BEA Liquid Data for WebLogic™

Administration Guide

Version 8.1
Document Date: December 2003
Revised: July 2004

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

About This Document

1. Overview of Liquid Data Administration

7. Configuring Access to Relational Databases

8. Configuring Access to XML Files

9. Configuring Access to Delimited Files

10.Configuring Access to Web Services

11.Configuring Access to Application Views

12.Configuring Access to Data Views

13.Deploying Liquid Data Components

14.Configuring Access to Custom Functions

15.Configuring Access to Complex Parameter Types

16.Configuring Stored Queries

17.Importing and Exporting Liquid Data Configurations

18.Managing the Liquid Data Server Repository

19.Security in Liquid Data

20.Monitoring the Server

21.Checking the Status of Liquid Data Resources

22.Configuring the Query Results Cache

23.Generating and Publishing Web Services

Index

About This Document

This document explains how to configure, manage and monitor BEA Liquid Data for WebLogic™.

This document covers the following chapters:

- Chapter 1, "Overview of Liquid Data Administration"
- Chapter 2, "Creating Liquid Data Domains"
- Chapter 3, "Starting and Stopping the Server"
- Chapter 4, "Using the WLS Administration Console"
- Chapter 5, "Configuring Liquid Data Server Settings"
- Chapter 6, "Viewing and Accessing All Configured Data Sources"
- Chapter 7, "Configuring Access to Relational Databases"
- Chapter 8, "Configuring Access to XML Files"
- Chapter 10, "Configuring Access to Web Services"
- Chapter 11, "Configuring Access to Application Views"
- Chapter 12, "Configuring Access to Data Views"
- Chapter 13, "Deploying Liquid Data Components"
- Chapter 14, "Configuring Access to Custom Functions"

- Chapter 17, “Importing and Exporting Liquid Data Configurations”
- Chapter 18, “Managing the Liquid Data Server Repository”
- Chapter 19, “Security in Liquid Data”
- Chapter 20, “Monitoring the Server”
- Chapter 21, “Checking the Status of Liquid Data Resources”
- Chapter 22, “Configuring the Query Results Cache”
- Chapter 23, “Generating and Publishing Web Services”

What You Need to Know

This document is intended mainly for system administrators who will be configuring and managing the Liquid Data data integration platform. Most configuration and management tasks are accomplished through the BEA WebLogic Server Administration Console, so a working knowledge of standard BEA WebLogic Server system administration is helpful in understanding the concepts in this guide. Configuring application views requires using the Application Integration (AI) Application View Console, so some familiarity with AI and application views is helpful if you plan to use application views as Liquid Data data sources.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the “e-docs” Product Documentation page at http://e-docs.bea.com.

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available on the Liquid Data documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the Liquid Data documentation Home page, click the PDF files button and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com/.

Related Information

For more information in general about Java and XQuery, refer to the following sources.

- The Sun Microsystems, Inc. Java site at:

  `http://java.sun.com/`

- The World Wide Web Consortium XML Query section at:

  `http://www.w3.org/XML/Query`

For more information about BEA products, refer to the BEA documentation site at:

`http://edocs.bea.com/`

Contact Us!

Your feedback on the BEA Liquid Data documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the Liquid Data documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA Liquid Data for WebLogic 8.1 release.

If you have any questions about this version of Liquid Data, or if you have problems installing and running Liquid Data, contact BEA Customer Support through BEA WebSupport at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR

Convention	Item
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f file-list]...`
	`[-l file-list]...`

|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

`...`	Indicates one of the following in a command line:
	• That an argument can be repeated several times in a command line
	• That the statement omits additional optional arguments
	• That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f file-list]...`
	`[-l file-list]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Overview of Liquid Data Administration

This section provides an overview of administrative tasks for BEA Liquid Data for WebLogic. It includes the following sections:

- Working with WebLogic Domains
- Starting the Server and Running the Administration Console
- Configuring Access to Data Sources
- Managing the Server Repository
- Implementing Security
- Configuring Query Results Caching
- Configuring Custom Functions
- Importing and Exporting Server Configurations
- Deploying Liquid Data in a Production Environment
- Generating Web Services from Stored Queries
- Ongoing Liquid Data Management Tasks
- Updating License Keys

This document focuses on the use of the administrative tasks that you perform using the WebLogic Server Administration Console, a Web-based tool that has been extended in Liquid Data to include tabs for configuring and managing Liquid Data, accessed via a Liquid Data node in the left pane.

