USING TEIID DESIGNER TO CONSUME A REST-BASED WEB SERVICE AS A RELATIONAL MODEL

Blaine Mincey
08/16/2012

The purpose of this paper is to provide an introduction on how to consume a REST-based Web Service as a relational data source using the Teiid Designer which is part of the JBoss Developer Studio (JBDS) Eclipse-based IDE. The Teiid Designer is a visual tool that enables rapid, model-driven definition, integration, management, and testing of data services. This tool is part of the Enterprise Data Services Platform (EDSP) product. EDSP is a powerful set of tools and runtime components that makes it easy for your applications and business processes to integrate and use data from many data sources.

CONTENTS

OVERVIEW 2

ENVIRONMENT 2

EXAMPLE REST-BASED WEB SERVICE 3

SAMPLE XML 4

CREATE NEW MODEL PROJECT 5

PHYSICAL DATA SOURCE 8

XML DATA FILE IMPORT OPTIONS 15

PREVIEW DATA 18

RESOURCES 20

QUESTIONS/COMMENTS/ISSUES 20

OVERVIEW

This technical brief will guide you through the steps required to build a Teiid project that will consume a REST-based web service as a relational data model. This relational model can then be used in conjunction with other virtualized relational models to further extend your data abstraction layer. Specifically, the following steps will be performed:

1. Create and deploy a RESTful service that can be invoked by the Enterprise Data Services Platform
2. Create a project in JBoss Developer Studio using the Teiid Designer
3. Model the physical structure of the RESTful service
4. Connect to the RESTful service through JBoss Developer Studio using a Connection Profile
5. Create the virtual abstraction layer for the data returned by the RESTful service
6. Transform the data into the virtual abstraction layer
7. Test the newly created virtual relational model

ENVIRONMENT

The following environment was utilized in order to produce the required artifacts for this example:

Operating System: RHEL 6.1 (Santiago)

JVM: Sun/Oracle 1.6.0_25 64-bit
JBoss Developer Studio 5

Enterprise Data Services 5.3

> **NOTE**: It should be noted that this specific environment is not required. There are JBDS binaries for not only RHEL, but Windows and Mac-OSX platforms as well. Additionally, the JBoss Enterprise Data Services Platform is supported with a compatible JVM so it is possible to run it on a variety of operating systems including RHEL, Windows, and Solaris.

There is an example REST-based Web Service available within a public repository on GitHub (http://www.github.com) that was utilized as the data source for this paper. Instructions will be provided later in the paper on how to acquire the source artifacts from GitHub as well as how to build and deploy the application. The following tools were used for the REST-based Web Service.

Git: 1.7.1

Maven: 3.0.3

EXAMPLE REST-BASED WEB SERVICE

In order to provide an easy way to get started with this example, a sample REST-based Web Service is available for download. No knowledge of REST or web services is required to build and deploy the application. A general understanding of both Git and Maven is required. Additional information on Git is available at http://git-scm.com/. This link provides information on how to setup Git for your target environment. Additionally, Maven information can be found at http://maven.apache.org/. This link provides a tutorial as well as the ability to download the Maven binaries. If this is your first time using Maven, you will need to point your Maven settings to the JBoss Maven repositories in order to access JBoss artifacts. The following article describes how to do this: http://community.jboss.org/wiki/MavenGettingStarted-Users.

The public GitHub repository for the example REST-based Web Service is at

https://github.com/blainemincey/CustomerRESTWebSvc.

In order to download the repository to your local workstation, execute the following command from a command or terminal window:

```
git clone git://github.com/blainemincey/CustomerRESTWebSvc.git
```

If git has been configured correctly on your system, this command will copy the public repository to your local workstation. After the command is completed, there should be a new directory, CustomerRESTWebSvc. Now, 'cd' (change directory) into this newly created directory. There should be a pom.xml file, a README file, and a src directory.

At this point, if Maven has been correctly installed on your system, you can simply execute the following Maven command in order to build the WAR file that will be deployed to the Data Services Platform.

```
mvn clean compile war:war
```

This command will compile the source files as well as build a deployable WAR file. This command will create a 'target' directory. This newly created directory will contain the CustomerRESTWebSvc.war file. This file can simply be copied to the running configuration's 'deploy' directory of the Enterprise Data Services Platform. In order to manually test the web service to ensure it is working correctly, open a browser to the following location:

