Lesson 1 Introducing the Data Services Platform Environment

BEA AquaLogic Data Services Platform provides the tools and components that let you build physical data services around individual physical data sources, and then develop the logical data services and business logic that integrate data from multiple physical and logical data services. The environment also lets you test the data service and manage data service metadata, caching, and security.

The basic menus, behavior, and look-and-feel associated with the WebLogic Workshop environment apply to DSP. However, there are several tools and components within WebLogic Workshop that are especially relevant to DSP. In this lesson, you will learn about a few of those tools and components. In addition, you will learn how to complete several basic tasks, such as starting and stopping WebLogic Server, that are essential to using WebLogic Workshop.

As the first lesson within the *AquaLogic Data Services Platform Samples Tutorial*, there are no dependencies on other lessons. However, your familiarity with WebLogic Workshop is assumed. Workshop is fully described in online documentation, which you can view at:

```
http://edocs.bea.com/workshop/docs81/index.html
```

Objectives

After completing this lesson, you will be able to:

- Navigate the DSP environment.
- Start and stop WebLogic Server.
- Save a Data Services application and associated files.

Overview

WebLogic Workshop consists of two parts: an Integrated Development Environment (IDE) and a standards-based runtime environment. The purpose of the IDE is to remove the complexity in building applications for the entire WebLogic platform. Applications you build in the IDE are constructed from high-level components rather than low-level API calls. Best practices and productivity are built into both the IDE and runtime.

Lab 1.1 Starting WebLogic Workshop

The first step is starting WebLogic Workshop and opening the RTLApp sample application, which you will use in the next lab.

Objectives

In this lab, you will:

- Start WebLogic Workshop.
- Open the RTLApp application.

Instructions

1. Choose Start → Programs → BEA WebLogic Platform 8.1 → WebLogic Workshop 8.1. If this is the first time you are starting WebLogic Workshop, then the SamplesApp project opens. Otherwise, the project that you last opened appears.
2. Choose File → Open → Application.

3. Open the RTLApp.work file from the following location:

```
<beahome> \weblogic81\samples\LiquidData\RTLApp\
```

Note: Depending on your computer settings, the .work extension may not be visible.

In Figure 1-1, the RTLApp application opens in Design View for the Case data service. If this is not the view that you see, double-click `Case.ds` located at `DataServices/ RTLServices` and select the Design View tab.

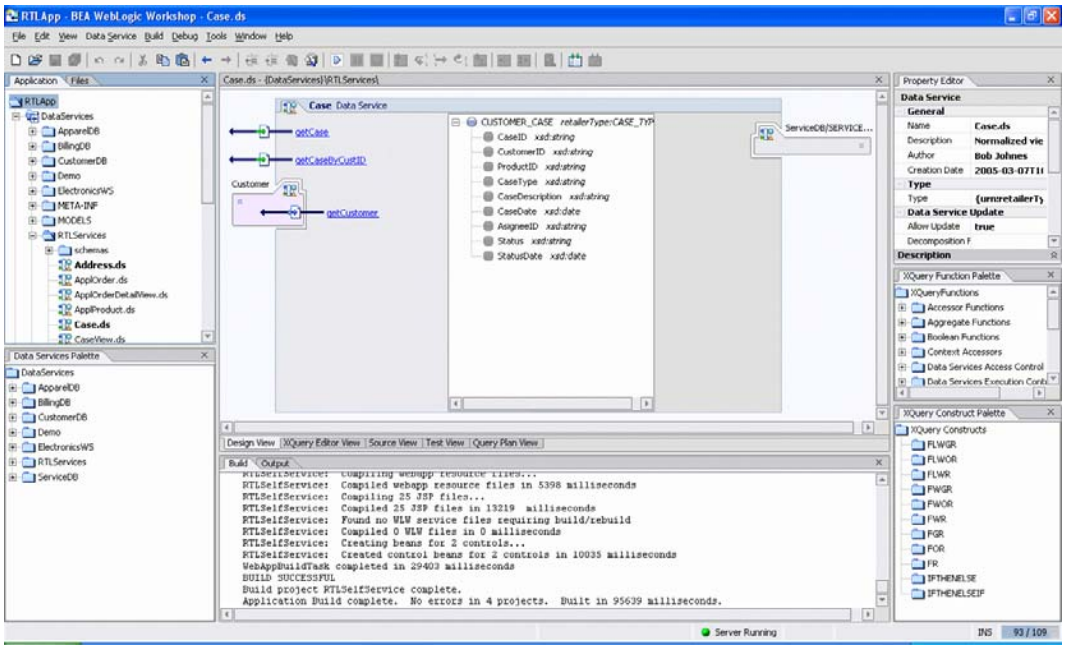


Figure 1-1 RTLApp in Design View for Case.ds

Note: The RTLApp application opens in the last active view. This action also resets the default WebLogic server home directory instance to the ldplatform sample domain.