For instructions on how to start and navigate the Administration Console, see Chapter 4, "Using the WLS Administration Console."

Note: This document assumes that you have already installed the BEA WebLogic Platform™ (according to the instructions in "Installing WebLogic Platform" in the WebLogic Platform documentation) and the Liquid Data software (according to the instructions in Installing Liquid Data).

Working with WebLogic Domains

Liquid Data comes with a preconfigured Samples domain (`ld_samples`) from which you can run the `startWebLogic` command or associated Windows Start menu command to start the Liquid Data server. The Samples domain can serve as your starting point for working with the Liquid Data samples, or for developing and testing your own data access and aggregation solutions.

To use Liquid Data beyond the Samples server, you need to either create a new Liquid Data domain or add Liquid Data to an existing WebLogic domain. Thereafter, you start WebLogic Server and Liquid Data in the domain. For instructions on how to create WebLogic domains or add Liquid Data to an existing WebLogic domain, see Chapter 2, "Creating Liquid Data Domains." In addition, for detailed instructions on deploying Liquid Data in various domain configurations, see "Deployment Tasks" in the *Deployment Guide*.

Starting the Server and Running the Administration Console

To configure and manage Liquid Data, you need to start a Liquid Data server in the WebLogic domain in which you want to work and access the WLS Administration Console for that server.

- For instructions on how to start and stop the Liquid Data Server, see Chapter 3, "Starting and Stopping the Server."
- For instructions on how to start the Administration Console and configure Liquid Data resources, see Chapter 4, "Using the WLS Administration Console."

Configuring Access to Data Sources

Before Liquid Data can retrieve information from a data source, access to the data source must be configured in the Administration Console. For each data source accessed in a Liquid Data query, you need to configure a data source description using the Data Sources tab on the Liquid Data node in the Administration Console. Additional configuration tasks are required for certain data source types. For detailed instructions, see Chapter 6, "Viewing and Accessing All Configured Data Sources."

Data Source Descriptions

A *data source description* is a group of configuration settings that Liquid Data uses to access a particular data source. Liquid Data requires a configured data source description before it can retrieve information from the data source. You use the Data Sources tab on the Liquid Data node in the Administration Console to create, edit, and remove data source descriptions and assign security policies. The information stored in a data source description varies by data source type, as described in "Supported Data Source Types" on page 1-3.

After you have created a Liquid Data data source description for a data source in the WebLogic Server Administration Console, the data source you configured will show up as a data source on the Builder Toolbar in the Data View Builder. You can then use the Data View Builder to create data views and queries of the information in the data source, often in combination with information from other configured data sources. Alternatively, you can submit hand-coded, ad-hoc queries to the Liquid Data server without using the Data View Builder. You can also invoke Liquid Data stored queries as EJB clients, JSP clients, or Web Services clients. In all cases, the data sources must be configured first.

Supported Data Source Types

The Liquid Data Server supports the use of the following data sources in queries:

Table 1-1 Supported Data Source Types

Type	Where to Find Configuration Instructions
Relational databases	Chapter 7, "Configuring Access to Relational Databases"
XML files	Chapter 8, "Configuring Access to XML Files"
Web Services	Chapter 10, "Configuring Access to Web Services"
Application views	Chapter 11, "Configuring Access to Application Views"
Data views	Chapter 12, "Configuring Access to Data Views"

The steps required to make a data source available to Liquid Data server (and thereby to the Data View Builder) depend on the type of data source that you want to work with. For example, to make a relational database available to the Liquid Data Server, you first need to configure a JDBC connection pool and a JDBC data source that uses the connection pool, and then create the Liquid Data data source description for the relational database. Similarly, configuring access to an XML file or data view involves storing the XML file or data view file in the Liquid Data Server repository, then creating a data source description to find the appropriate file.