http://localhost:8080/CustomerRESTWebSvc/MyRESTApplication/customerList

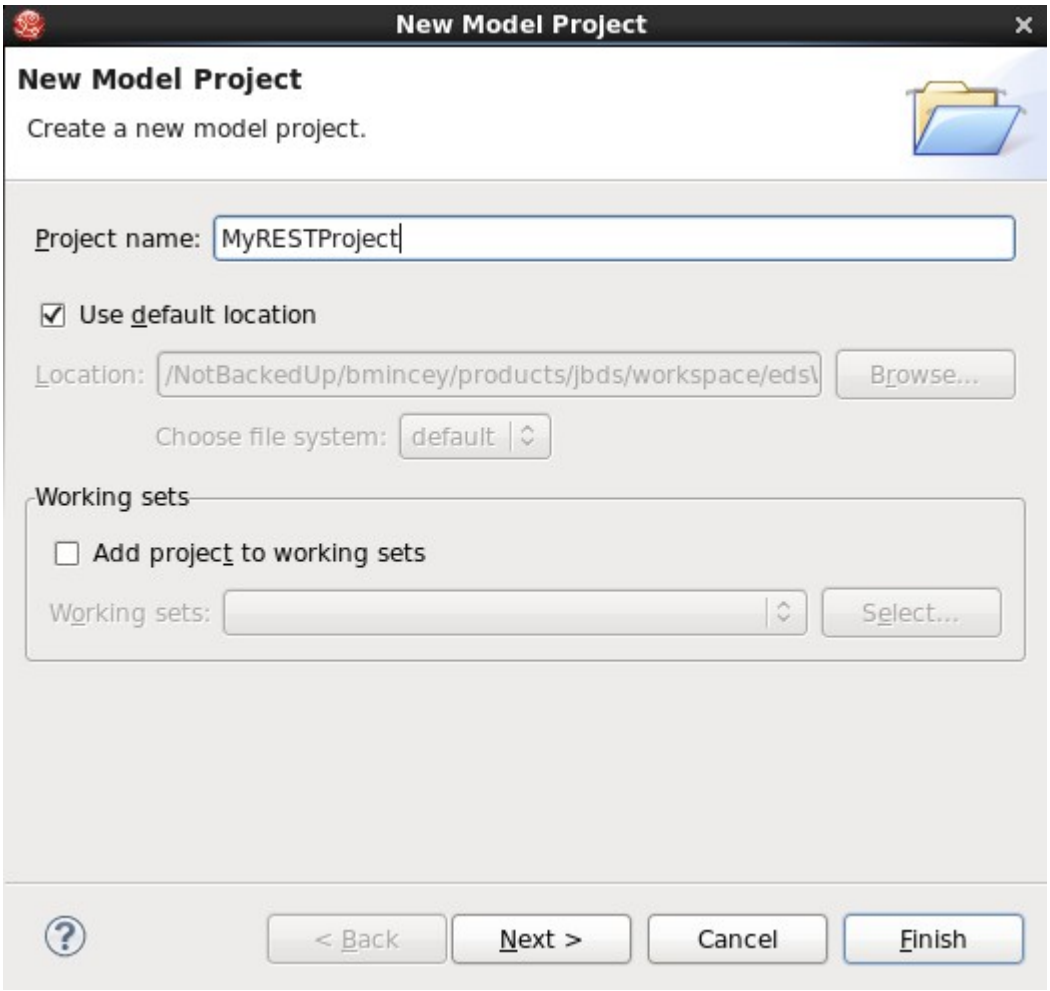
SAMPLE XML

The following XML snippet serves as an example of the literal XML result from the REST-based Web Service that was deployed in the previous step. This is simply a single record from the web service. It should be noted that well over 100 records will be returned upon executing the REST Web Service.

