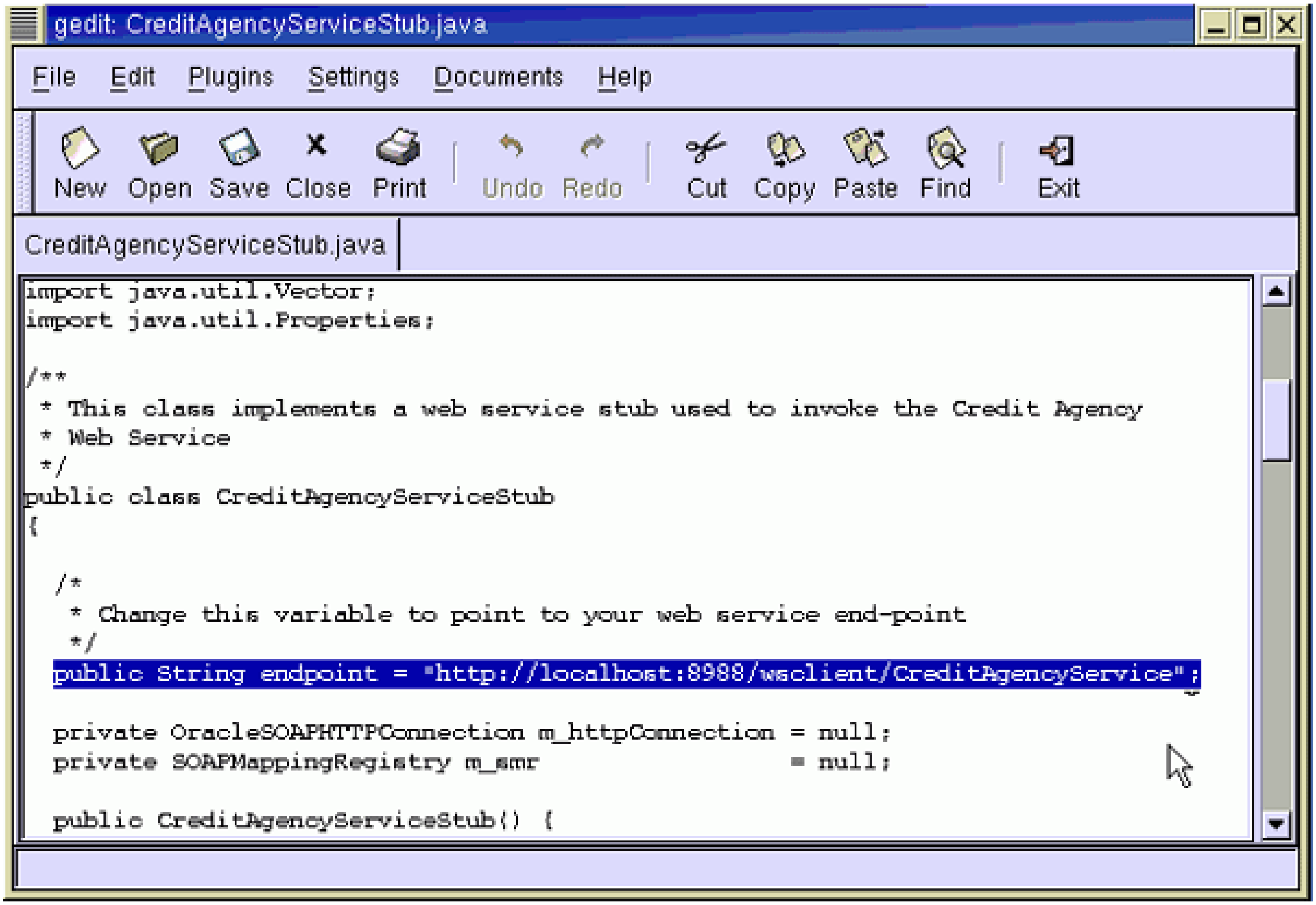
```
cd /home/oracle/wkdir
chmod 777 load.sh
./load.sh
```



```
granting : execute on class oracle/security/jwallet/txtwallet/TxtWalletPvtKey tc
skipping : class oracle/security/jwallet/txtwallet/TxtWalletPvtKey
synonym  : oracle/security/jwallet/txtwallet/TxtWalletPvtKey
granting : execute on class oracle/security/jwallet/txtwallet/TxtWalletReader tc
skipping : class oracle/security/jwallet/txtwallet/TxtWalletReader
synonym  : oracle/security/jwallet/txtwallet/TxtWalletReader
The following operations failed
    class oracle/dms/address/Optic: resolution
    class oracle/dms/javadaemon/UtilDaemon: resolution
    class oracle/dms/javadaemon/UtilDaemonCtl: resolution
    class oracle/dms/javadaemon/ServiceImpl: resolution
    class oracle/dms/javadaemon/ServiceContextImpl: resolution
    class oracle/dms/javadaemon/CollectorService: resolution
    class oracle/dms/javadaemon/DcmPlugIn: resolution
    class oracle/dms/javadaemon/UtilDaemonPlugIn: resolution
    class oracle/dms/javadaemon/CollectorPlugIn: resolution
    class oracle/dms/javadaemon/UtilDaemon_Skel: resolution
    class oracle/dms/jmx/UtilAgent: resolution
    class oracle/dms/jmx/UtilAgentPlugIn: resolution
    class oracle/dms/jmx/CollectorMBean: resolution
    class oracle/dms/jmx/TableMBean: resolution
    class oracle/security/ssl/OracleSSLCredential: creation (createFailed)
exiting  : Failures occurred during processing
bash-2.05$
```

☐ **4.** Check the Web service endpoint in the client. Open the file **`/home/oracle/wkdir/WSClientSample/src/oracle`**
`/otnsamples/wsclient/CreditAgencyServiceStub.java`
Verify that the string variable, `endpoint` , points to your Web service end-point as follows:

```
public String endpoint = "http://localhost:8988/wsclient/CreditAgencyService";
```



```
import java.util.Vector;
import java.util.Properties;

/**
 * This class implements a web service stub used to invoke the Credit Agency
 * Web Service
 */
public class CreditAgencyServiceStub
{

  /*
   * Change this variable to point to your web service end-point
   */
  public String endpoint = "http://localhost:8988/wsclient/CreditAgencyService";

  private OracleSOAPHTTPConnection m_httpConnection = null;
  private SOAPMappingRegistry m_smr                 = null;

  public CreditAgencyServiceStub() {
```

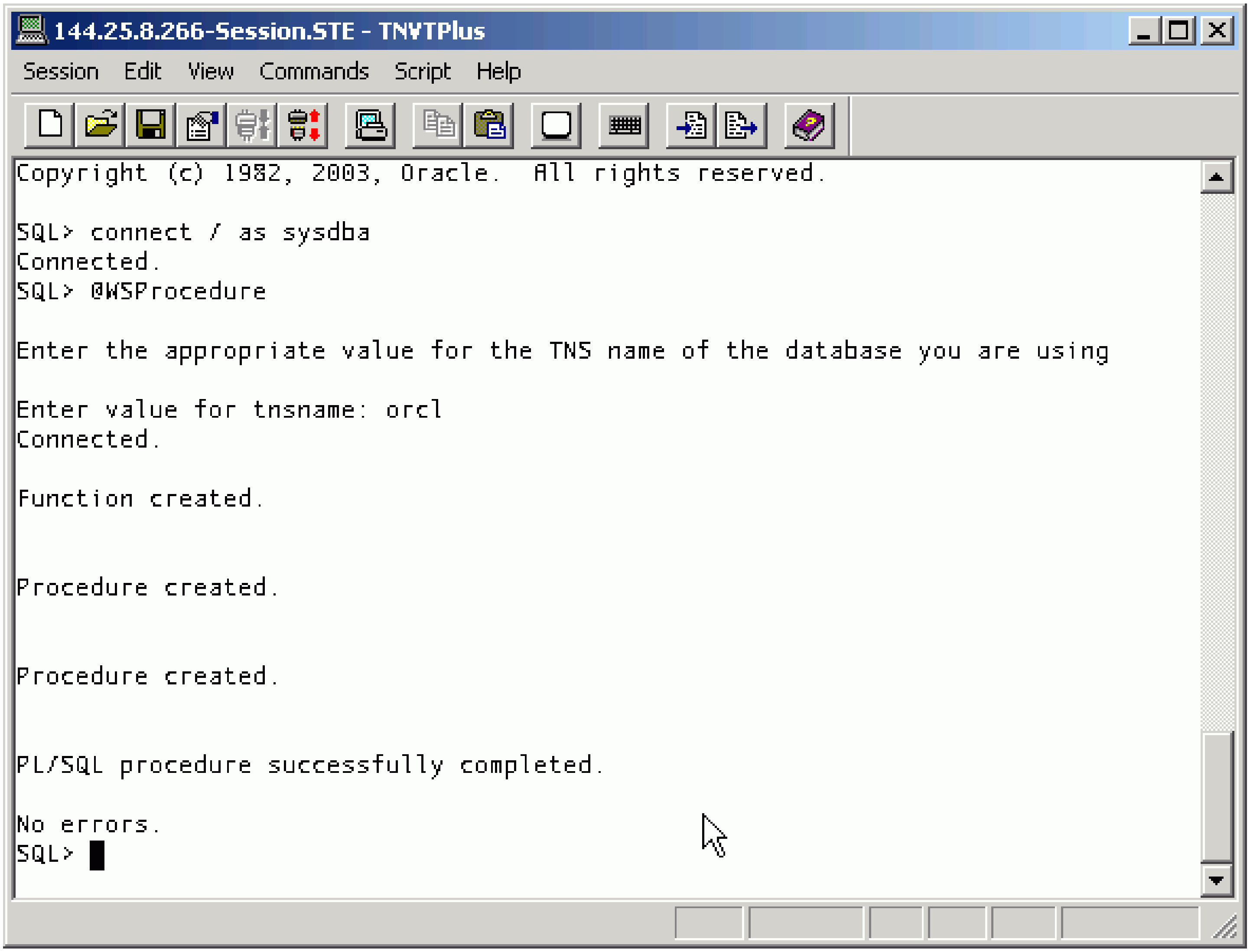
☐ **5.** Load the Web service client and stub file to the database. Switch back to your terminal window and issue the following commands:

```
cd /home/oracle/wkdir/WSClientSample/src/oracle/otnsamples/wsclient
loadjava -thin -user ws/ws@localhost:1521:orcl
   CreditAgencyServiceStub.java CreditAgencyServiceClient.java
```

☐ **6.** Publish the Web service client and create a database job that can invoke the Web service. From your terminal window execute the following commands:

```
cd /home/oracle/wkdir/WSClientSample/config
sqlplus /nolog
connect / as sysdba;
@WSProcedure
exit
```

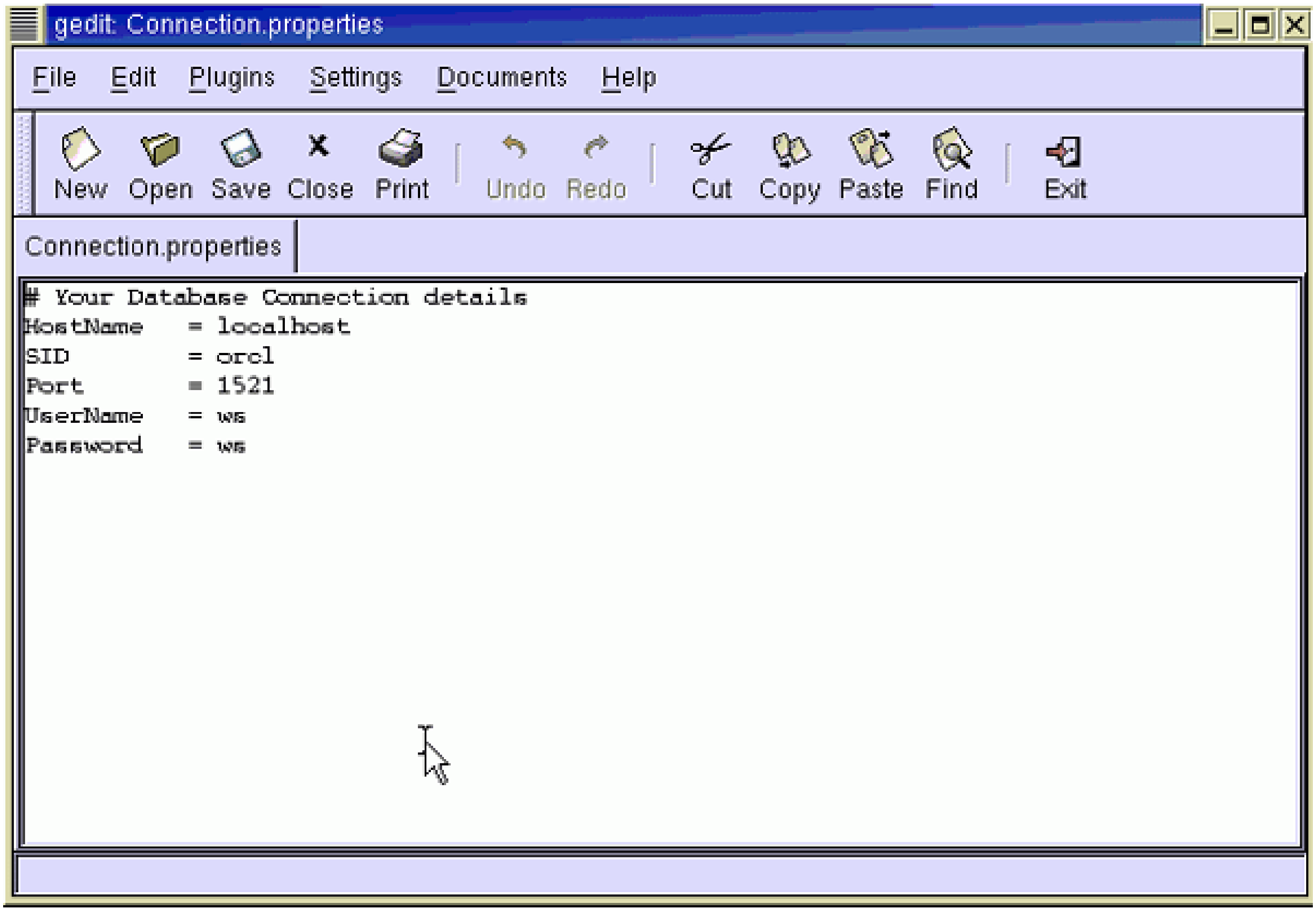
When prompted for the tnsnames value, type **orcl** . This SQL script publishes the stored procedure created in the previous step to the database and creates a job that invokes CreditAgencyServiceClient.java once every six minutes.



```
Copyright (c) 1982, 2003, Oracle.  All rights reserved.

SQL> connect / as sysdba
Connected.
SQL> @WSProcedure

Enter the appropriate value for the TNS name of the database you are using

Enter value for tnsname: orcl
Connected.

Function created.


Procedure created.


Procedure created.


PL/SQL procedure successfully completed.

No errors.
SQL>
```