Managing the Server Repository

The server repository is the central location for storing and sharing stored queries, data views, XML data, source and target schemas, Web Service WSDL files, generated Web Services, and custom function libraries.

You need to configure the location of the server repository on the General tab on the Liquid Data node in the Administration Console, as described in Chapter 5, "Configuring Liquid Data Server Settings." You also need to populate and configure the repository on the Repository tab on the Liquid Data node in the Administration Console. For instructions, see Chapter 18, "Managing the Liquid Data Server Repository".

Implementing Security

WebLogic Server provides the foundation for Liquid Data security. Liquid Data deployments can use the full range of security features that WebLogic Server provides, including security policies. Liquid Data uses security policies to control how users access and execute a query, and how users access specific data source elements (such as particular tables in a database, service calls in an application view, or a web service) for ad-hoc queries or custom functions. For details, see Chapter 19, "Security in Liquid Data."

Configuring Query Results Caching

Liquid Data can cache query results for stored queries (but not ad-hoc queries) to enhance overall Liquid Data performance. If you want to cache results for stored queries in this deployment, you must explicitly enable results caching on the General tab on the Liquid Data node in the Administration Console, as described in Chapter 5, "Configuring Liquid Data Server Settings." In addition, for each stored query that you want cached, you need to explicitly configure its caching policy in the Repository, as described in Chapter 22, "Configuring the Query Results Cache."

Configuring Custom Functions

Liquid Data provides a set of standard functions for use in creating queries and data views. Users can extend Liquid Data by creating custom functions to perform specialized tasks. If custom functions are used in this deployment, you need to configure access to them, as described in Chapter 14, "Configuring Access to Custom Functions."

Importing and Exporting Server Configurations

If you need to copy your Liquid Data server configuration to another server (such as from a development environment to a production environment), you can use the Administration Console to export the Liquid Data server configuration from the source server and import it on a target server. For more information, see Chapter 17, "Importing and Exporting Liquid Data Configurations," and also "Copying a Server Configuration to Another Server" in "Deployment Tasks" in the *Deployment Guide*.

Deploying Liquid Data in a Production Environment

Liquid Data is deployed as an enterprise archive file (`LDS.ear`) on a WebLogic domain. For instructions on how to deploy the `LDS.ear` file, see Chapter 13, "Deploying Liquid Data Components." For detailed information about deploying Liquid Data in various types of WebLogic domains, see "Deployment Tasks" in the *Deployment Guide*.

Generating Web Services from Stored Queries

Using the Administration Console, you can publish Liquid Data stored queries as Web Services. Web-based applications can then invoke Liquid Data queries as Web Service clients. For more information, see Chapter 23, "Generating and Publishing Web Services."

Ongoing Liquid Data Management Tasks

Ongoing managing and monitoring tasks include starting and stopping the server; updating data source configurations; and setting up and monitoring logs and reports on Liquid Data Server performance and lifecycle.

It is a good practice to frequently export you Liquid Data configuration and to store the resulting file in a secure environment. This is especially the case whenever you change your Liquid Data configuration since a recently exported configuration will allow you to "roll back" in case of problems with your new configuration or its interaction with the Platform server. For more information, see

Chapter 17, “Importing and Exporting Liquid Data Configurations,” and also “Copying a Server Configuration to Another Server” in “Deployment Tasks” in the *Deployment Guide*.

For detailed information on server management and monitoring, see Chapter 20, “Monitoring the Server.”

For detailed information about tuning Liquid Data performance, see “Tuning Performance” in the *Deployment Guide*.