Lab 1.2 Navigating the DSP Integrated Development Environment (IDE)

Within the WebLogic Workshop environment, there are several tools and components that are relevant to developing DSP applications and projects. Five of the most frequently used are:

- Application Pane
- Design View
- XQuery Editor View
- Source View
- Test View

Screenshots of the environment are taken from within the RTLApp application.

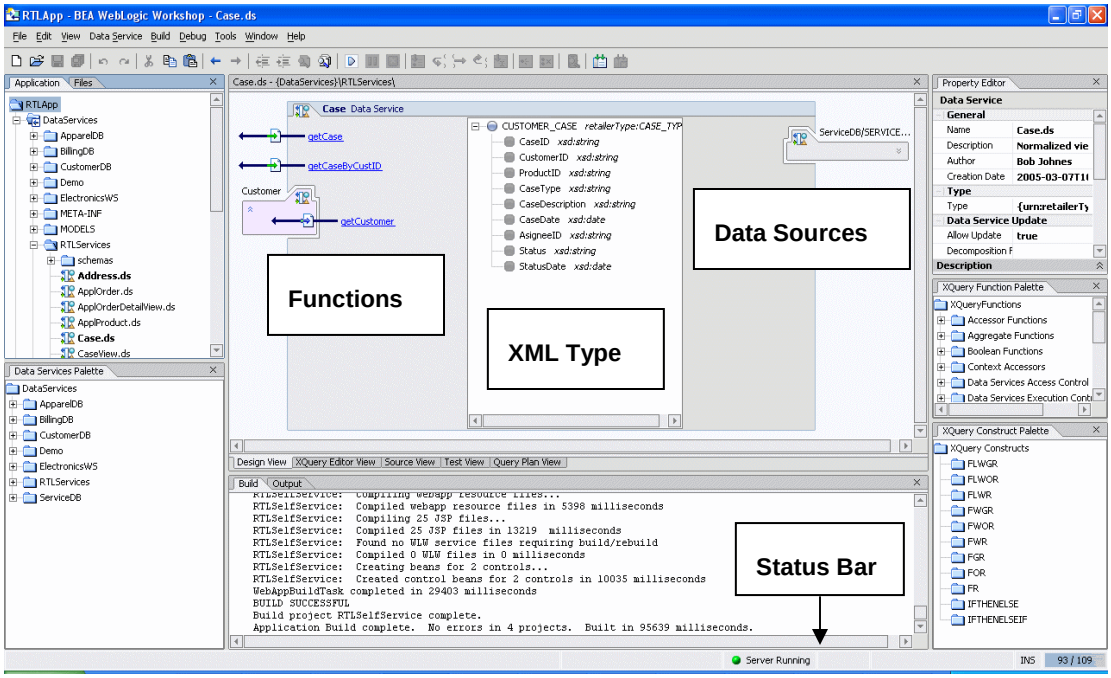


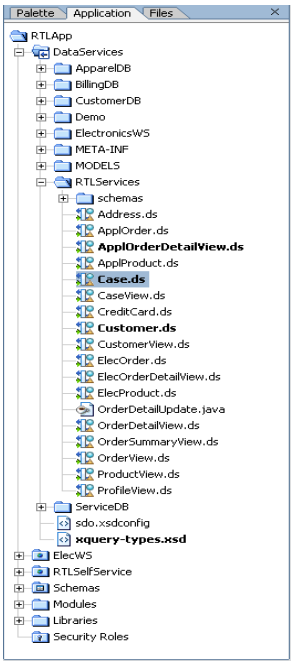
Figure 1-2 Data Services Platform Running in WebLogic Workshop

Objectives

In this lab, you will:

- Explore five of the most frequently used development tools.
- Discover the features and functions of these tools.

Application Pane



The *Application* pane displays a hierarchical representation of a DSP application.