☐ **7.** Open the file **`/home/oracle/wkdir/WSClientSample/config/Connection.properties`** and verify that:

HostName = **localhost**
SID = **orcl**
Port = **1521**
UserName = **ws**
Password = **ws**



```
# Your Database Connection details
HostName    = localhost
SID         = orcl
Port        = 1521
UserName    = ws
Password    = ws
```

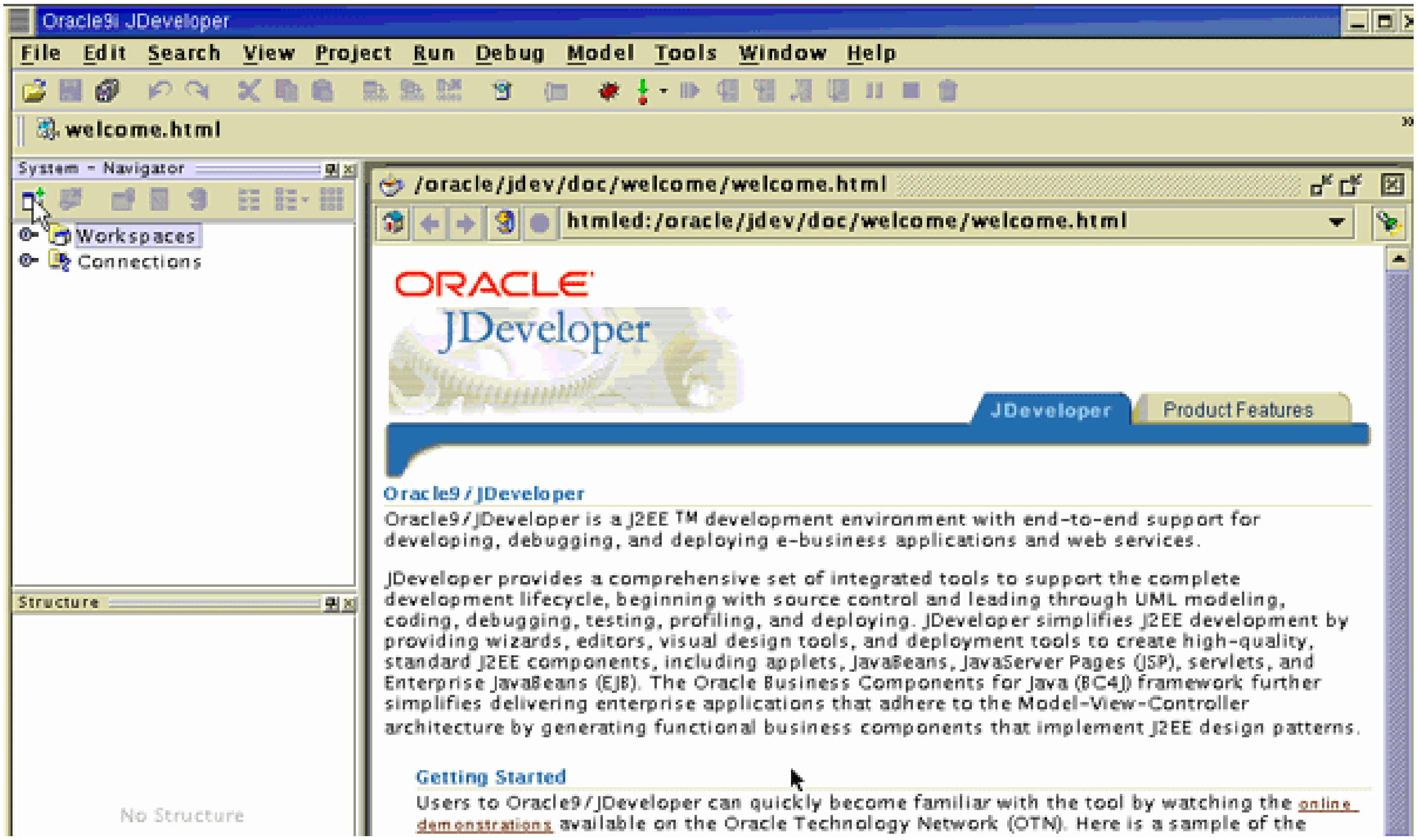
Open the Application

Back to List

Now you will open the application in JDeveloper. Perform the following:

☐ **1.** Launch **JDeveloper** .

☐ **2.** Click and highlight **Workspaces** in the System-Navigator window. Click **+** icon at the top left hand corner of System-Navigator.


Oracle9i JDeveloper
File Edit Search View Project Run Debug Model Tools Window Help
welcome.html
System - Navigator
Workspaces
Connections
Structure
No Structure
/oracle/jdev/doc/welcome/welcome.html
htmled:/oracle/jdev/doc/welcome/welcome.html
ORACLE
JDeveloper
JDeveloper
Product Features

Oracle9/JDeveloper

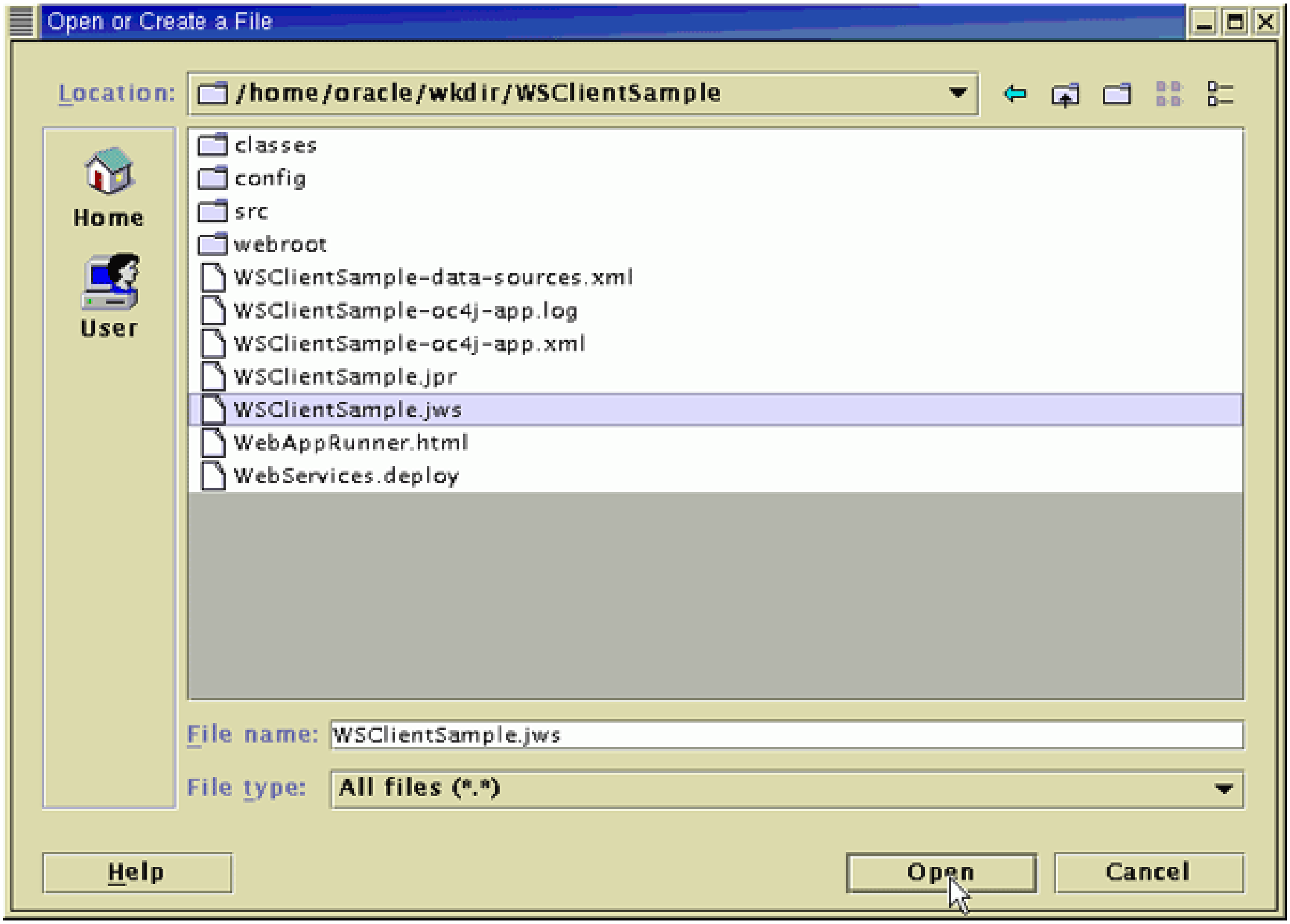
Oracle9/JDeveloper is a J2EE TM development environment with end-to-end support for developing, debugging, and deploying e-business applications and web services.

JDeveloper provides a comprehensive set of integrated tools to support the complete development lifecycle, beginning with source control and leading through UML modeling, coding, debugging, testing, profiling, and deploying. JDeveloper simplifies J2EE development by providing wizards, editors, visual design tools, and deployment tools to create high-quality, standard J2EE components, including applets, JavaBeans, JavaServer Pages (JSP), servlets, and Enterprise JavaBeans (EJB). The Oracle Business Components for Java (BC4J) framework further simplifies delivering enterprise applications that adhere to the Model-View-Controller architecture by generating functional business components that implement J2EE design patterns.

Getting Started

Users to Oracle9/JDeveloper can quickly become familiar with the tool by watching the online demonstrations available on the Oracle Technology Network (OTN). Here is a sample of the

☐ **3.** Select and open the file **`/home/oracle/wkdir/WSClientSample/WSClientSample.jws`** .

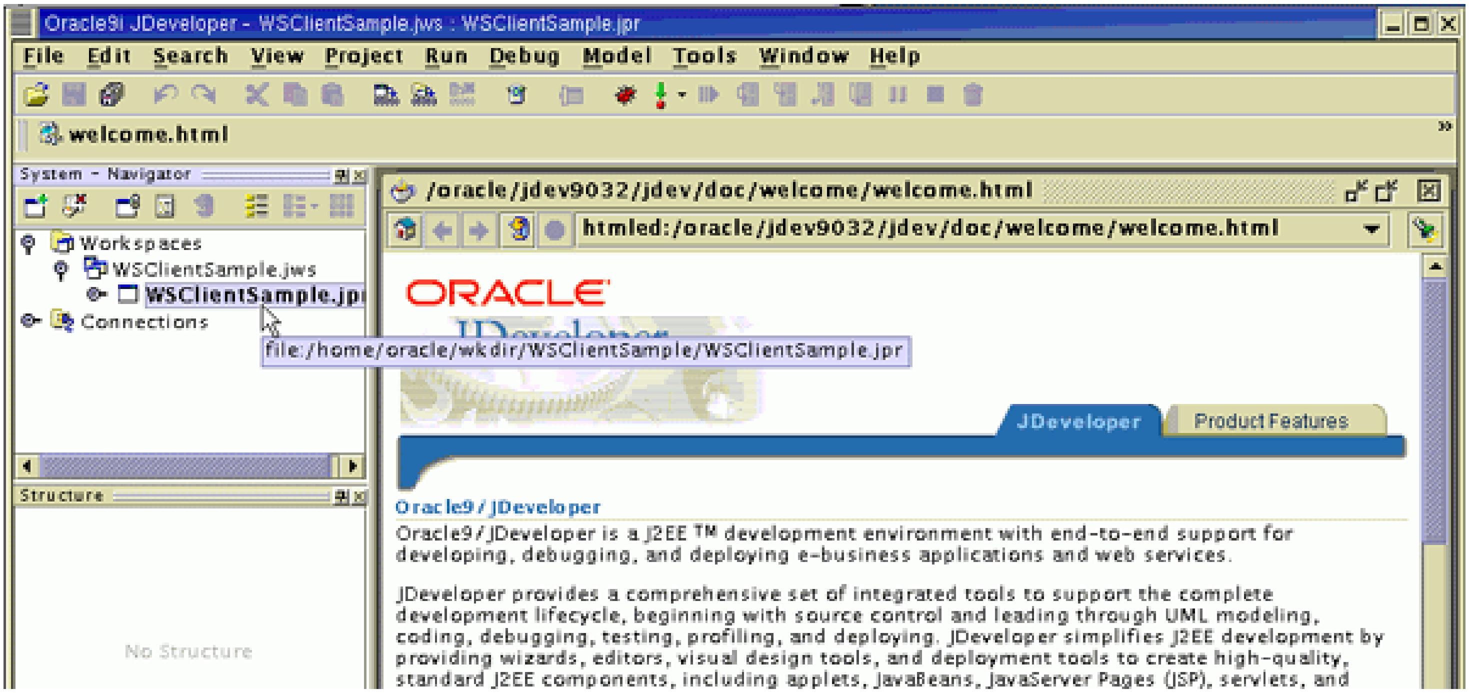


Location: /home/oracle/wkdir/WSClientSample

File name: WSClientSample.jws

File type: All files (*.*)

☐ **4.** Expand the tree to ensure that the project **WSClientSample.jpr** is included.