Updating License Keys

Liquid Data requires a valid product license to run. The Liquid Data license is an extension of the WebLogic server license. For details about the BEA product license, see Installing and Updating a WebLogic Server License in the WebLogic Server documentation.

Creating Liquid Data Domains

This chapter introduces the concept of WebLogic domains, and explains how to create new WebLogic domains for BEA Liquid Data for WebLogic, or to add Liquid Data to an existing WebLogic domain. The following topics are included:

- Understanding WebLogic Domains and Administration
- Understanding the Relationship of Liquid Data to WebLogic Domains
- Creating a New Domain
- Adding Liquid Data to an Existing Domain

For detailed information about adding Liquid Data to other types of WebLogic domains, such as WebLogic Platform, WebLogic Portal, WebLogic Integration, or WebLogic Workshop domains, see "Deployment Tasks" in the *Deployment Guide*.

Understanding WebLogic Domains and Administration

A WebLogic *domain* is a collection of WebLogic resources managed as a single unit. A WebLogic domain includes one or more instances of WebLogic Server and may include WebLogic Server clusters. For more information about domains, see "WebLogic Server Domains" in *Configuring and Managing WebLogic Server*.

The *Administration server* is the central point of control for an entire domain. If there is only one server in a domain, that server is the Administration server in addition to the other functions it provides. Any other servers in the domain are *Managed servers*.

Understanding the Relationship of Liquid Data to WebLogic Domains

Liquid Data is an application and a set of associated resources that are deployed in a WebLogic *domain*. Starting, stopping, and managing Liquid Data is accomplished by starting the WebLogic Server in a particular domain in which Liquid Data is deployed, and using the Administration Console for that server to configure and manage Liquid Data resources for that domain. The full installation of Liquid Data includes a preconfigured Samples domain as a getting started example.

When you are ready to set up your own Liquid Data domains and servers, you must create new Liquid Data domains or add Liquid Data to your existing WebLogic Server or WebLogic Integration domains.

Creating a New Domain

If you are creating a new WebLogic domain for use with Liquid Data, you can use the WebLogic Platform Configuration Wizard. For more information, see "Creating a New WebLogic Domain" in the WebLogic Platform documentation.

When you install Liquid Data into a WebLogic Platform, several Liquid Data configuration templates are installed for use with the Configuration Wizard. For details on creating these domains, as well as on extending an existing domain to use Liquid Data, see the Liquid Data *Deployment Guide*.

Adding Liquid Data to an Existing Domain

Once you have WebLogic Server domain in which you want to use Liquid Data, the next step is to *deploy* the *Liquid Data* application and resources into that domain. For more information see "Deployment Tasks" in the *Deployment Guide*.

Starting and Stopping the Server

This section describes how to start and stop the BEA Liquid Data for WebLogic server. It includes the following sections:

- Starting WebLogic Server and the Liquid Data Server
- Stopping the Server
- Next Steps

Starting WebLogic Server and the Liquid Data Server

Before you can configure or manage Liquid Data, you must start the Liquid Data server, which runs as an application in a WebLogic domain. Starting the WebLogic Server in the appropriate WebLogic domain automatically starts Liquid Data.

When you run the `startWebLogic.cmd` (Windows) or `startWebLogic.sh` (UNIX) command for a domain, WebLogic Server is started, and the Liquid Data applications and resources specified in the configuration file for the domain are automatically deployed on the server. The Liquid Data pre-configured domains are shown in "Start the Server in the Appropriate WebLogic Domain" on page 3-2 along with quick summary of Windows menu paths and UNIX command line paths for starting the server.

Note: The instructions that follow are tailored for starting the WebLogic Server in conjunction with Liquid Data. For general information on starting the WebLogic Server, see Starting and Stopping WebLogic Servers (http://edocs.bea.com/wls/docs81/ConsoleHelp/startstop.html) in the WebLogic Server documentation.