A Workshop *application* is a collection of all resources and components—projects, schemas, modules, libraries, and security roles—deployed as a unit to an instance of WebLogic Server. Only one application can be active at a time. Open files display in boldface type.

If the *Application* pane is not open, complete one of the following options:

- Choose View → Application.
- Press Alt+1.

Design View

Design View presents an editable, graphical representation of a data service. It is a single point of consolidation for a data service's query functions and other business logic. Using Design View, you can:

- View the data service's XML type, native data types, functions, and data source relationships.
- Add functions and data source relationships.
- Create an XML type definition for elements within the data service, such as xs:string or xs:date.
- Associate the data service with an XML Schema Definition (`.xsd`) that defines the unified view for all retrieved data.

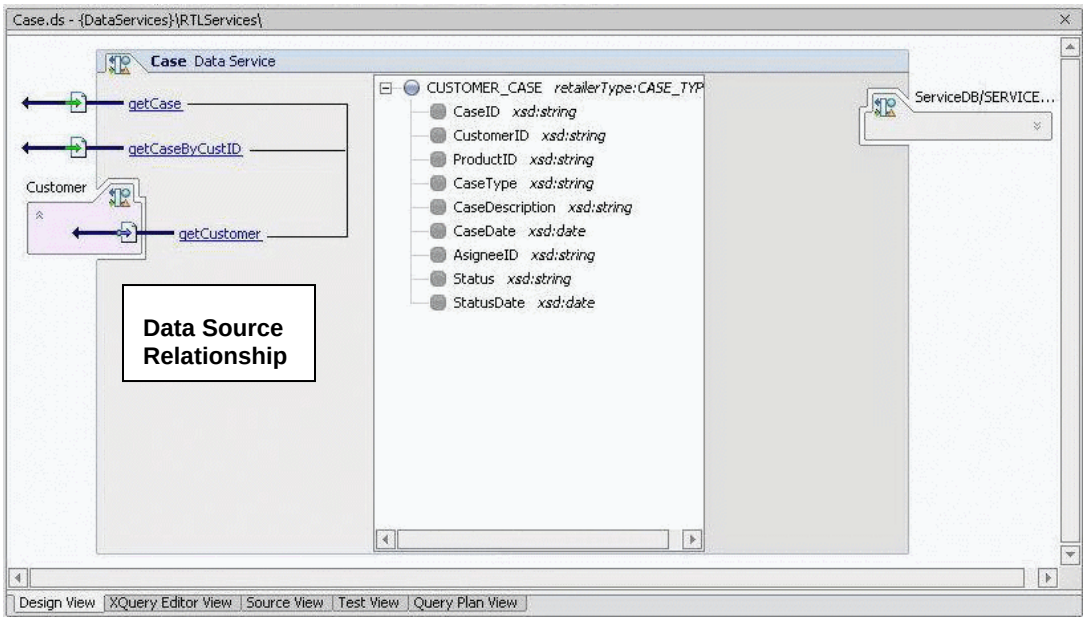


Figure 1-3 Design View of a Logical Data Service

If Design View is not open, complete the following steps:

1. Open a data service such as `Case.ds` located in `DataServices/RTLServices`.
2. Select the Design View tab.

XQuery Editor View

XQuery Editor View provides a graphical, drag-and-drop approach to constructing queries. Using this view, you can inspect or edit the query Return type and add the data source nodes, parameters, expressions, conditions, and source-to-target mappings that comprise data service query functions.