Oracle9i JDeveloper - WSClientSample.jws : WSClientSample.jpr
File Edit Search View Project Run Debug Model Tools Window Help
welcome.html
System - Navigator
Workspaces
WSClientSample.jws
WSClientSample.jpr
Connections
file:/home/oracle/wkdir/WSClientSample/WSClientSample.jpr
Structure
No Structure
/oracle/jdev9032/jdev/doc/welcome/welcome.html
htmled:/oracle/jdev9032/jdev/doc/welcome/welcome.html
ORACLE
JDeveloper
Product Features

Oracle9/JDeveloper

Oracle9/JDeveloper is a J2EE TM development environment with end-to-end support for developing, debugging, and deploying e-business applications and web services.

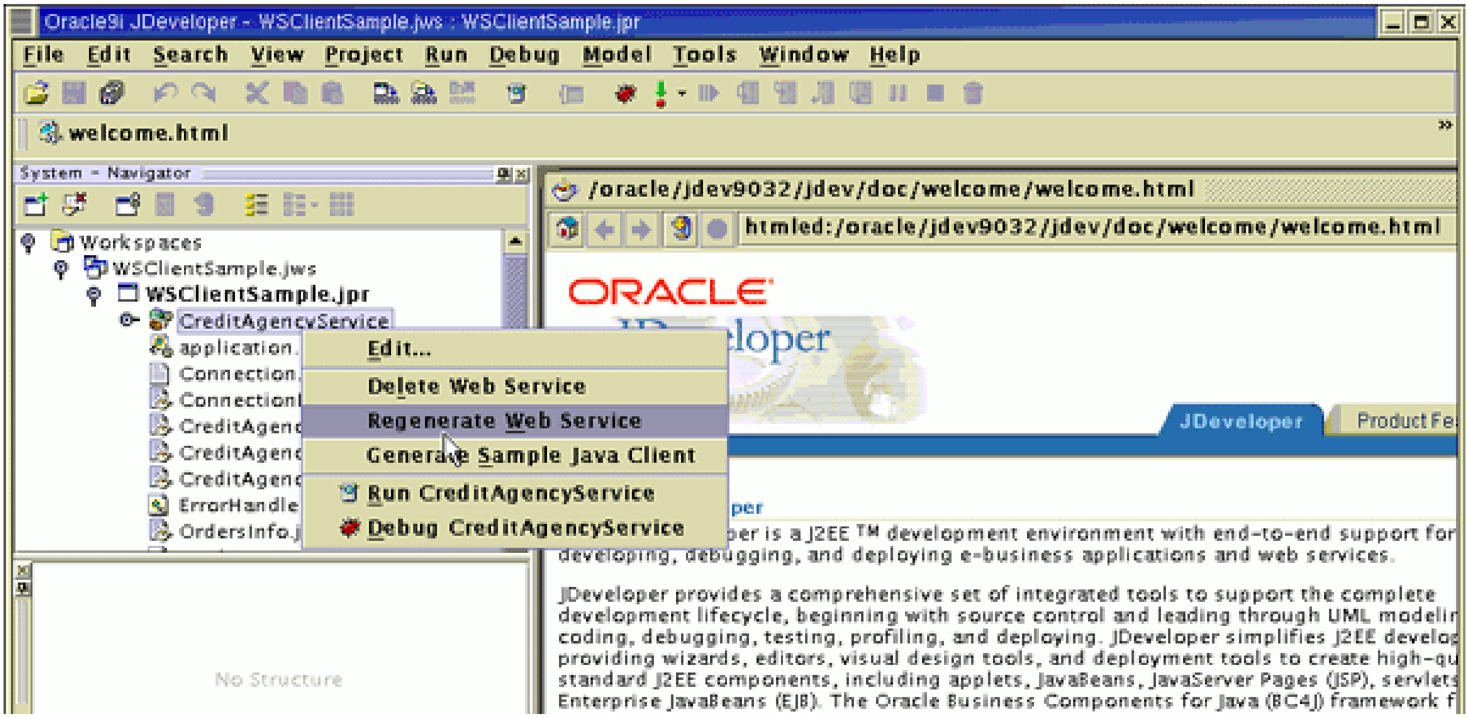
JDeveloper provides a comprehensive set of integrated tools to support the complete development lifecycle, beginning with source control and leading through UML modeling, coding, debugging, testing, profiling, and deploying. JDeveloper simplifies J2EE development by providing wizards, editors, visual design tools, and deployment tools to create high-quality, standard J2EE components, including applets, JavaBeans, JavaServer Pages (JSP), servlets, and

Run the Application

Back to List

The table USERS in the ws schema is populated with two user names:otn-user and oracle-user. The credit limit for otn-user is $2000 and the credit limit for oracle-user is $3000. You will run the application and place an order by otn-user for under $2000 and another order that exceeds the credit limit of $2000. Perform the following:

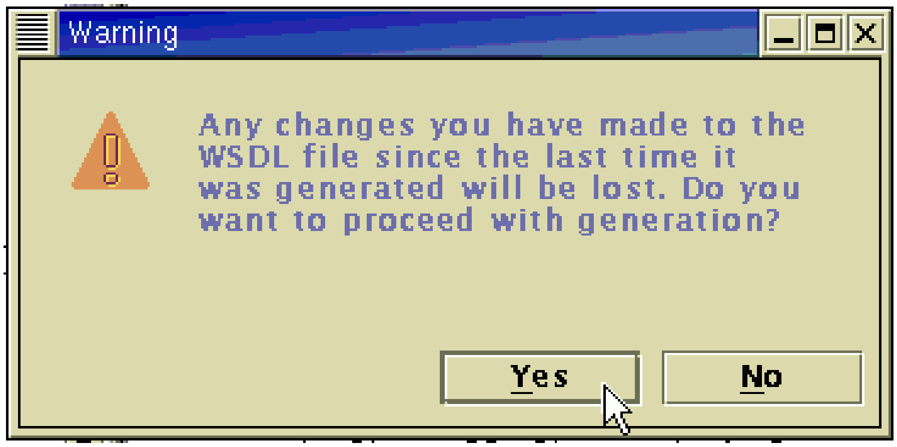
☐ **1.** Right-click on **CreditAgencyService** and select **Regenerate Web Service** .



per is a J2EE ™ development environment with end-to-end support for developing, debugging, and deploying e-business applications and web services.

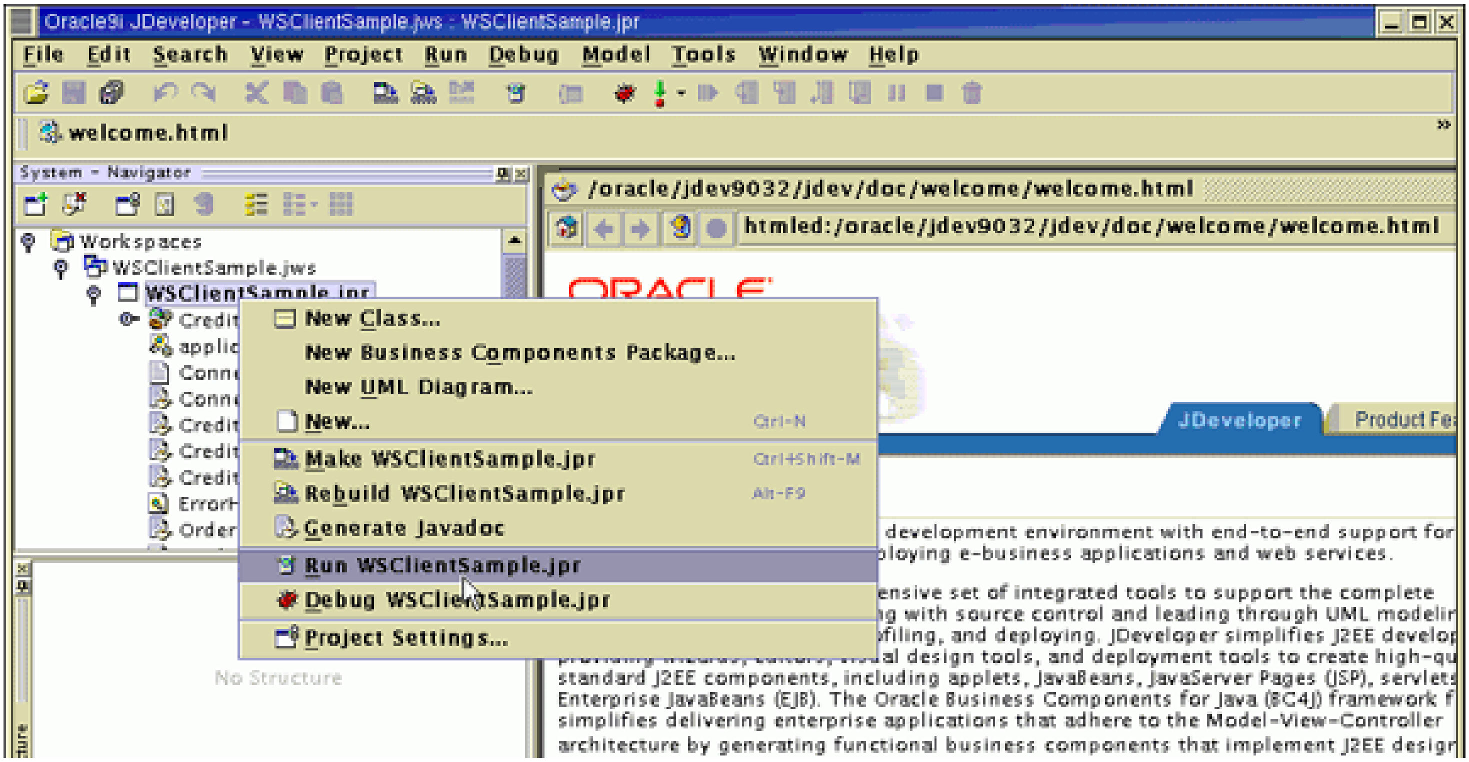
JDeveloper provides a comprehensive set of integrated tools to support the complete development lifecycle, beginning with source control and leading through UML modelin coding, debugging, testing, profiling, and deploying. JDeveloper simplifies J2EE develop providing wizards, editors, visual design tools, and deployment tools to create high-qu standard J2EE components, including applets, JavaBeans, JavaServer Pages (JSP), servlets Enterprise JavaBeans (EJB). The Oracle Business Components for Java (BC4J) framework f

☐ **2.** Click **Yes** to close the warning window.

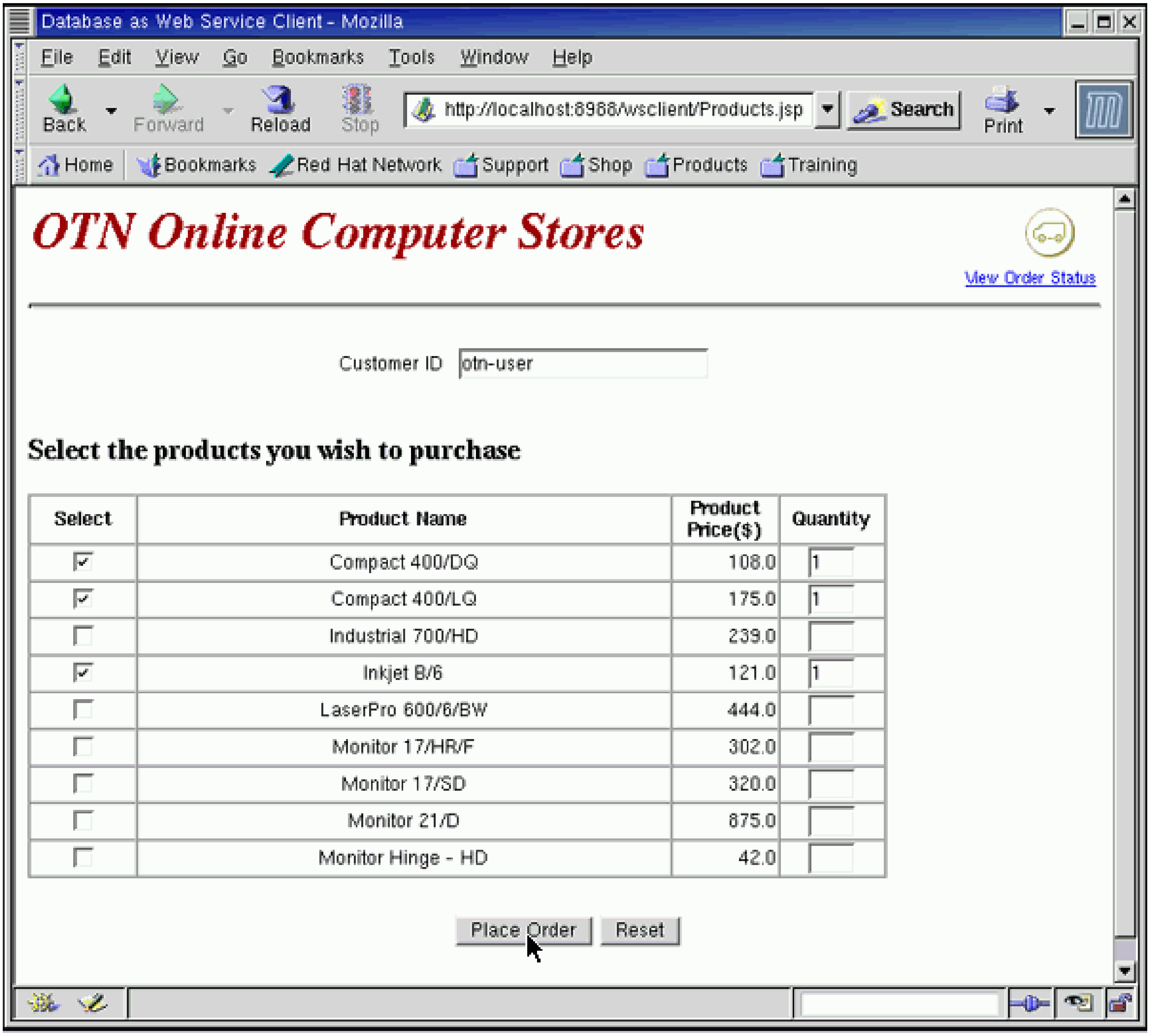


Any changes you have made to the WSDL file since the last time it was generated will be lost. Do you want to proceed with generation?