Start the Server in the Appropriate WebLogic Domain

You must start the Liquid Data server in the appropriate WebLogic domain. A pre-configured domain is provided for the Samples server. To create new Liquid Data servers of your own, you need to use the WebLogic Platform Configuration Wizard. For more information, see "Deployment Tasks" in the *Deployment Guide* and "Creating Domains Using the Configuration Wizard" in the WebLogic Platform documentation.

Note: Make sure you run the First-Time Samples Configuration before running the Samples server for the first time.

Table 3-1 Liquid Data Samples Pre-configured Domain and Start Commands for Samples Server

Platform	Windows and UNIX Paths to Start Samples Server in Each Domain
Windows	• Star t—> Programs —> BEA WebLogic Platform 8. 1—> Liquid Data for WebLogic 8.1 —>*Launch Samples Server* Or • `WL_HOME\samples\domains\liquiddata\startWebLogic.cmd`
UNIX	`WL_HOME/samples/domains/liquiddata/startWebLogic.sh`

Which server you start depends upon whether you want to use the Samples server which comes with preconfigured data sources, or one of your own servers in a new domain you create with the WebLogic Platform Configuration Wizard.

Starting the Server

The instructions in this section describe how to start WebLogic Server in a standalone WebLogic domain. For multi-node or clustered domains, see the instructions in "Deployment Tasks" in the *Deployment Guide.*

Note: If you are already running an instance of WebLogic Server that uses the same listen port as the one to be used by the server you are starting, you must stop the first server before starting the second server.

To start the server:

1. At the command prompt, go to the domain directory (`BEA_HOME/user_projects/domain_name`), such as `c:\bea\user_projects\mydomain`.

2. Run the server startup script: `startWebLogic.cmd` (Windows) or `startWebLogic.sh` (UNIX).

   The startup script displays a series of messages, finally displaying something similar to the following message when the server has started successfully:

   ```
   <Oct 8, 2002 3:50:42 PM PDT> <Notice> <WebLogicServer> <000360> <Server
   started in RUNNING mode>
   ```

Stopping the Server

You can stop your entire Liquid Data system—WebLogic Server (WLS), the Liquid Data server, and its resources deployed in a pre-configured domain—from the WLS Administration Console.

The instructions in this section describe how to stop WebLogic Server in a standalone WebLogic domain. For multi-node or clustered domains, see the instructions in "Deployment Tasks" in the *Deployment Guide.*

Note: We recommend using the Administration Console to shut down the server gracefully rather than shutting down from a DOS window or UNIX shell.

To stop the WLS server using the Administration Console:

1. If you have not already done so, start the Administration console in a Web browser and open the URL for your server in the form:

   `http://`*`HostName`*`:`*`Port`*`/console`

   For example, to start the Administration Console for a local instance of WebLogic Server (running on your own machine), type the following URL in a Web browser address field:

   `http://localhost:7001/console/`

   Note: For complete details on how to start the Administration Console see "Starting the Administration Console" on page 4-2 in Chapter 3, "Starting and Stopping the Server."

2. In the left pane, expand the Servers node.

3. Click on the server running in your Liquid Data domain that you want to stop.

   A set of tabs for configuring and monitoring the server is shown.

4. Click on the Control tab.

5. Click on the Shut down this server... link.

6. Click Yes to confirm the server shutdown.

Note: On Unix, you must also manually kill the running Pointbase instance.

Next Steps

Once you have the server started, you need to start the WLS Administration Console. You can use the WLS Administration Console to perform all Liquid Data configuration, management, and monitoring tasks. For information on how to start the console and find the Liquid Data node on the console, see Chapter 4, “Using the WLS Administration Console.”