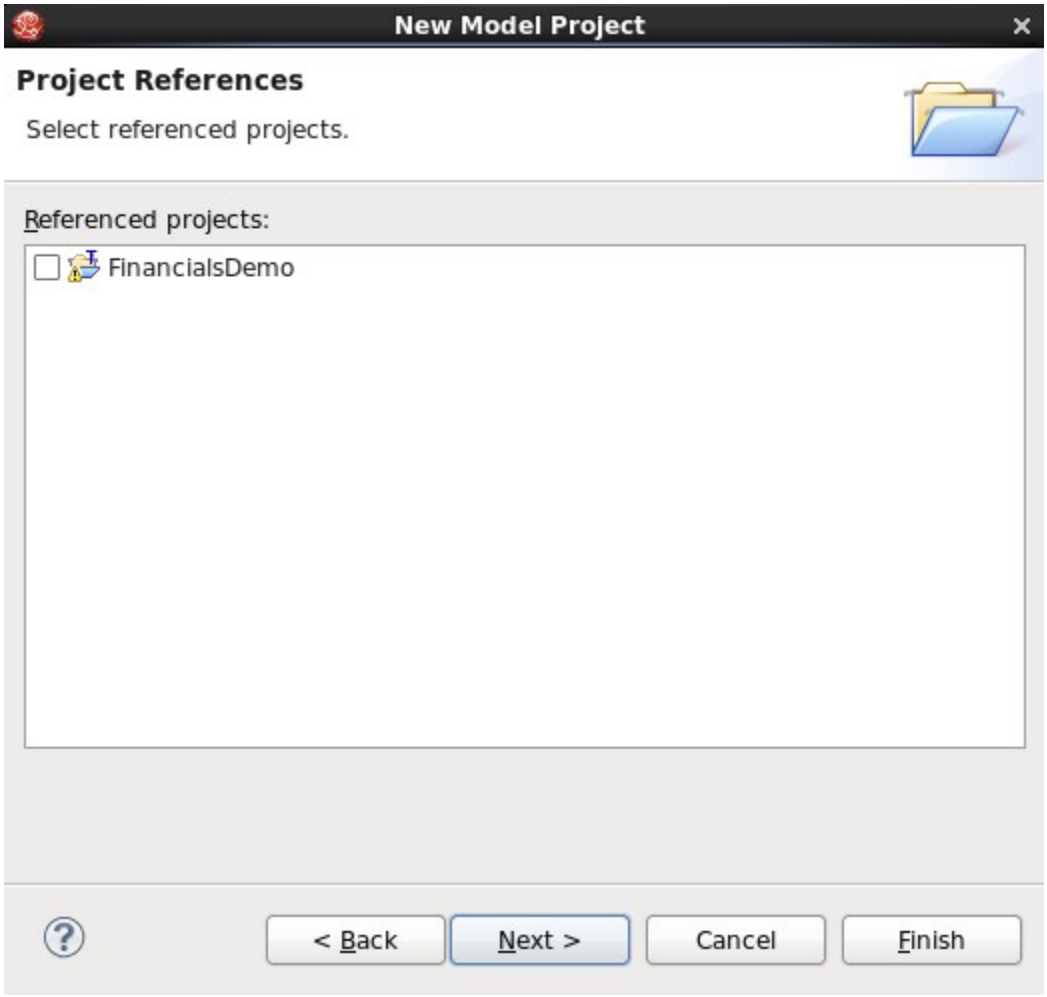
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<customers>
      <customer>
            <customernumber>103</customernumber>
            <customername>Atelier graphique</customername>
            <contactlastname>Schmitt</contactlastname>
            <contactfirstname>Blaine</contactfirstname>
            <phone>40.32.2555</phone>
            <addressline1>54, rue Royale</addressline1>
            <addressline2 />
            <city>Nantes</city>
            <state />
            <postalcode>44000</postalcode>
            <country>France</country>
            <salesrepemployeenumber>1370</salesrepemployeenumber>
            <creditlimit>21000.0</creditlimit>
      </customer>
</customers>
```

CREATE NEW MODEL PROJECT

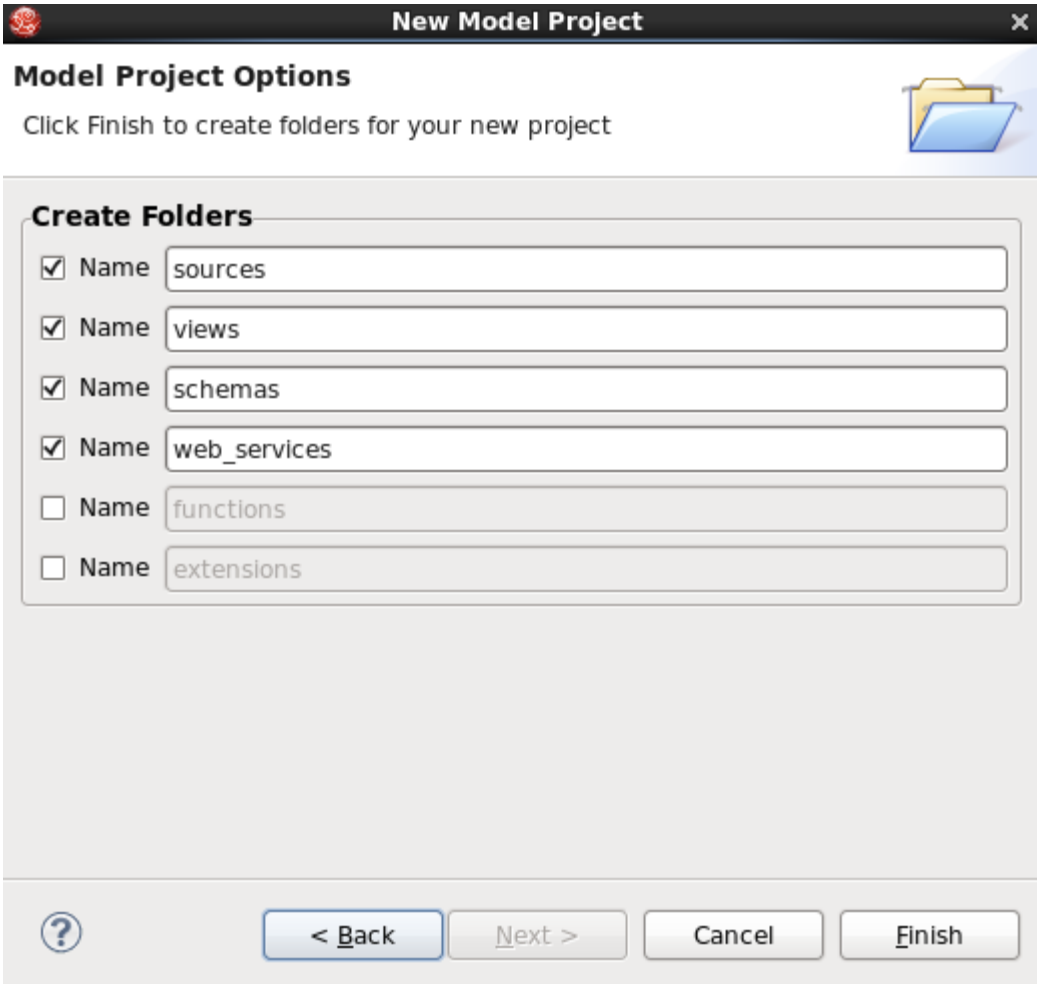
With JBDS opened in the Teiid Designer perspective, right-click within the Model Explorer and select “New” and then “Teiid Model Project”.



Enter a “Project name”. For the purposes of this paper, the project name will be “MyRESTProject” has been chosen. Now, click the “Next” button.