☐ **3.** Right-click on the project **WSClientSample.jpr** and select **Run WSClientSample.jpr** .


Oracle9i JDeveloper - WSClientSample.jws : WSClientSample.jpr
File Edit Search View Project Run Debug Model Tools Window Help
welcome.html
System - Navigator
Workspaces
WSClientSample.jws
WSClientSample.jpr
New Class...
New Business Components Package...
New UML Diagram...
New... Ctrl-N
Make WSClientSample.jpr Ctrl+Shift-M
Rebuild WSClientSample.jpr Alt-F9
Generate Javadoc
Run WSClientSample.jpr
Debug WSClientSample.jpr
Project Settings...
No Structure
/oracle/jdev9032/jdev/doc/welcome/welcome.html
htmled:/oracle/jdev9032/jdev/doc/welcome/welcome.html
ORACLE
JDeveloper
Product Fe

☐ **4.** When you run the application, you will see the screen shown below on the default browser. Enter **otn-user** for Customer ID and choose some of the products. If you do not specify the quantity, it defaults to 1. Then click **Place Order** .


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
http://localhost:8988/wsclient/Products.jsp
Search
Print
Home
Bookmarks
Red Hat Network
Support
Shop
Products
Training
View Order Status
Place Order
Reset

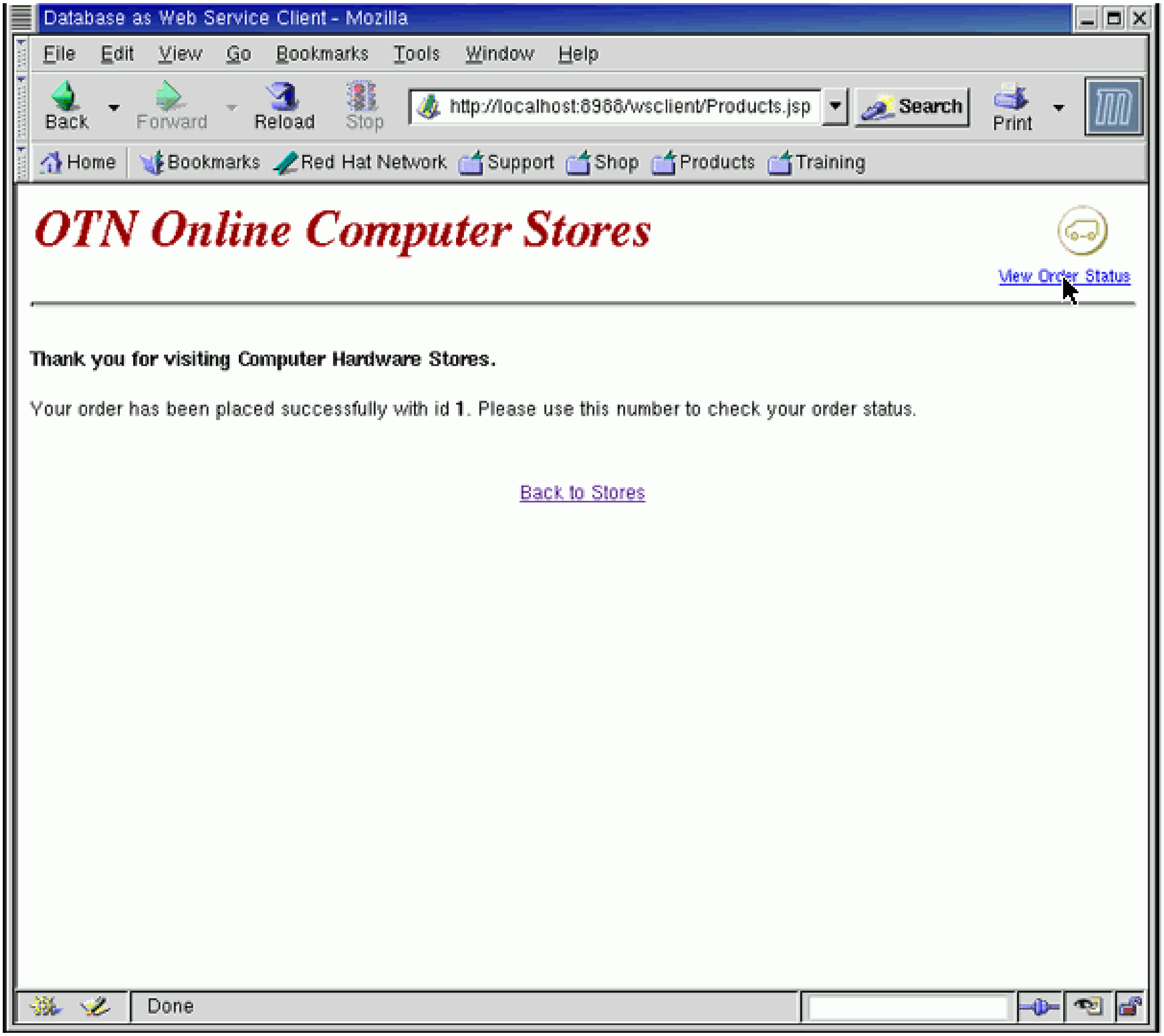
OTN Online Computer Stores

Customer ID otn-user

Select the products you wish to purchase

Select	Product Name	Product Price($)	Quantity
☑	Compact 400/DQ	108.0	1
☑	Compact 400/LQ	175.0	1
☐	Industrial 700/HD	239.0	
☑	Inkjet B/6	121.0	1
☐	LaserPro 600/6/BW	444.0	
☐	Monitor 17/HR/F	302.0	
☐	Monitor 17/SD	320.0	
☐	Monitor 21/D	875.0	
☐	Monitor Hinge - HD	42.0	

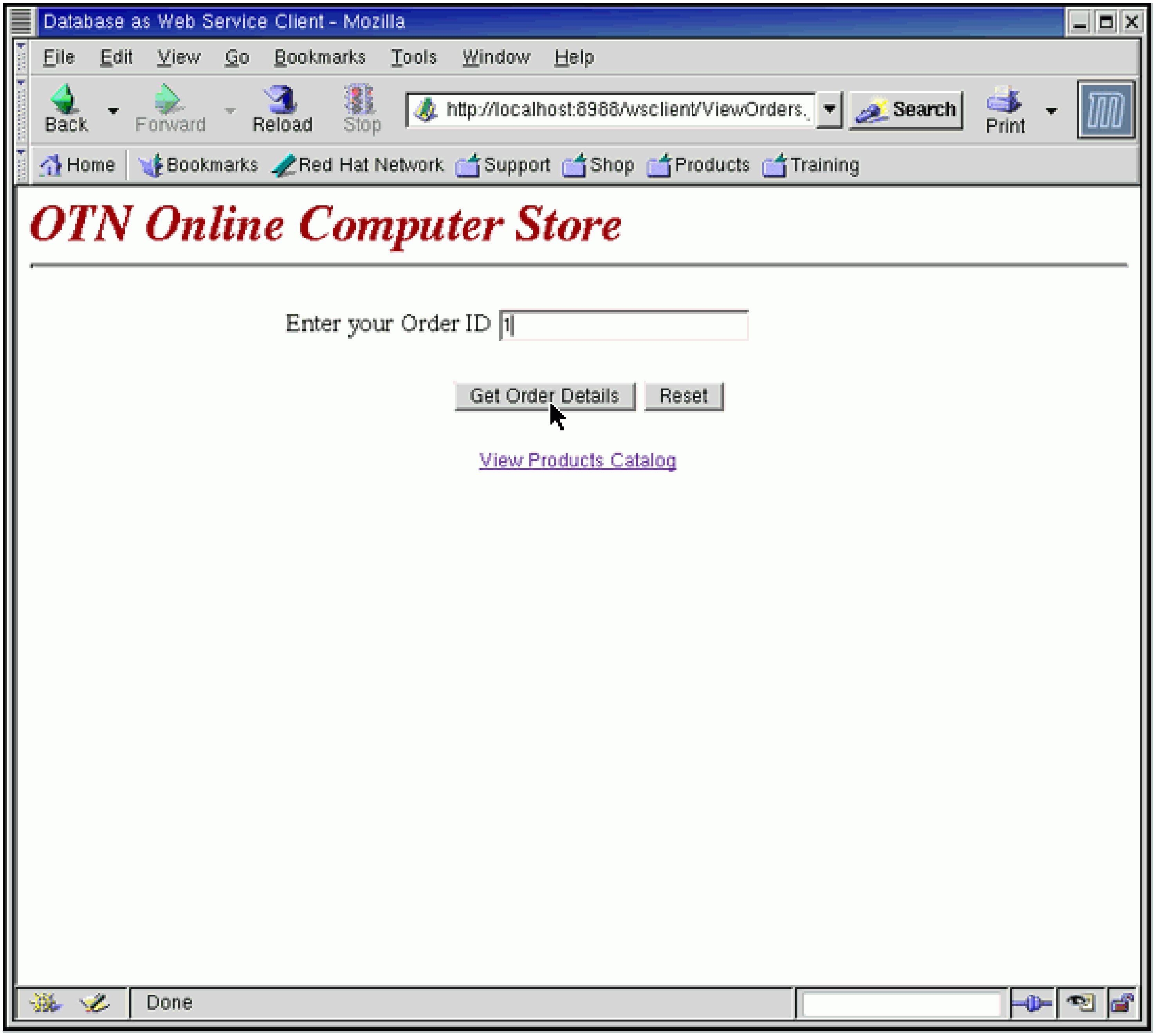
☐ **5.** Observe that the order is placed successfully with order id 1. Click on **View Order Status** .


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:8988/wsclient/Products.jsp
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
OTN Online Computer Stores
View Order Status
Back to Stores
Done

Thank you for visiting Computer Hardware Stores.

Your order has been placed successfully with id **1**. Please use this number to check your order status.

☐ **6.** Type the order id **1** in the field and click **Get Order Details** .

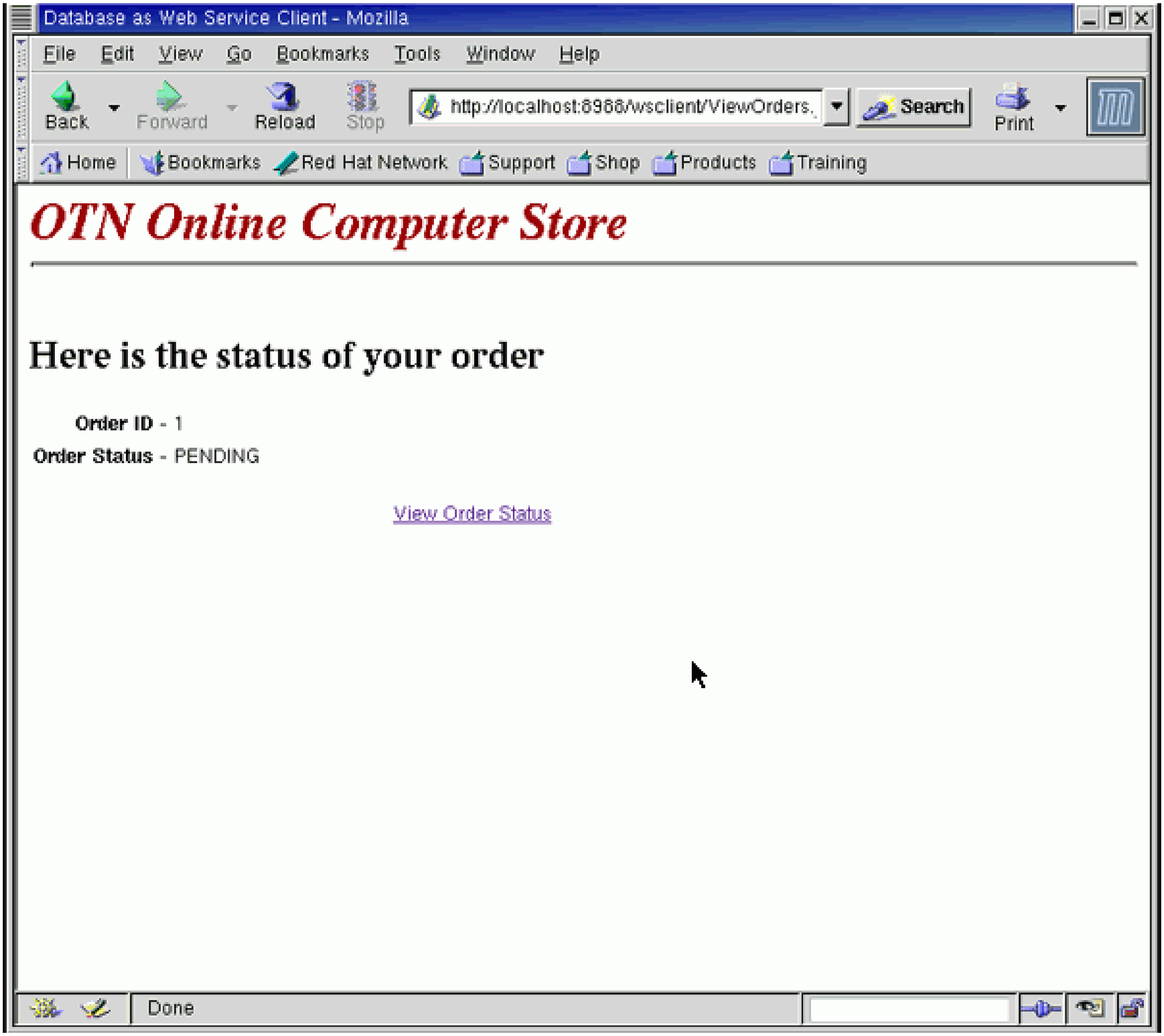

Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:8988/wsclient/ViewOrders.
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
Get Order Details
Reset
View Products Catalog
Done