CHAPTER 4

Using the WLS Administration Console

This chapter describes how to use the WebLogic Server Administration Console, which includes tabs for configuring BEA Liquid Data for WebLogic. It includes the following sections:

- Using the Administration Console to Manage Liquid Data
- Starting the Administration Console
- Overview of the Administration Console
- Finding the Liquid Data Node in the Navigation Tree

Using the Administration Console to Manage Liquid Data

You can configure, manage, and monitor Liquid Data through the BEA WebLogic Server Administration Console. When Liquid Data is installed, it is deployed as an application in an instance of WebLogic Server. When you start the Liquid Data Server, the hosting WebLogic server is automatically started. Upon installation, Liquid Data becomes a *managed resource* known to the WLS JMX management framework. You will use the tabs on the Liquid Data node in the Administration Console to ad and configure Liquid Data data sources.

Starting the Administration Console

To start the Administration Console:

1. Start the WebLogic Server in the WebLogic domain in which Liquid Data is deployed. For more information, see "Starting WebLogic Server and the Liquid Data Server" on page 3-1.

2. Open a web browser (either Netscape 4x or higher, or Internet Explorer 4.x or higher) and open the following URL:

   `http://`*`hostname`*`:`*`port`*`/console`

   Where

   – *`hostname`* is the machine name or IP address of the host server

   – *`port`* is the address of the port on which the host server is listening for requests (7001 by default)

   For example, to start the Administration Console for a local instance of WebLogic Server (running on your own machine), type the following URL in a Web browser address field:

   `http://localhost:7001/console/`

   If you started the Administration Server using Secure Socket Layer (SSL), you must add `s` after `http`, as follows:

   `https://`*`hostname`*`:`*`port`*`/console`

   Note: On Windows, you can start the Administration Console through the start menu: Start—>Programs—>BEA WebLogic Platform 8.1—>Liquid Data for WebLogic 8.1—>Liquid Data Samples—>Admin Console

3. When the login page appears, enter the user name and the password you used to start the Administration Server.

   Note: To change certain Liquid Data server settings, such as caching or file swapping, you must log in with a username that belongs to the `LDAdmin` group.

If you have your browser configured to send HTTP requests to a proxy server, then you may need to configure your browser so that it does not send Administration Server HTTP requests to the proxy. If the Administration Server is on the same machine as the browser, then ensure that requests sent to localhost or 127.0.0.1 are not sent to the proxy.

The Administration Console is accessible via a URL in the following form:

```
http://localhost:7001/console/
```

Overview of the Administration Console

When you start the Administration Console, a server home page is shown in the main display area on the right, as shown in Figure 4-1. You can use the topic links on the home page initially to navigate to top level resource nodes, or use the navigation tree in the left pane. The left pane in the Administration Console contains a hierarchical tree — the domain tree — for navigating to tables of data, configuration pages and monitoring pages, or accessing logs. By selecting (that is, left-clicking) an item in the domain tree, you can display a table of data for resources of a particular type (such as WebLogic Servers) or configuration and monitoring pages for a selected resource.