This brings up the “Project References” window. You may or may not have any referenced projects. However, for this paper, we will not select any referenced projects. Now, click the “Next” button.

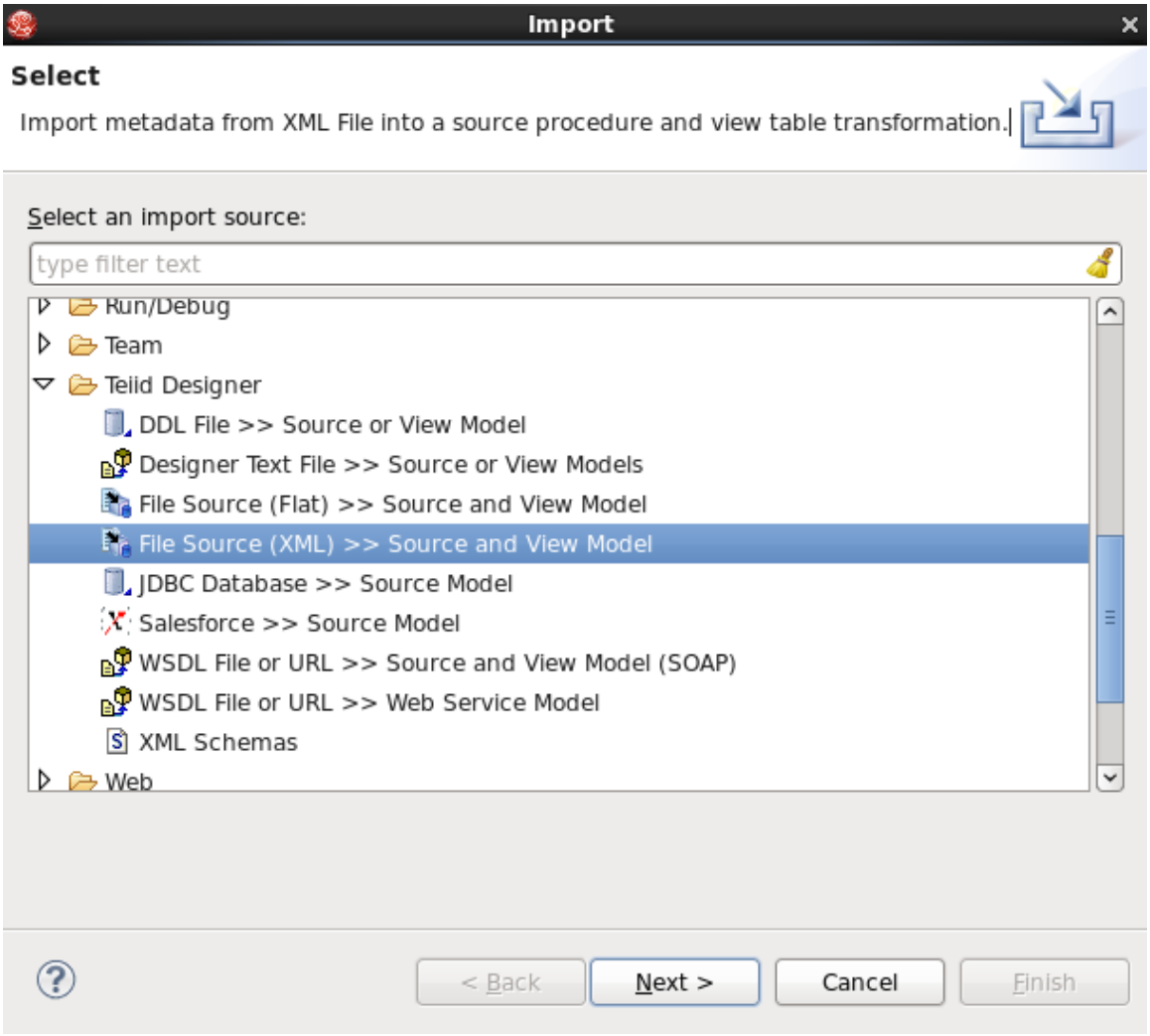


This brings up the “Model Project Options”. This option gives you the ability to create several folders for your project in order to stay organized and follow Teiid Designer best practices. In this instance, keep all of the default selections and click the “Finish” button.

PHYSICAL DATA SOURCE

The solution to many of today's data challenges is data virtualization. JBoss Enterprise Data Services Platform provides you with the ability to create a data abstraction, or data virtualization, layer on top of your physical data sources. It is then possible to create a variety of virtual data layers on top of your physical data sources and expose them as a single interface. For the purpose of this paper, you will be able to model a REST based web service and make it appear as a relational data source. From this point, it is possible to join the data source with other virtual models providing you with the means to turn the data you have into the information you need.

Now, right-click on the "sources" folder and select "Import". The following screen will pop-up.