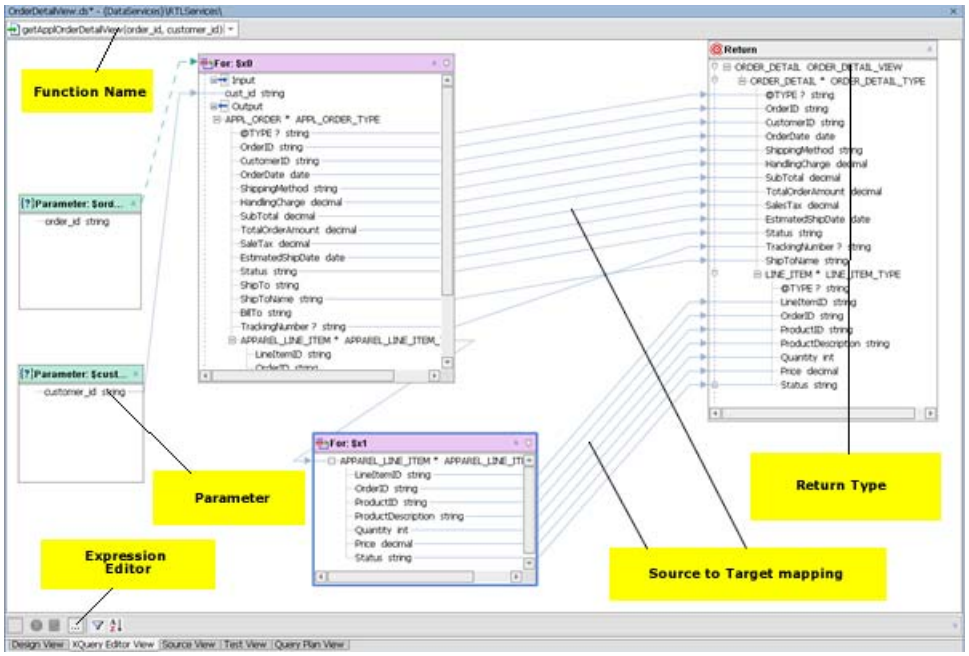


Figure 1-4 Sample XQuery Editor View

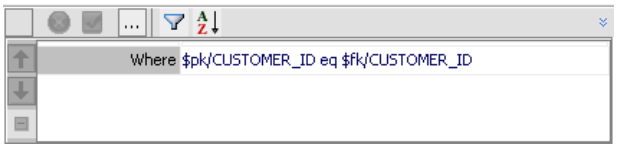
If XQuery Editor View is not open:

1. Open a data service such as `Case.ds` located in `DataServices/RTLServices`
2. Select the XQuery Editor View tab.

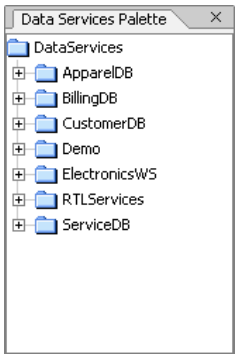
XQuery Editor View Tools

XQuery Editor View includes several editors and palettes that simplify the construction of queries:

- **Expression Editor.** Lets you add where and order by conditions to let or for nodes. The Expression Editor is only active when you click on the specific node header.



- **Data Services Palette.** Lets you add previously-defined query functions as data sources. Each function displays as a for node, which serves as a for clause within the FLWOR (for-let-where-order by-return) statement that is the heart of an XQuery.



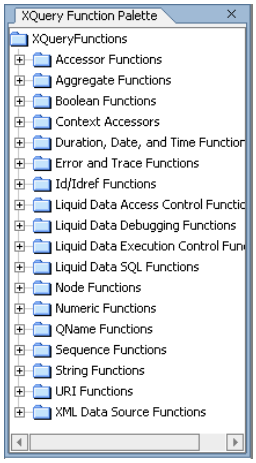
  To add data sources, drag and drop an item from the Data Services Palette into the XQuery Editor View work area. After you drop the node into XQuery Editor View, the node's data source schema (shape) displays in the XQuery Editor View.

  If the Data Services Palette is not open, choose View → Windows → Data Services Palette.

  XQuery Function Palette. Lets you add any of the more than 100 built-in functions provided within the XQuery language. In addition, you can add any of the special built-in functions defined by BEA.

  To add a built-in function, drag and drop the selected item into the Expression Editor.

  If XQuery Function Palette is not open, choose View → Windows → XQuery Function Palette.



Any work created in XQuery Editor View is immediately reflected in Source View, which permits you to augment the graphical approach to constructing queries with direct work on the XQuery syntax. Two-way editing is supported. Changes you make in Source View are reflected in XQuery Editor View, and vice versa.

Source View

Source View lets you view and/or modify a data service's XQuery annotated source code. Although DSP provides extensive visual design tools for developing a data service, sometimes you may need to work directly with the underlying XQuery syntax.

Two-way editing is supported. Changes you make in Source View are reflected in XQuery Editor View, and vice versa.