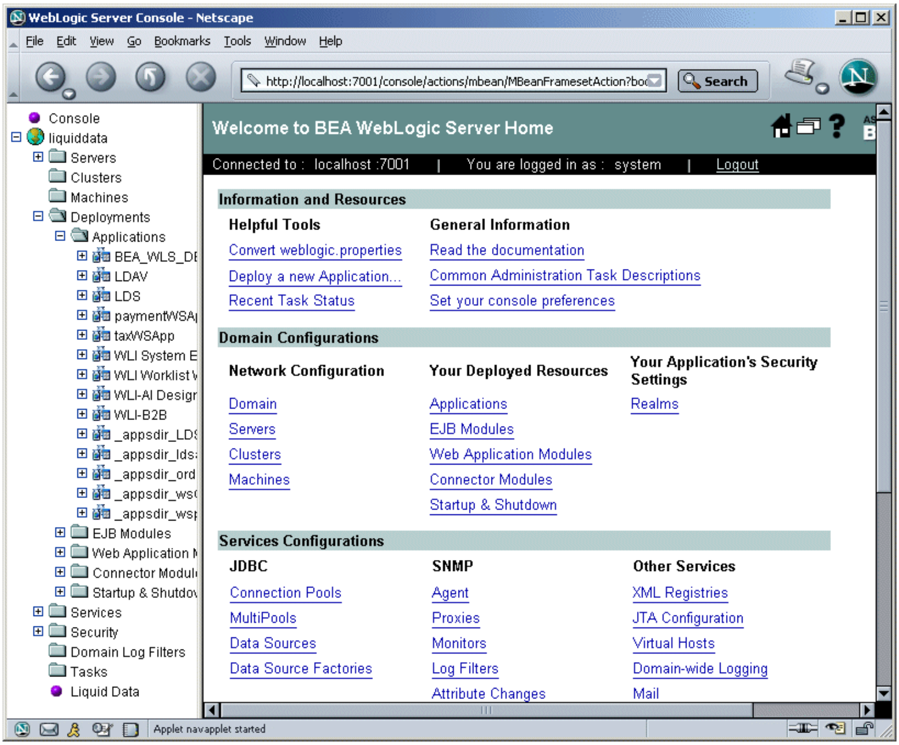
You can expand and collapse nodes in the tree by clicking on the + and - signs next to the nodes as follows:

- A plus sign is (+) next to a node indicates that the node contains subnodes it is expandable. To expand a collapsed container node, click on the + beside it. Its next level subnodes will display. You can continue expanding subnodes if they have + next to them.
- A minus sign (-) next to a node indicates that the node is a container that is fully expanded. To collapse an expanded container node, click on the - beside it.
- A node with neither - or + beside is either an empty folder with no resources as yet or a fixed resource with no subnodes. As you add resources to folders, these will become expandable containers.

To manage Liquid Data you will need to access and use console pages for standard WebLogic Server resources as well as console pages specific to Liquid Data resources.

For a detailed overview on using the Administration Console, see Starting the Administration Console (http://edocs.bea.com/wls/docs81/adminguide/overview.html#start_admin_console) in the WebLogic Server documentation.

Figure 4-1 Home Page of the WebLogic Server Administration Console


WebLogic Server Console - Netscape
File Edit View Go Bookmarks Tools Window Help
http://localhost:7001/console/actions/mbean/MBeanFramesetAction?boo
Search
Applet navapplet started

- Console
- liquiddata
  - Servers
  - Clusters
  - Machines
  - Deployments
    - Applications
      - BEA_WLS_DE
      - LDAV
      - LDS
      - paymentWSA
      - taxWSApp
      - WLI System E
      - WLI Worklist V
      - WLI-AI Design
      - WLI-B2B
      - _appsdir_LDS
      - _appsdir_lds
      - _appsdir_ord
      - _appsdir_ws
      - _appsdir_wsp
    - EJB Modules
    - Web Application M
    - Connector Modul
    - Startup & Shutdov
  - Services
  - Security
  - Domain Log Filters
  - Tasks
  - Liquid Data

Welcome to BEA WebLogic Server Home

Connected to : localhost :7001 | You are logged in as : system | Logout

Information and Resources

Helpful Tools	General Information
Convert weblogic.properties	Read the documentation
Deploy a new Application...	Common Administration Task Descriptions
Recent Task Status	Set your console preferences

Domain Configurations

Network Configuration	Your Deployed Resources	Your Application's Security Settings
Domain	Applications	Realms
Servers	EJB Modules	
Clusters	Web Application Modules	
Machines	Connector Modules	
	Startup & Shutdown	