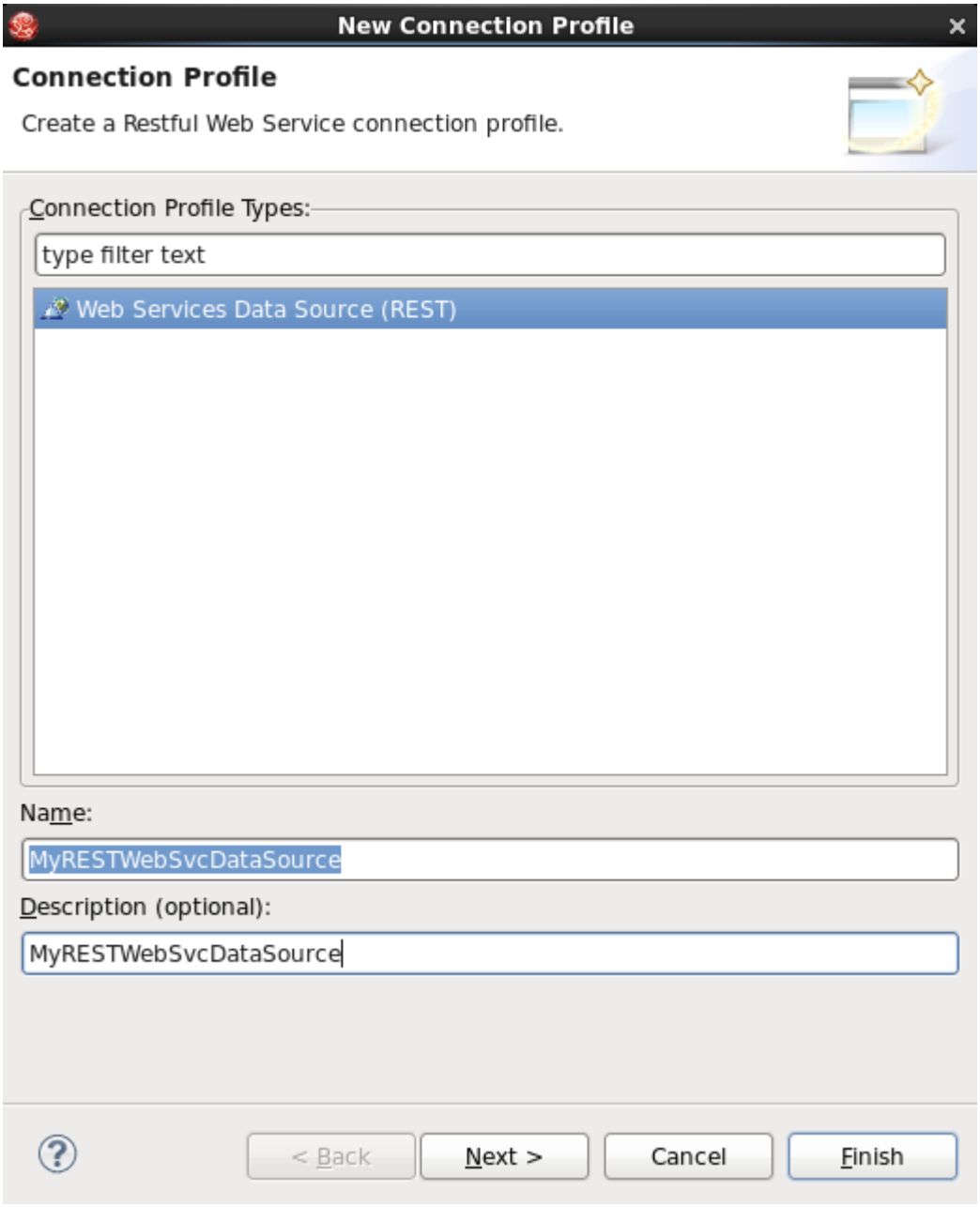


Be sure to select “File Source (XML) >> Source and View Model” and click the “Next” button.

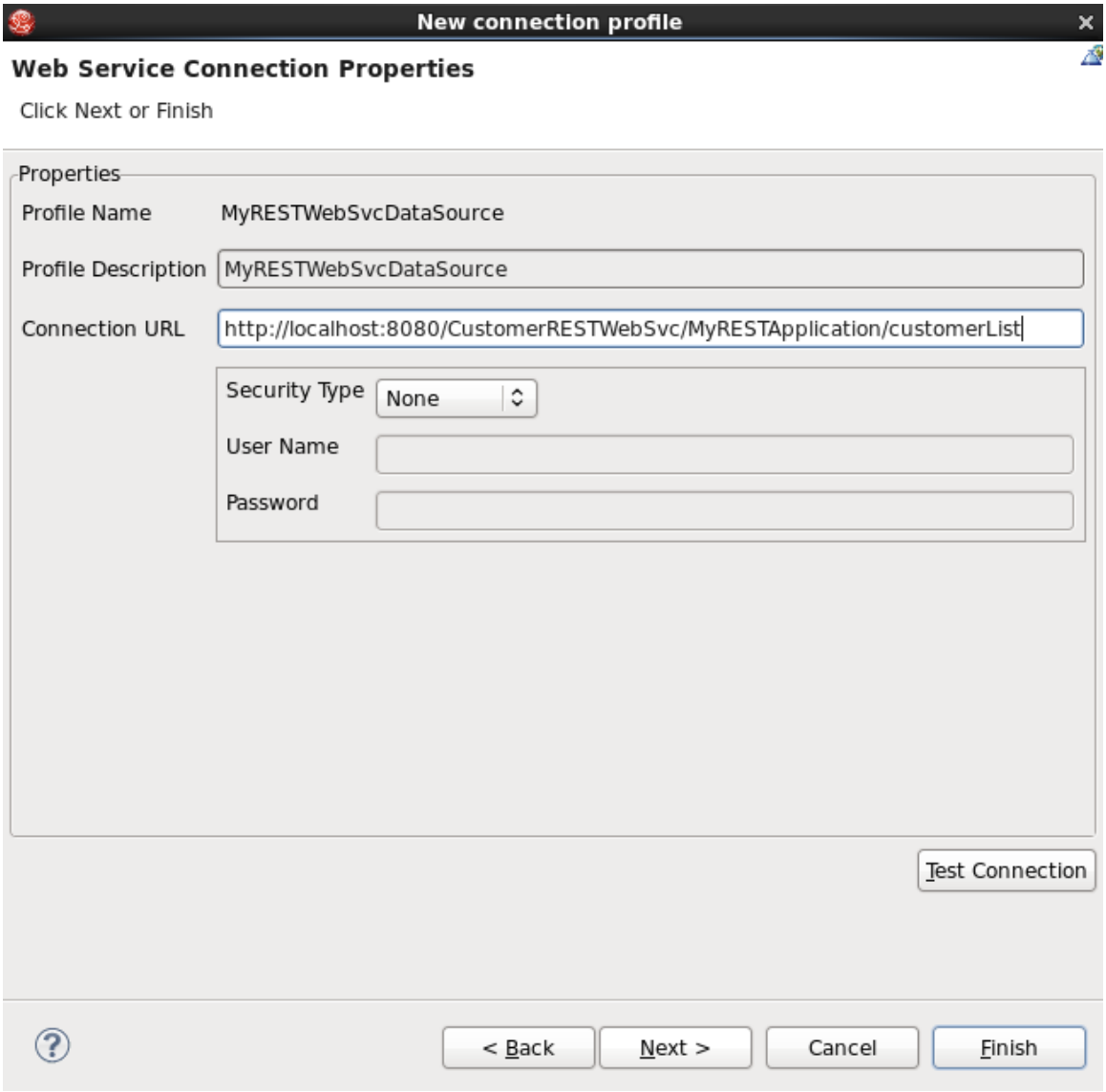
Now, the “XML Import File Options” window is made available. Be sure to select the “XML file via remote URL” radio button and click the “Next” button. This will open the “XML Data File Source Selection” window as indicated below.

To the right of “Data File Source”, click the “New” button in order to define our REST-based web service.

This brings up the “XML Connection Profile Options” window as indicated above. Be sure to select the “REST Web Services URL” and click the “OK” button. After clicking OK, the “Connection Profile” wizard will appear. Be sure to enter a name for the connection profile. As indicated in the image below, the name chosen is “MyRESTWebSvcDataSource”. Once your window looks like the one below, click the “Next” button.



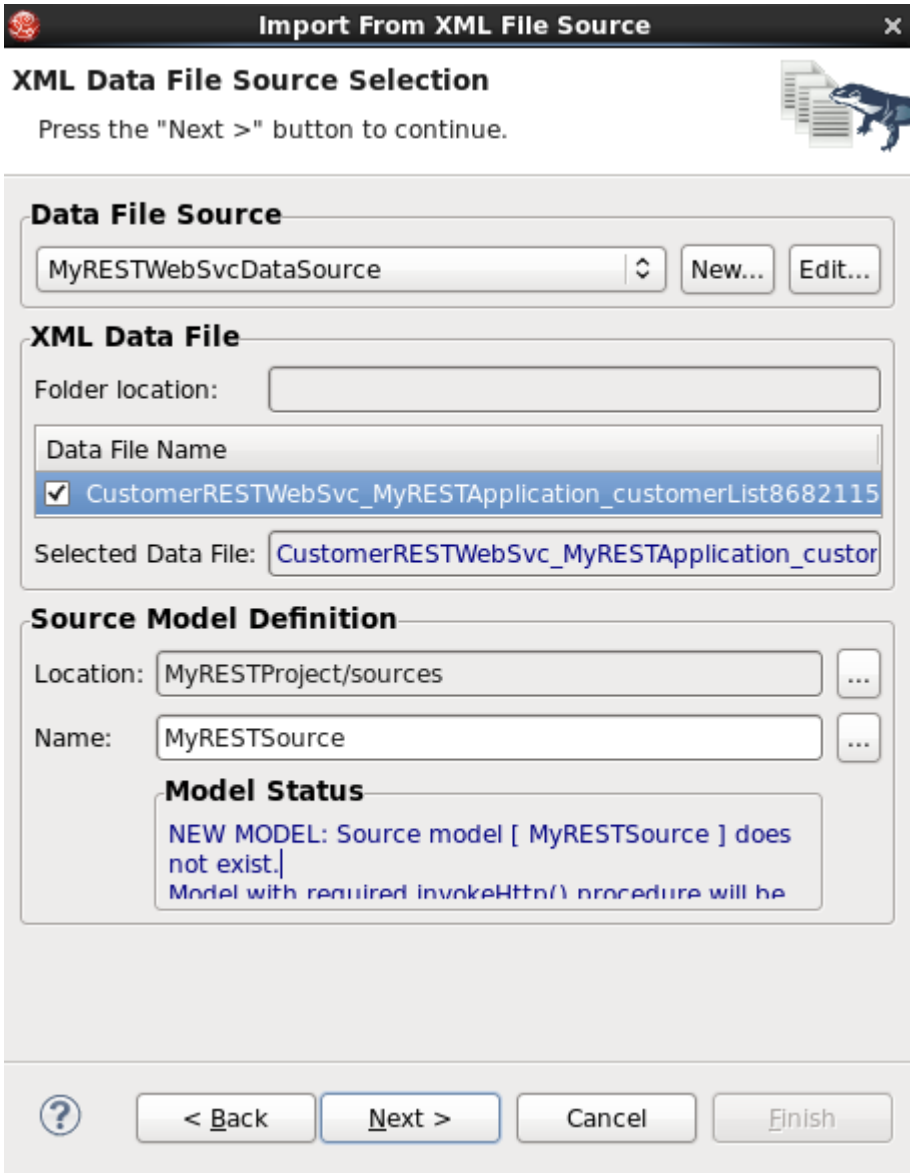
Once you click “Next”, the “Web Service Connection Properties” window pops-up. Enter in the connection URL of the REST Web Service that was deployed to your server earlier in this paper. The window should look like that below.