OTN Online Computer Store

Enter your Order ID 1

☐ **7.** You have successfully placed the order. As the vendor, you will now go to the shipment module to check the pending orders. In the URL change ViewOrders.jsp to **ShipOrders.jsp** so the URL is now the following:

```
http://host:8988/wsclient/ShipOrders.jsp
```


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:8988/wsclient/ViewOrders.
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
Done

OTN Online Computer Store

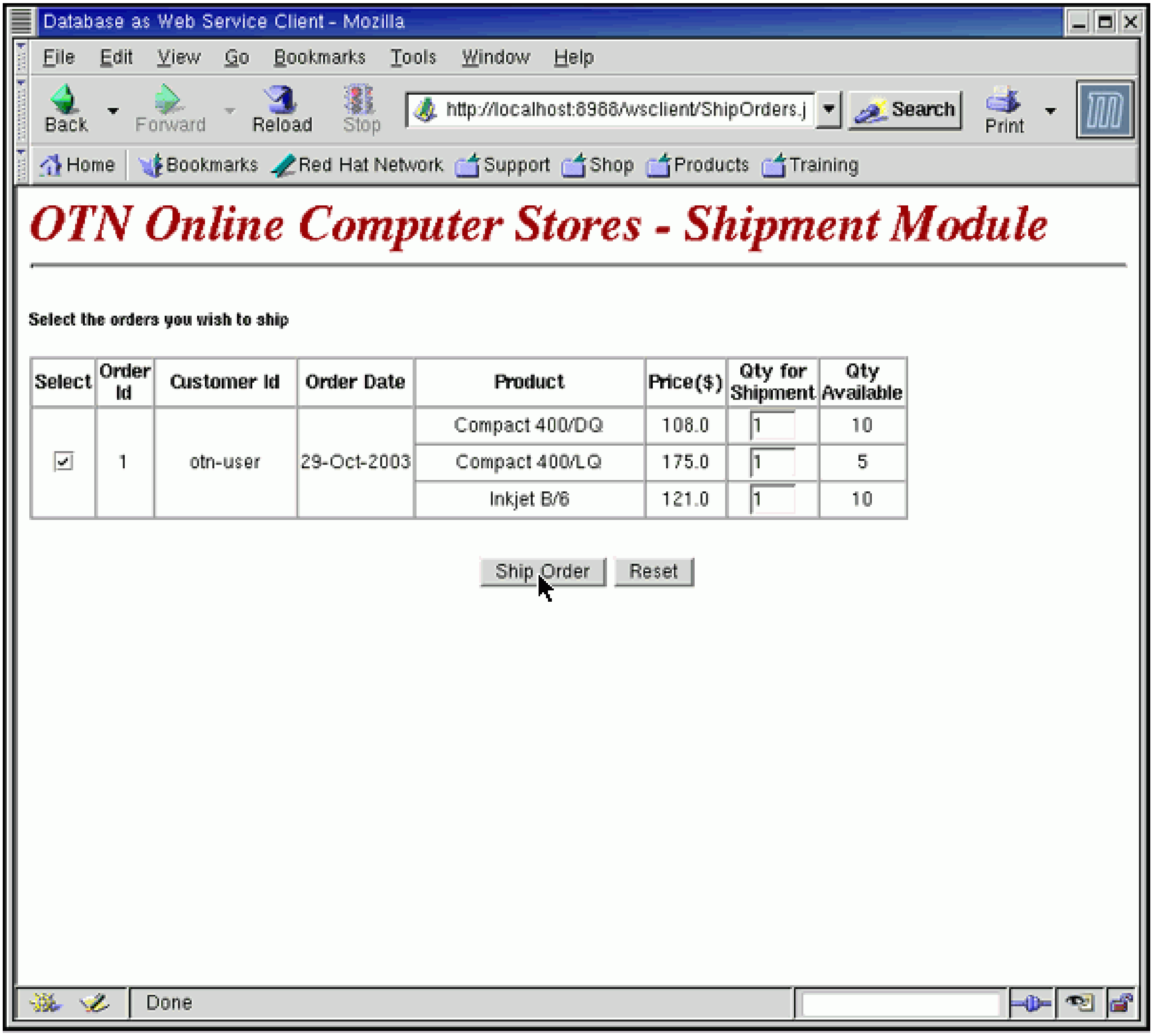
Here is the status of your order

Order ID - 1

Order Status - PENDING

View Order Status

☐ **8.** Select your order for shipping and then click **Ship Order** .

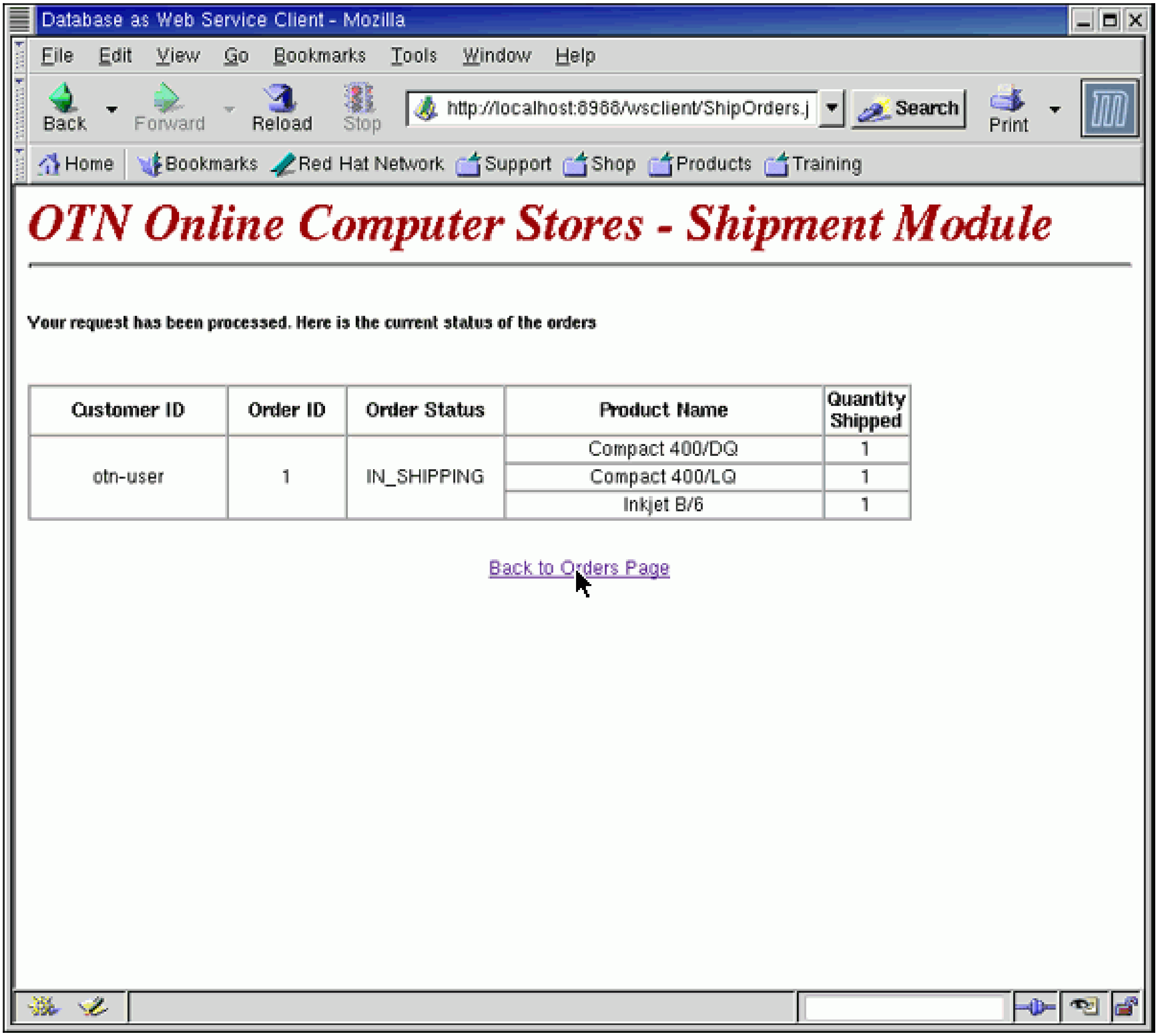

Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:8988/wsclient/ShipOrders.j
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
Ship Order
Reset
Done

OTN Online Computer Stores - Shipment Module

Select the orders you wish to ship

Select	Order Id	Customer Id	Order Date	Product	Price($)	Qty for Shipment	Qty Available
☑	1	otn-user	29-Oct-2003	Compact 400/DQ	108.0	1	10
				Compact 400/LQ	175.0	1	5
				Inkjet B/6	121.0	1	10

☐ **9.** Observe that the status of the order has changed from PENDING to IN_SHIPPING. The shipping process is automated and configured to run every six minutes. The process scans the database for orders that are ready to be shipped. Click **Back to Orders page** .

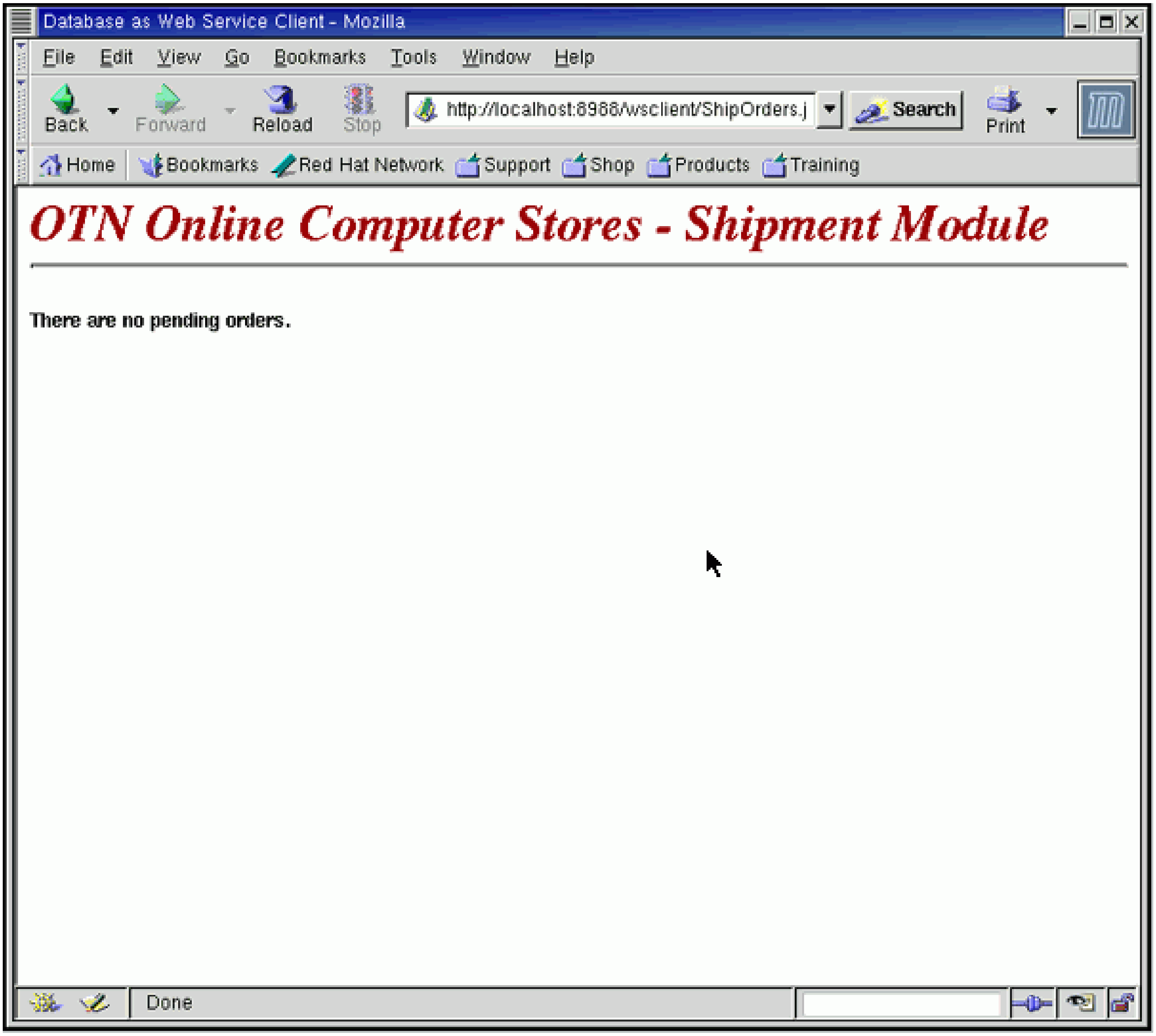

Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:8988/wsclient/ShipOrders.j
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
Back to Orders Page

OTN Online Computer Stores - Shipment Module

Your request has been processed. Here is the current status of the orders

Customer ID	Order ID	Order Status	Product Name	Quantity Shipped
otn-user	1	IN_SHIPPING	Compact 400/DQ	1
			Compact 400/LQ	1
			Inkjet B/6	1

☐ **10.** You see that you have no more orders pending. Based on the amount of purchase and the credit balance of the customer, the status of the order will be updated when the batch job is run next time. Remember that the batch job is designed to run every six minutes.