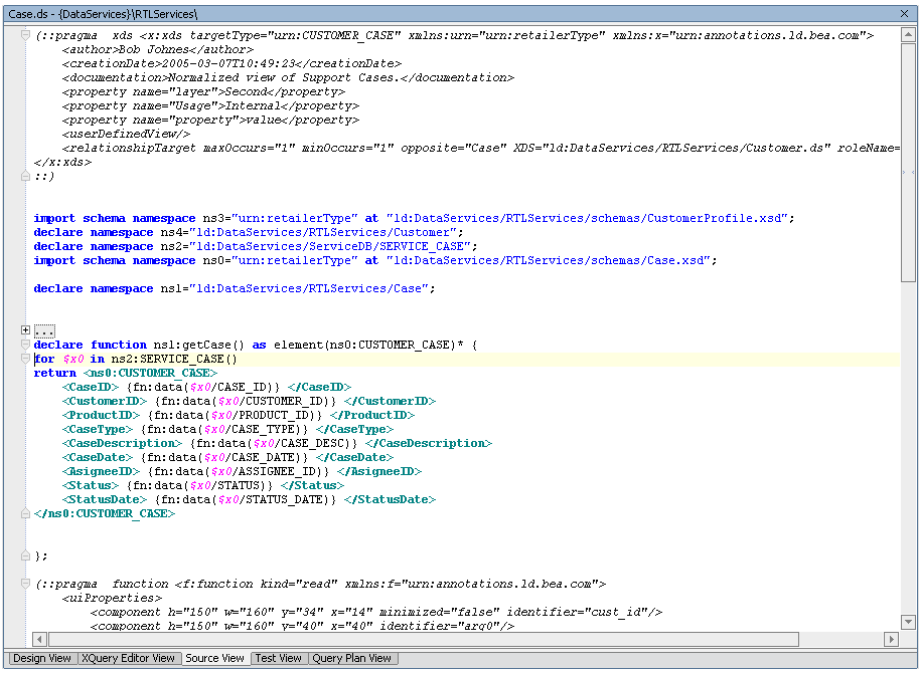


Figure 1-5 Source View

If Source View is not open, complete the following steps:

1. Open a data service such as `Case.ds` located in `DataServices/RTLServices`.
2. Select the Source View tab.

Within Source View, you can use the XQuery Construct Palette, which lets you add any of several built-in generic FLWOR statements to the XQuery syntax. You can then customize the generic statement to match your particular needs.

To add a FLWOR construct, drag and drop the selected item into the appropriate *declare function* space.

If XQuery Construct Palette is not open, choose View → Windows → XQuery Construct Palette.

Test View

Test View provides a means of running developed query functions within the IDE. Options available in Test View depend on the query being tested. For example, if the query supports parameters, then the Parameters section appears, providing a field for each parameter required by the query.

Using Test View, you can select a specific function, specify appropriate parameters, and execute the query to determine that it is functioning properly. In addition, you can edit the results of the query and pass the modifications back to the underlying data source.

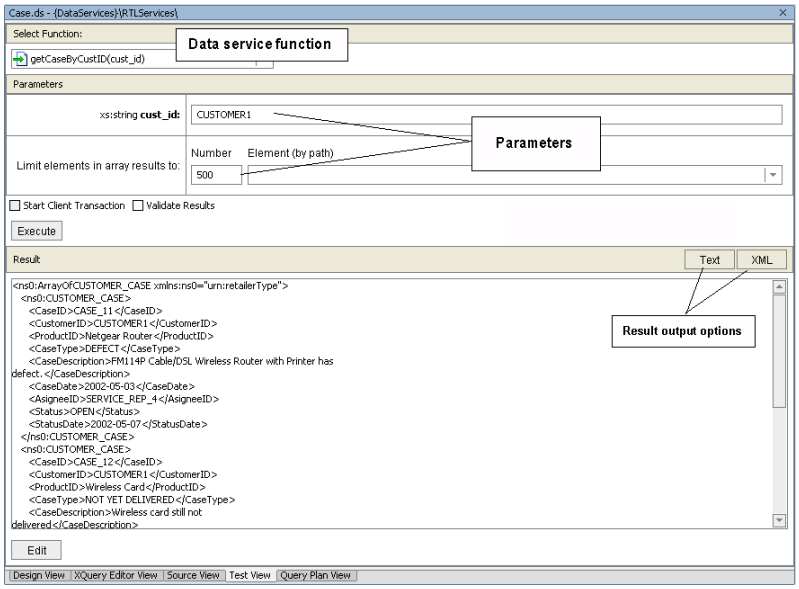


Figure 1-6 Test View

If Test View is not open, complete the following steps:

1. Open a data service such as `Case.ds` located in `DataServices/RTLServices`.
2. Select the Test View tab.

Lab 1.3 Starting WebLogic Server

WebLogic Server need not be running while you are designing a DSP project. However, before you import source metadata or test a developed function, you must start an instance of WebLogic Server.