Again, the Connection URL should be http://localhost:8080/CustomerRESTWebSvc/MyRESTApplication/customerList. At this point, it is recommended to click the "Test Connection" button to ensure that everything is connected properly up to this point. If you get an error, go to a browser and enter in the connection URL and ensure something is returned and that your web service has been deployed correctly. Additionally, it may be required to repeat the steps in the paper in order to ensure the web service and connection profile are correct.

If you get a successful ping of the web service, click the “Finish” button.

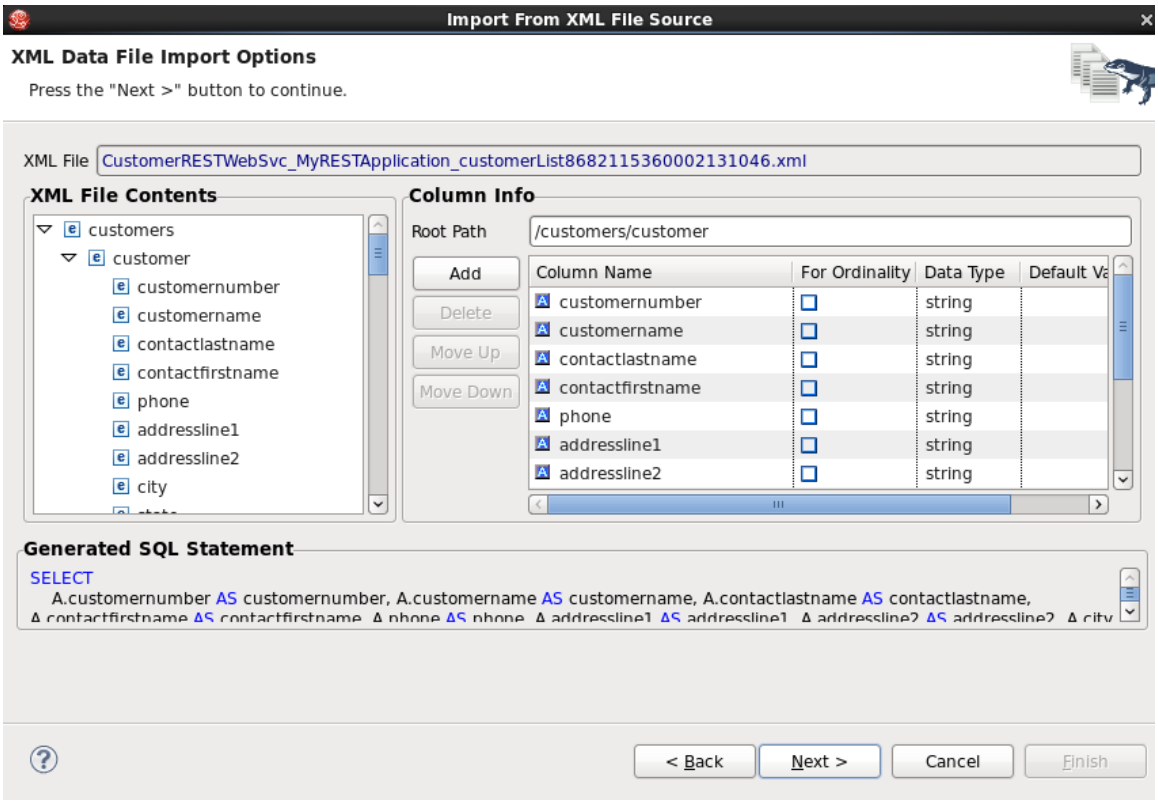
This will now bring you back to the “XML Data File Source Selection” window. Your Data File Source should have the entry that we just created. Be sure to enter a “Name” for the “Source Model Definition”. In this case, the name selected is “MyRESTSource”. Your window should resemble that as below.



Once your entry looks like that above, click the “Next” button.

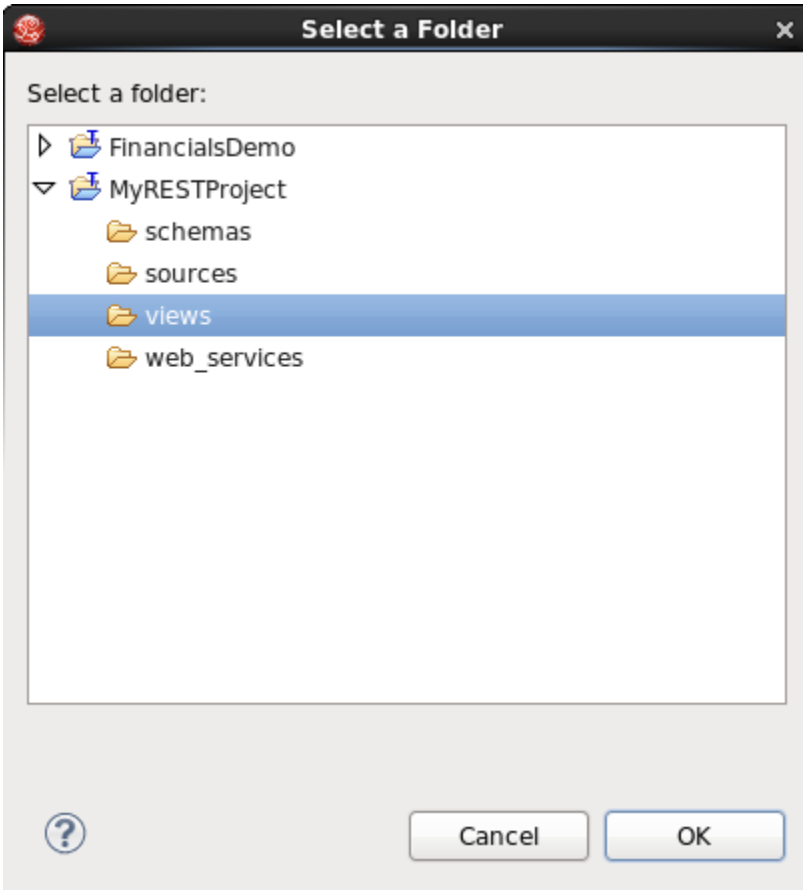
XML DATA FILE IMPORT OPTIONS

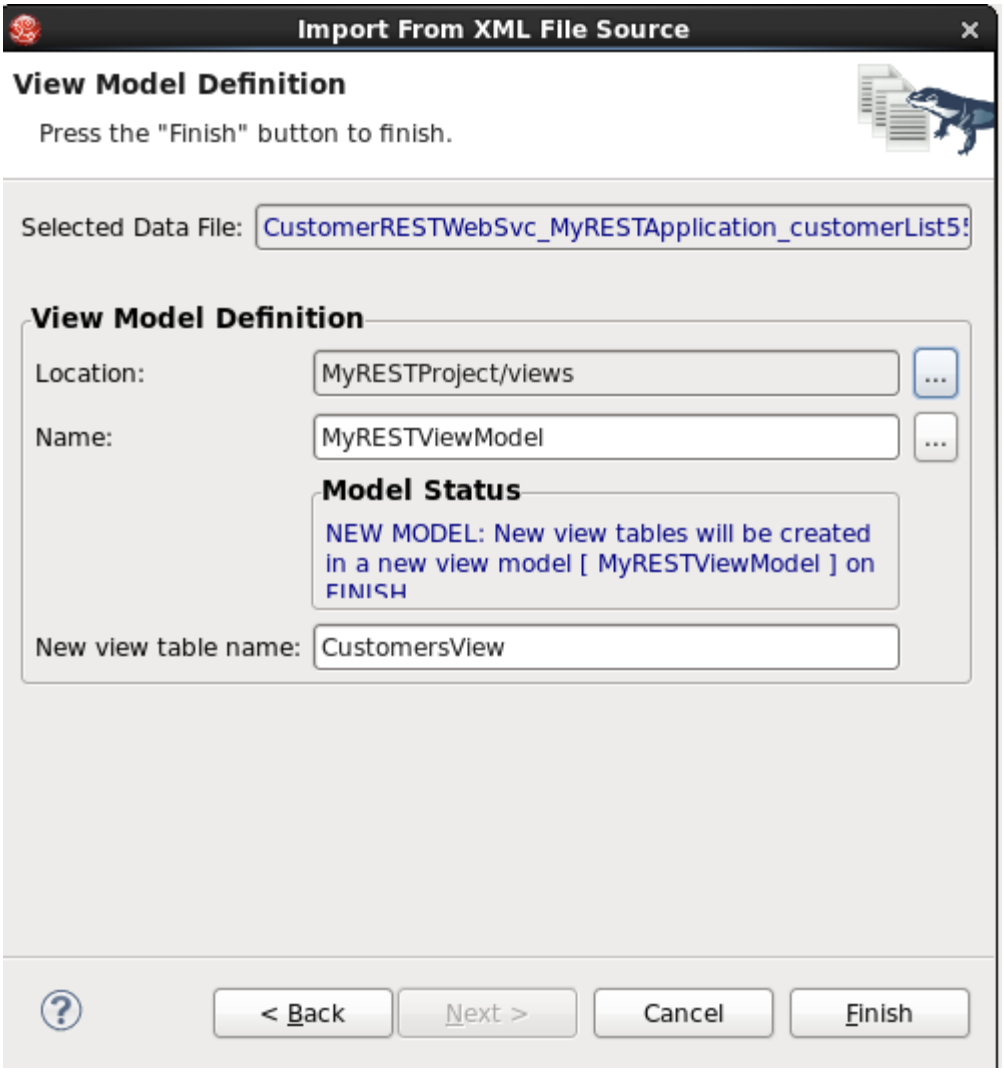