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:8988/wsclient/ShipOrders.j
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
Done

OTN Online Computer Stores - Shipment Module

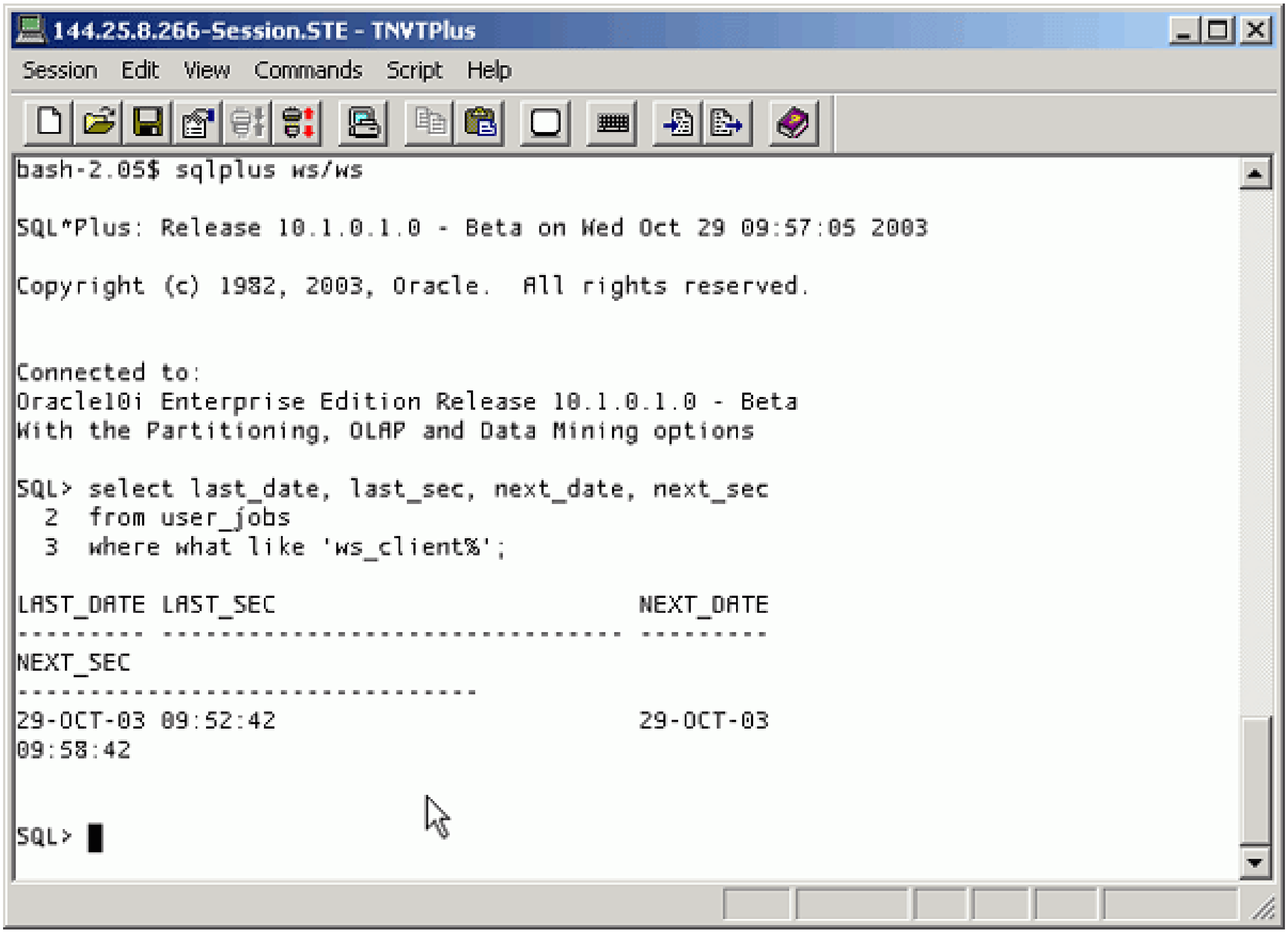
There are no pending orders.

☐ **11.** Open a **SQL*Plus** session and execute the following command:

```
connect ws/ws
```

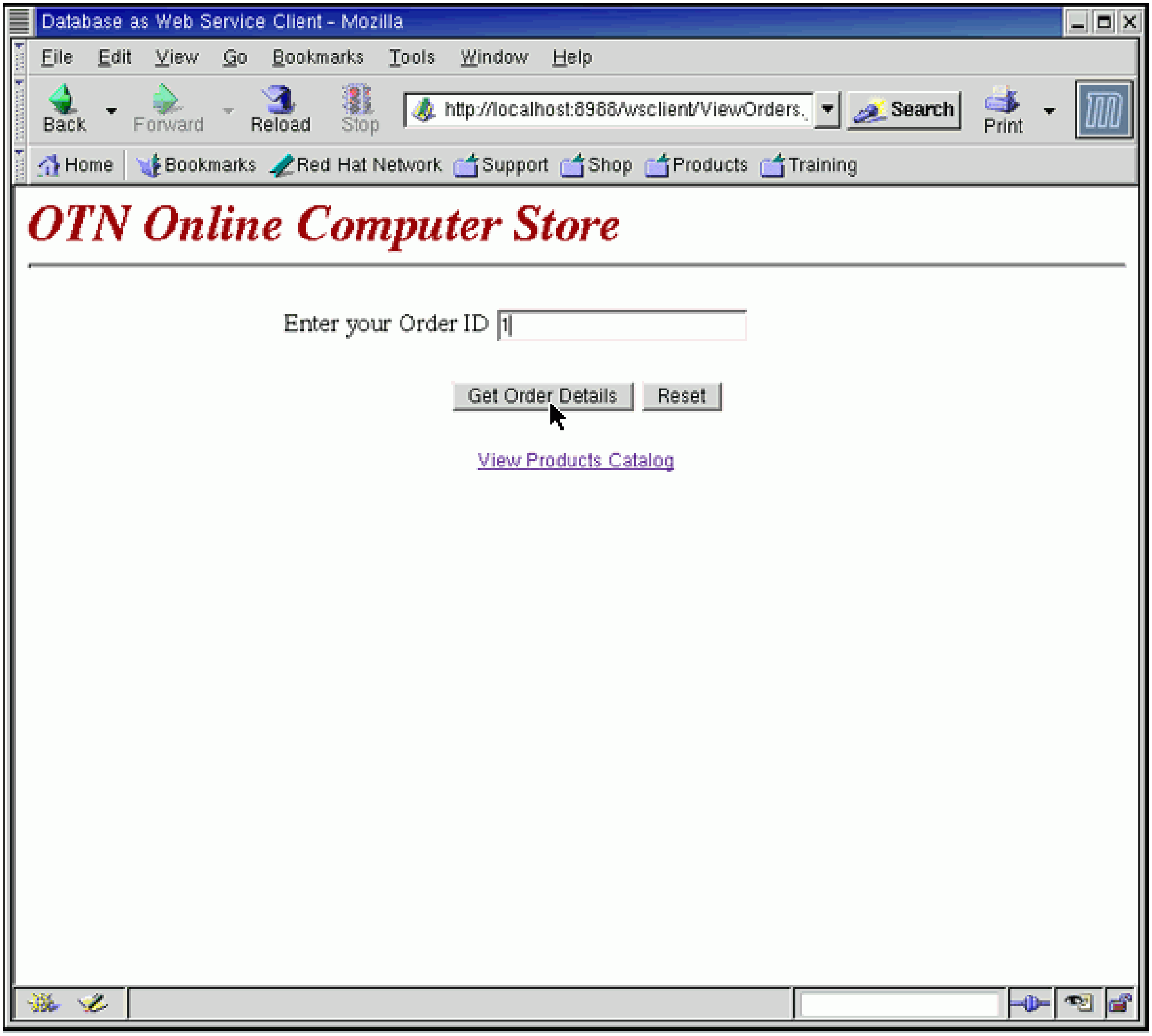
Check when the job will run next by issuing the following query:

```
SELECT last_date, last_sec, next_date, next_sec
FROM user_jobs
WHERE what like 'ws_client%';
```


144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help

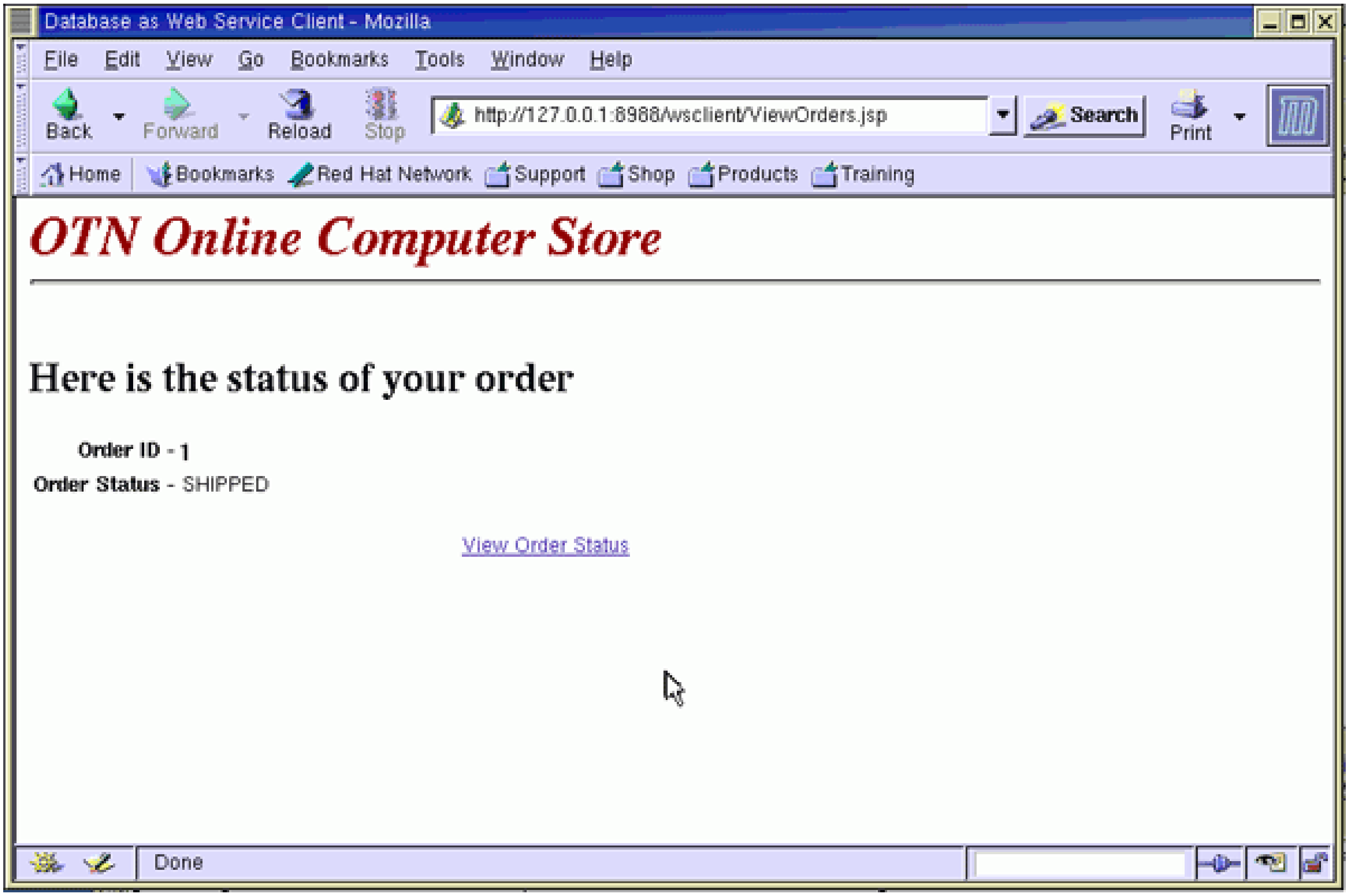
```
bash-2.05$ sqlplus ws/ws

SQL*Plus: Release 10.1.0.1.0 - Beta on Wed Oct 29 09:57:05 2003

Copyright (c) 1982, 2003, Oracle.  All rights reserved.


Connected to:
Oracle10i Enterprise Edition Release 10.1.0.1.0 - Beta
With the Partitioning, OLAP and Data Mining options

SQL> select last_date, last_sec, next_date, next_sec
  2  from user_jobs
  3  where what like 'ws_client%';

LAST_DATE LAST_SEC                         NEXT_DATE
--------- -------------------------------- ---------
NEXT_SEC
--------------------------------
29-OCT-03 09:52:42                         29-OCT-03
09:58:42


SQL>
```

☐ **12.** Switch back to your browser and check the order status. Replace ShipOrders.jsp in the location bar with **ViewOrders.jsp** . Enter **1** for the Order ID and click **Get Order Details** .