Any DSP projects that you create will run on your system's installation of WebLogic Server, at least until you deploy them.

Note: Multiple versions of WebLogic Server can exist, even on local, sample systems. If you have previously run an instance of WebLogic Server you should shut down that server and change your WebLogic Workshop server settings. This can be done through the Workshop Tools→Application Properties dialog box.

Objectives

In this lab, you will:

- Discover ways to start WebLogic Server.
- Confirm that your server is running.

Instructions

There are three ways to start WebLogic Server. Start the server using one of the following ways:

Menu Command	**WebLogic Workshop → Tools → WebLogic Server → Start WebLogic Server**
Shortcut Keys	Ctrl + Shift + S
From Status Bar	Right-click the red Server Stopped icon, located at the bottom of the WebLogic Workshop window. Then click Start WebLogic Server.

Starting the WebLogic Server may take some time. During the server startup sequence, you *may* see the following message box:

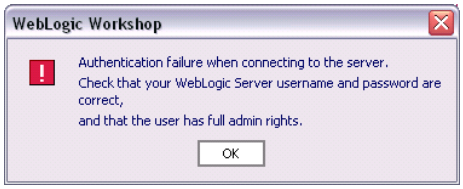
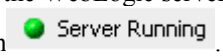


Figure 1-7 (Possible) WebLogic Server Startup Message

If this box displays, click OK.

When WebLogic Server is running, the WebLogic server icon, which appears on the WebLogic Workshop status bar, will turn green 

.

Lab 1.4 Stopping WebLogic Server

There may be times when you want to stop WebLogic Server while still working within DSP for WebLogic Workshop.

Objectives

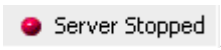
In this lab, you will:

- Discover how to stop WebLogic Server.
- Confirm that the server is not running.

Instructions

You can stop WebLogic Server using any one of the following ways:

Menu Command	**WebLogic Workshop → Tools → WebLogic Server → Stop WebLogic Server**
Shortcut Keys	Ctrl + Shift + T
Procedure	Right-click the green Server Running icon, located at the bottom of the WebLogic Workshop window. Then click Stop WebLogic Server.

Check the WebLogic Server icon of WebLogic Workshop to determine whether WebLogic Server is stopped. If WebLogic Server is stopped, the icon will turn red 

.

Lab 1.5 Saving Your Work

As you build your data services, you may want to save your work on a regular basis.

Objectives

In this lab, you will:

- Discover three ways to save your work while working within the application.
- Discover how to save one or more files when exiting the application or closing WebLogic Workshop.

Instructions

You can save your work using the following commands:

Menu Command	Icon
• **File → Save**	
• **File → Save As**	Not Applicable
• **File → Save All**	

Save All is generally recommended for DSP applications. The Save As and Save All options are only available if you have made changes to your application.

In addition, if you exit WebLogic Workshop and there are any unsaved changes, you are provided with an option to save either specific or all edited files.

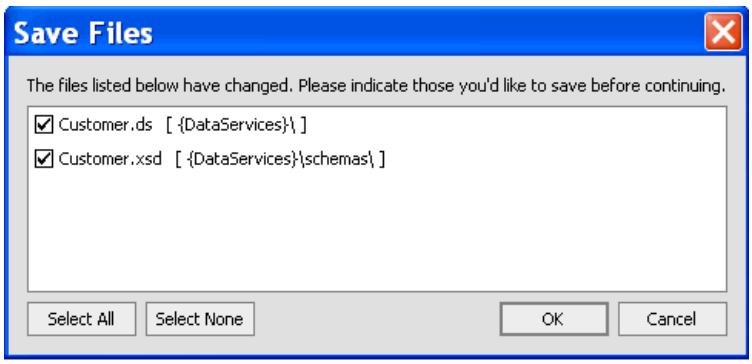


Figure 1-8 Save File Options on Exiting WebLogic Workshop

Lesson Summary

In this lesson, you learned how to:

- Use several of the key tools within DSP for WebLogic Workshop environment.
- Start and stop the WebLogic Server.
- Save files within a Data Services application.