This will open the “XML Data File Import Options” window. Expand the “XML File Contents” where only “customers” is visible. Once expanded, right click on a customer element. Select “set as root path”. Now, expand the customer element. Double-click each child element. For example, there will be customernumber, customername, etc. For the purposes of this paper, all columns are selected. Your window should look like that below.



Now, click the “Next” button.

This will bring up the “View Model Definition” window. Add a name to the View Model as well as the “New View Table Name”. Also, be sure to select the “...” to the right of “Location” and select the “views” folder from our Teiid project. The selection should look like that below. Click OK.

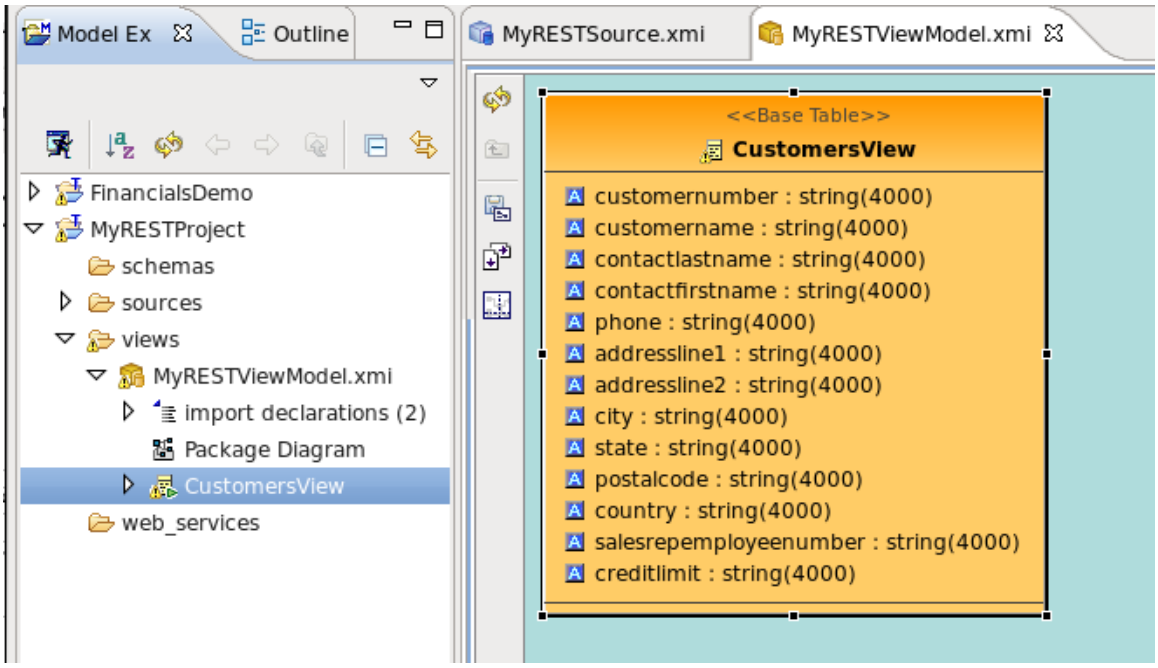

Select a Folder
Select a folder:
FinancialsDemo
MyRESTProject
schemas
sources
views
web_services
Cancel
OK



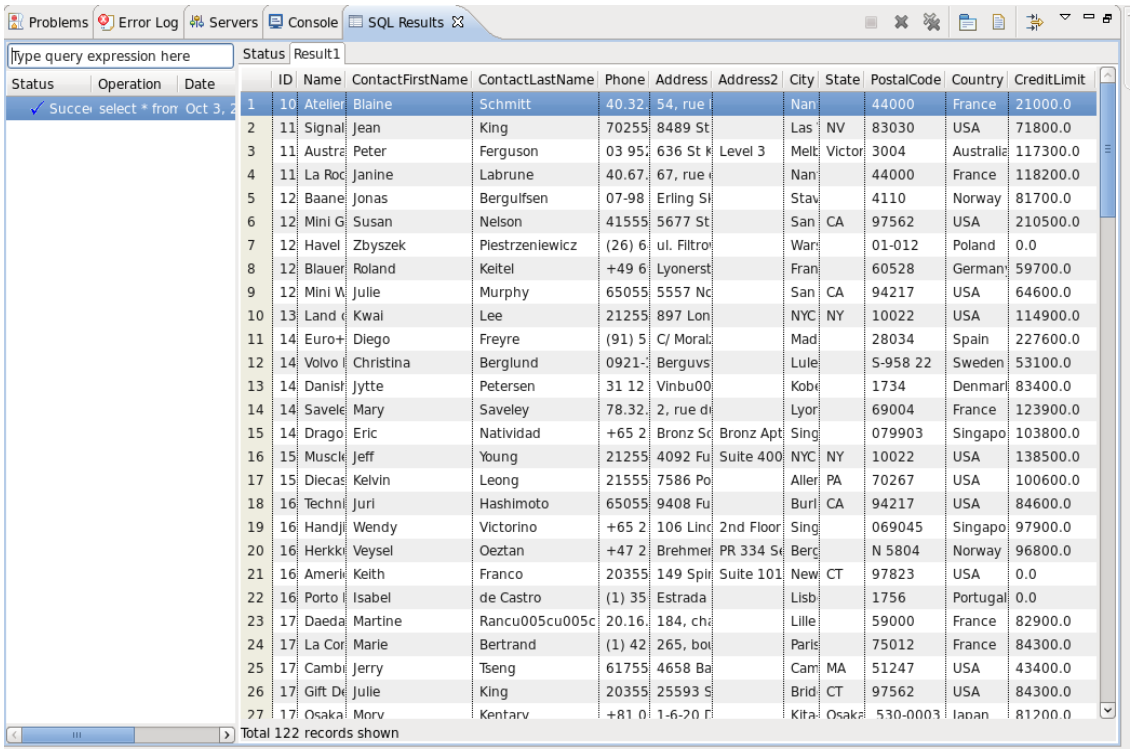
For this paper, the View Model name is “MyRESTViewModel” and the New view table name is “CustomersView”. Once your window resembles that above, click the “Finish” button. Once the model has been generated, be sure to select “Save All”.

PREVIEW DATA

Now, let's preview the data. Expand the views folder from the Teiid Model Explorer for our project. Expand the MyRESTViewModel.xmi entry. Select the CustomersView table. Once selecting this, the “running man” icon will become enabled along the top of the Model Explorer tab. If everything has been completed correcly, upon clicking this icon will produce a result in the “SQL Results” tab along the bottom of JBDS.



For illustrative purposes, the image below is the resulting preview of the data.



RESOURCES

JBoss Enterprise Data Services Documentation

http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Data_Services/index.html

QUESTIONS/COMMENTS/ISSUES

If you have questions or comments about this whitepaper, please enter them in the Red Hat customer portal for this specific whitepaper: https://access.redhat.com/knowledge/techbriefs . If you have a technical issue following this whitepaper please open a support case: https://access.redhat.com/support/cases/new.

Copyright © 2011 Red Hat, Inc. “Red Hat,” Red Hat Linux, the Red Hat “Shadowman” logo, and the products listed are trademarks of Red Hat, Inc., registered in the U.S. and other countries. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.