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:8988/wsclient/ViewOrders.
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
OTN Online Computer Store
Get Order Details
Reset
View Products Catalog

Enter your Order ID 1

☐ 13. Observe that the status has changed from IN_SHIPPING to **SHIPPED** . If it has not, the batch job has not run yet. Try again in a few minutes until the status does change.


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://127.0.0.1:8988/wsclient/ViewOrders.jsp
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
Done

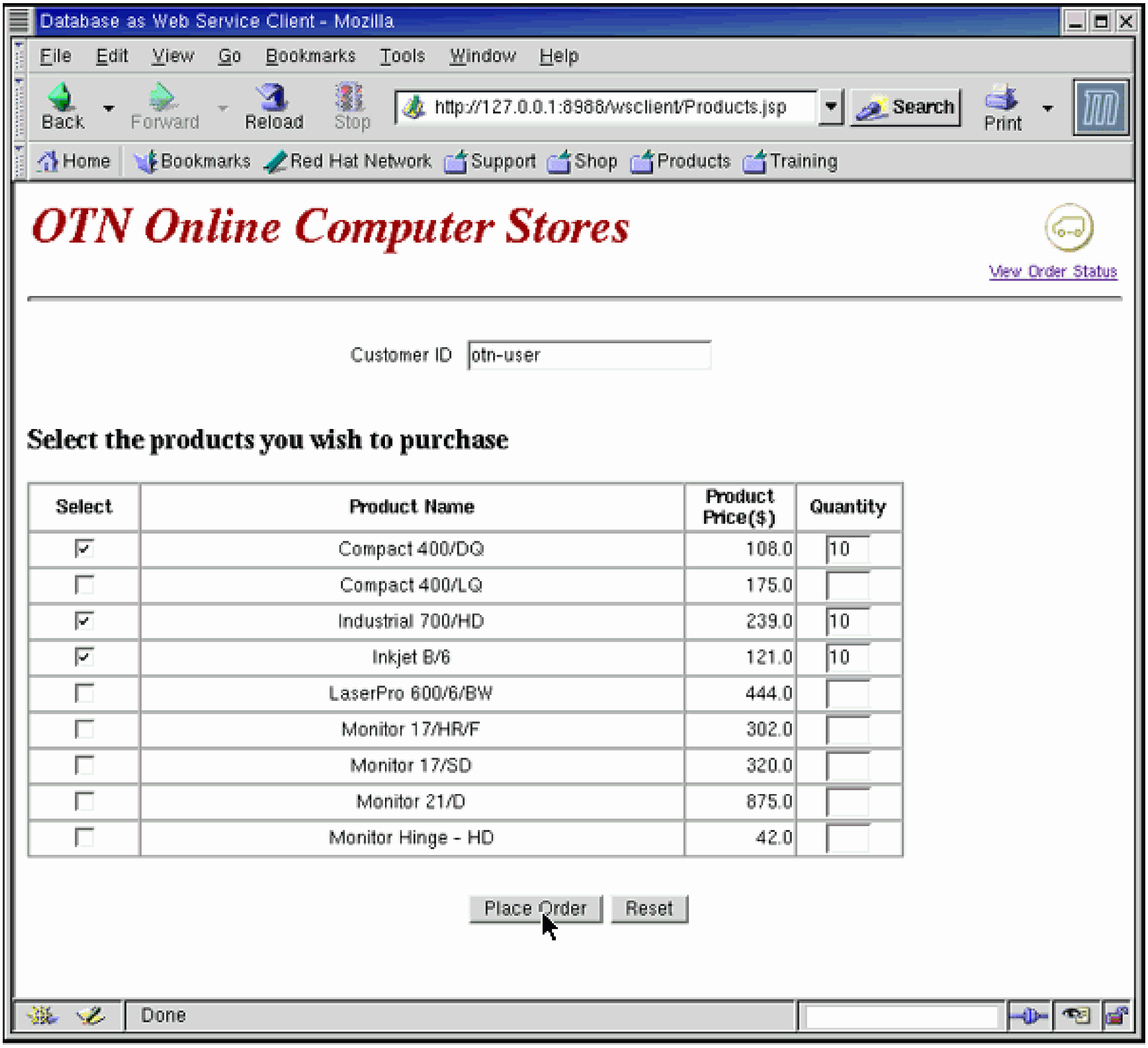
OTN Online Computer Store

Here is the status of your order

Order ID - 1
Order Status - SHIPPED

View Order Status

☐ **14.** Now you will place another order such that the total amount of the order exceeds the credit limit. Change the URL to **`http://localhost:8988/wsclient/Products.jsp.`** Enter **otn-user** for Customer ID and choose three of the products and enter a quantity of **10** for each. Then click **Place Order** .


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
http://127.0.0.1:8988/wsclient/Products.jsp
Search
Print
Home
Bookmarks
Red Hat Network
Support
Shop
Products
Training
View Order Status
Place Order
Reset
Done

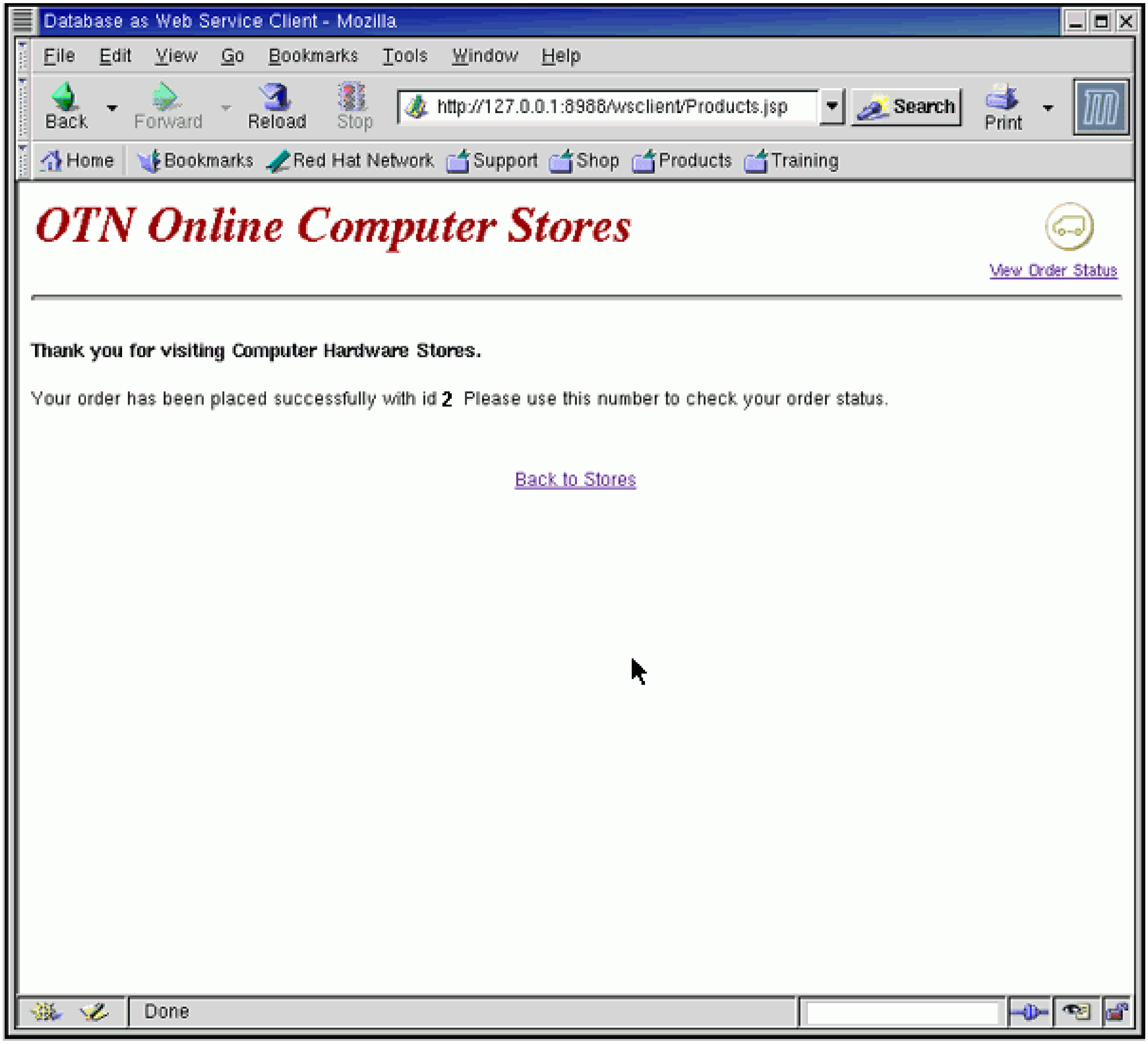
OTN Online Computer Stores

Customer ID otn-user

Select the products you wish to purchase

Select	Product Name	Product Price($)	Quantity
☑	Compact 400/DQ	108.0	10
☐	Compact 400/LQ	175.0	
☑	Industrial 700/HD	239.0	10
☑	Inkjet B/6	121.0	10
☐	LaserPro 600/6/BW	444.0	
☐	Monitor 17/HR/F	302.0	
☐	Monitor 17/SD	320.0	
☐	Monitor 21/D	875.0	
☐	Monitor Hinge - HD	42.0	

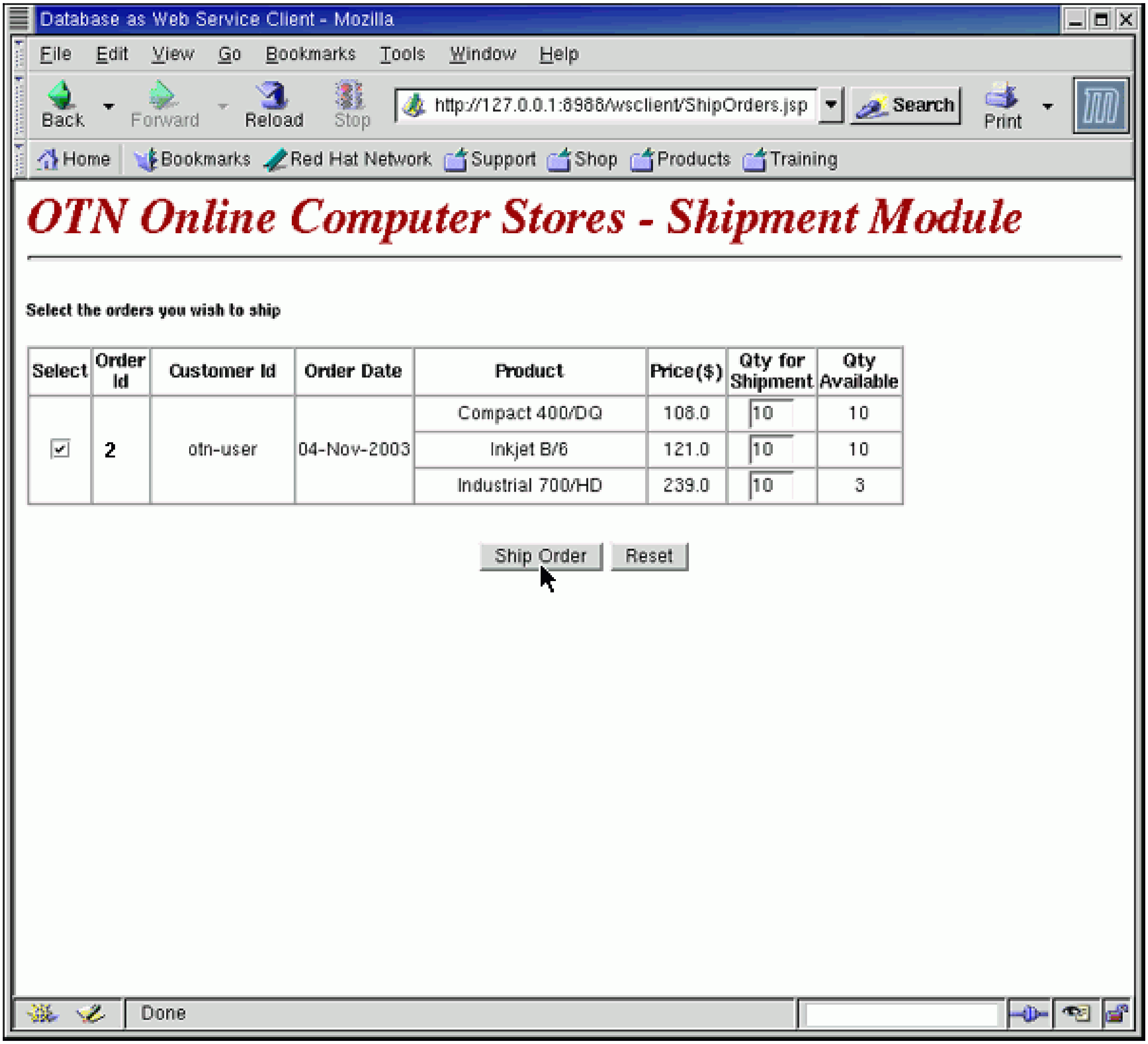
☐ **15.** Your order has been placed. Change the URL to **`http://localhost:8988/wsclient/ShipOrders.jsp`** .


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://127.0.0.1:8988/wsclient/Products.jsp
Search Print
Home Bookmarks Red Hat Network Support Shop Products Training
OTN Online Computer Stores
View Order Status
Back to Stores
Done

Thank you for visiting Computer Hardware Stores.

Your order has been placed successfully with id **2** Please use this number to check your order status.

☐ **16.** Select your order and click **Ship Order.**

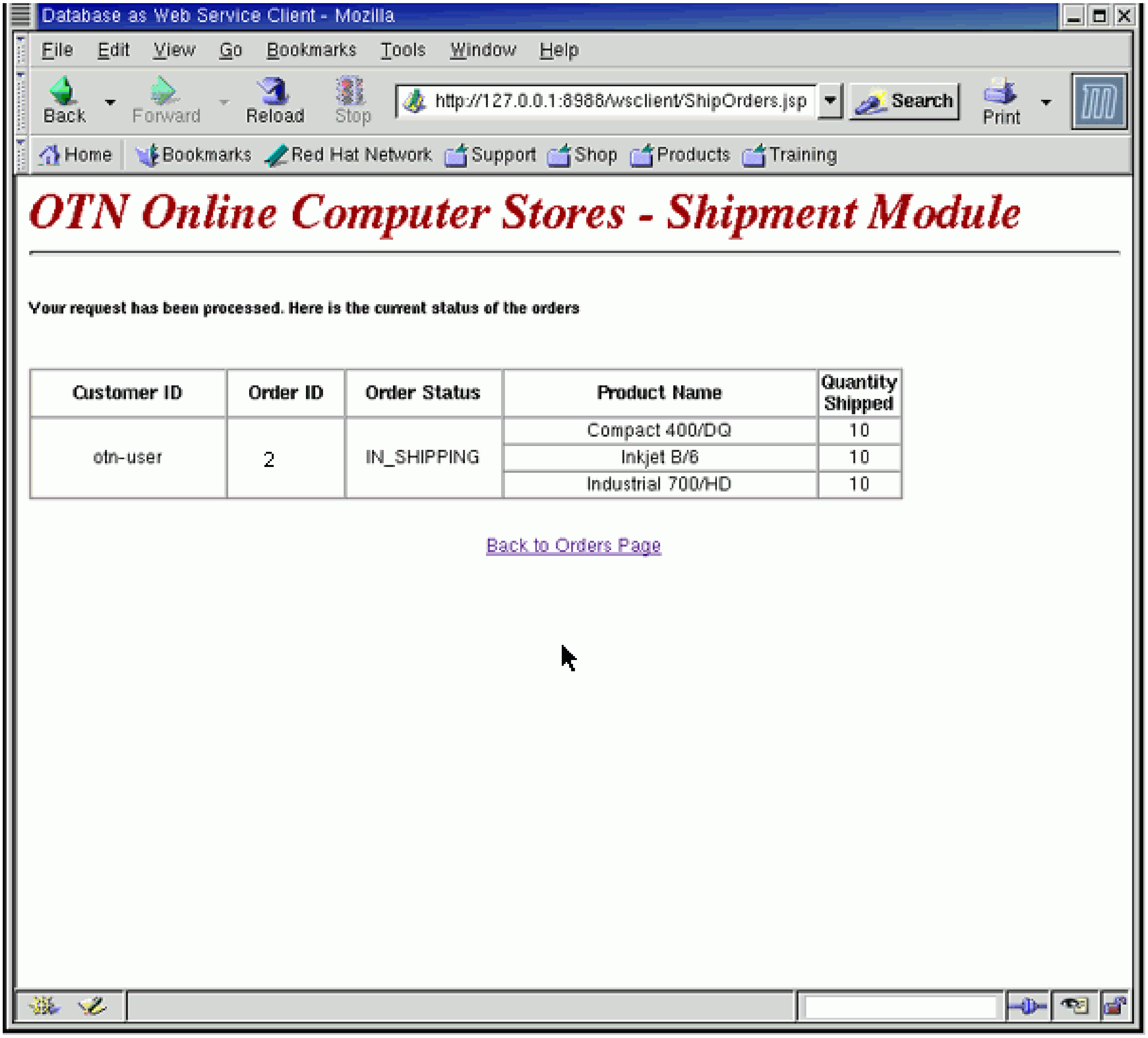

Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://127.0.0.1:8988/wsclient/ShipOrders.jsp
Search Print
Home Bookmarks Red Hat Network Support Shop Products Training
Ship Order
Reset
Done

OTN Online Computer Stores - Shipment Module

Select the orders you wish to ship

Select	Order Id	Customer Id	Order Date	Product	Price($)	Qty for Shipment	Qty Available
☑	2	otn-user	04-Nov-2003	Compact 400/DQ	108.0	10	10
				Inkjet B/6	121.0	10	10
				Industrial 700/HD	239.0	10	3

☐ **17.** Your order was shipped. You need to wait for the batch job to execute again.


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://127.0.0.1:8988/wsclient/ShipOrders.jsp
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training

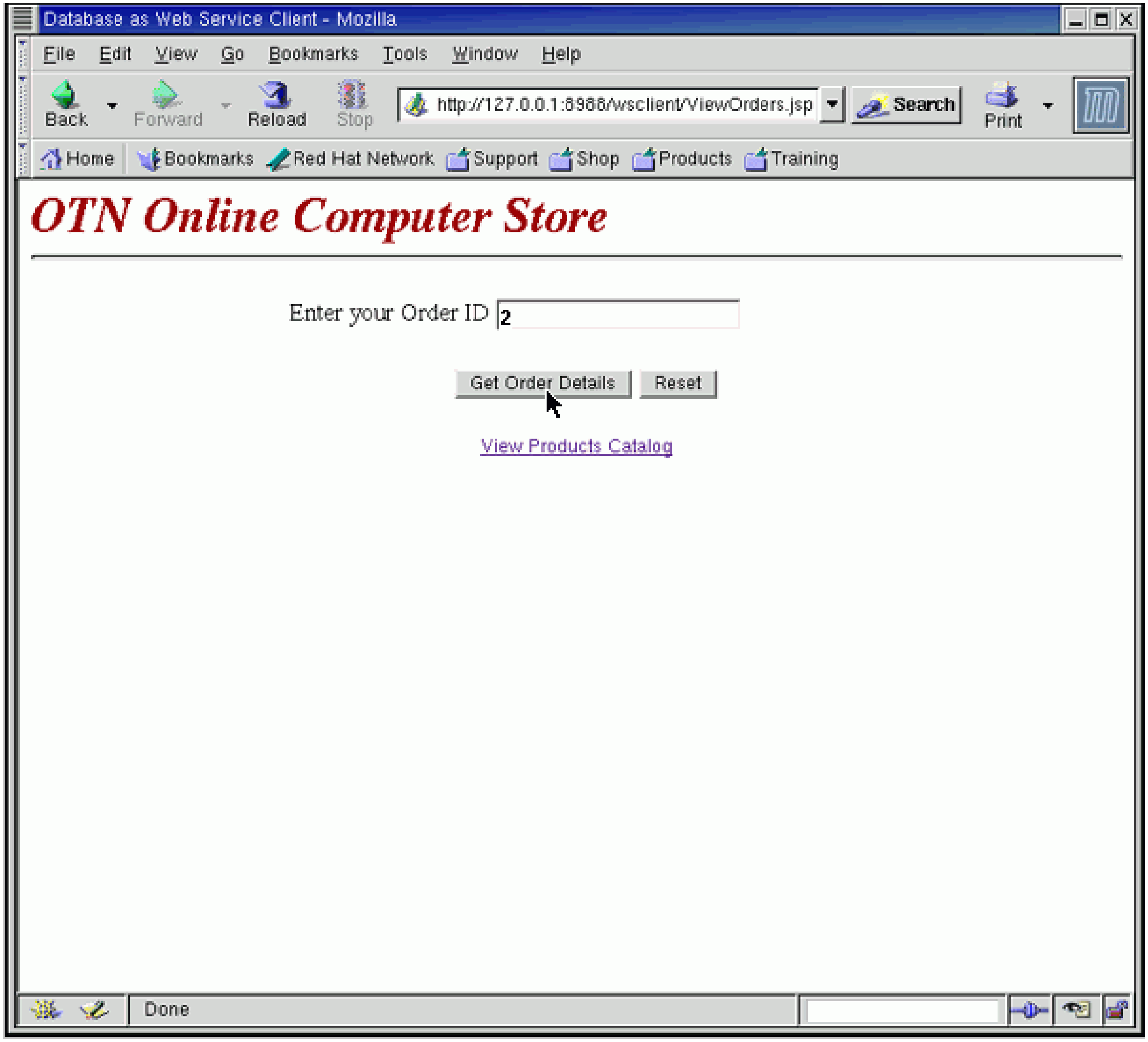
OTN Online Computer Stores - Shipment Module

Your request has been processed. Here is the current status of the orders

Customer ID	Order ID	Order Status	Product Name	Quantity Shipped
otn-user	2	IN_SHIPPING	Compact 400/DQ	10
			Inkjet B/6	10
			Industrial 700/HD	10

Back to Orders Page

☐ **18.** Change the URL to **`http://localhost:8988/wsclient/ViewOrders.jsp`** . Enter **2** for Order ID and click **Get Order Details** .

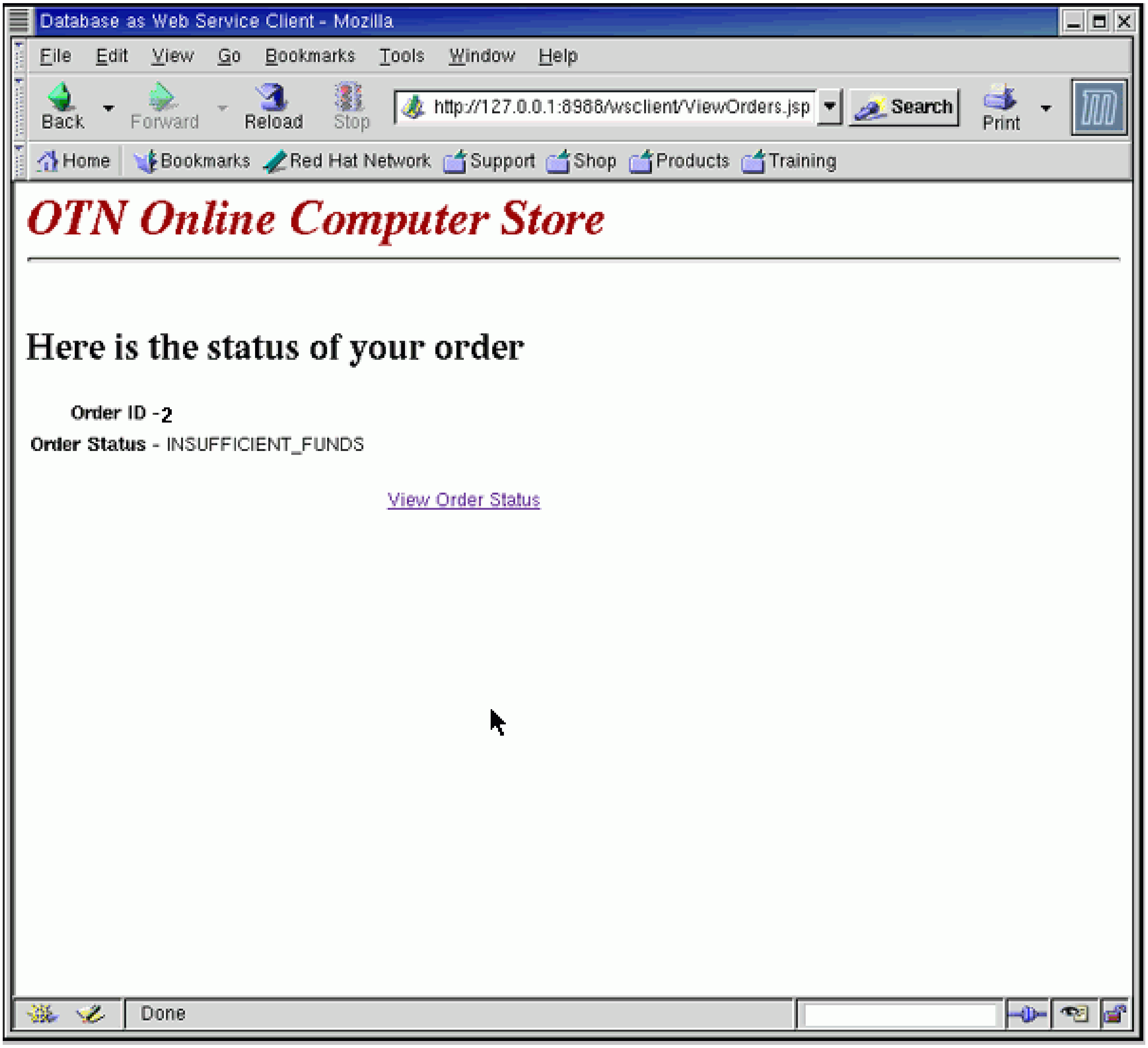


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://127.0.0.1:8988/wsclient/ViewOrders.jsp
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
Get Order Details
Reset
View Products Catalog
Done

OTN Online Computer Store

Enter your Order ID 2

☐ **19.** You see that there is insufficient funds because the credit limit was exceeded.


Database as Web Service Client - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://127.0.0.1:8988/wsclient/ViewOrders.jsp
Search Print
Home Bookmarks Red Hat Network Support Shop Products Training
OTN Online Computer Store
View Order Status
Done

Here is the status of your order

Order ID - 2

Order Status - INSUFFICIENT_FUNDS