Services Configurations

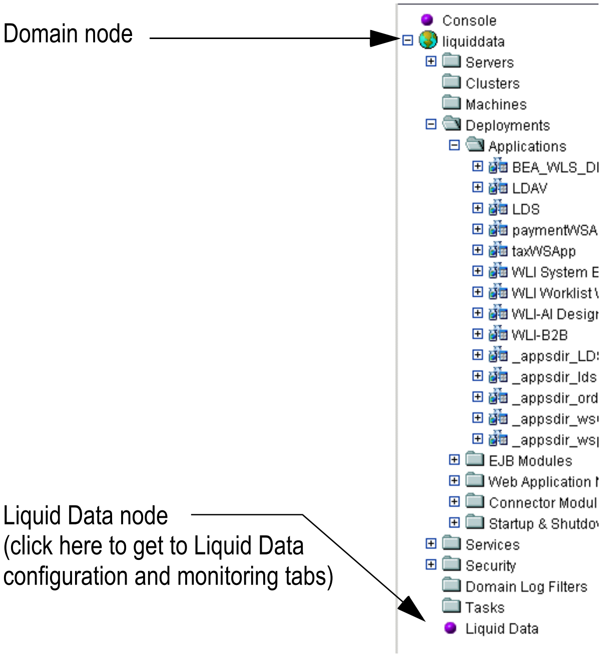
JDBC	SNMP	Other Services
Connection Pools	Agent	XML Registries
MultiPools	Proxies	JTA Configuration
Data Sources	Monitors	Virtual Hosts
Data Source Factories	Log Filters	Domain-wide Logging
	Attribute Changes	Mail

Finding the Liquid Data Node in the Navigation Tree

The Liquid Data node is under the domain node is under the domain node at the Administration Console navigation tree. In the figure below, the navigation tree for the Liquid Data Samples server is shown. The domain name for the Samples is `ld_samples`.

To access the Liquid Data data source configuration and monitoring tabs, click the Liquid Data node in the navigation tree.

Figure 4-2 Liquid Data Resources Shown in WLS Administration Console Navigation Tree



Domain node

Liquid Data node
(click here to get to Liquid Data
configuration and monitoring tabs)

CHAPTER 5

Configuring Liquid Data Server Settings

This chapter describes how to configure server settings for BEA Liquid Data for WebLogic. It includes the following sections:

- Configuring Server Settings
- Modifying Server Settings

In a standalone (single server) domain, server settings apply to a single instance of Liquid Data server. In a clustered domain, to all Managed servers in the domain. Server settings include the repository directory, threads for application views and Web Services data sources, the security mode, whether to cache results for stored queries, the swap files directory for stored queries, and the classpath for custom functions.

Note: To change Liquid Data server settings, you must log in to the Administration Console with a username that belongs to the `LDAdmin` group.

Configuring Server Settings

You can set configuration options on the Liquid Data server that apply to all processing on the selected server for any type of data source. This section describes the fields in the Administration Console and provides a procedure for configuring Liquid Data server settings. It contains the following sections:

- Server Console Page Field Descriptions
- To Configure Liquid Data Server Settings

Server Console Page Field Descriptions

The following table describes the fields in the Liquid Data console server page.

Table 5-1 Liquid Data Server Configuration Settings

Field	Description
Repository Directory	Full or relative path to the root directory of the Liquid Data repository that contains the data sources configured for this Liquid Data server. For relative paths, the path is relative to the current domain directory. For more information, see "Server Repository Location" on page 18-3. In a clustered environment, all managed Liquid Data servers must mount (on Unix) or be mapped to (on Windows), the volume containing the directory specified here. If the specified directory does not exist, Liquid Data will create it automatically.
Custom Functions Classpath	Classpath for libraries used by custom functions that do *not* reside in the `custom_lib` folder in the server repository. Use semicolons to separate paths. File pathnames or URLs can be used. For example: `c:/path/cf1.jar;c:/path2/cf2.jar` For information on how to configure custom function descriptions, see Chapter 14, "Configuring Access to Custom Functions" and "Server Repository File System Hierarchy" on page 18-4.

Table 5-1 Liquid Data Server Configuration Settings (Continued)

Field	Description
Maximum Threads	Maximum number of threads in the Liquid Data server pool used to handle query requests for application view, web service, and custom function data sources. The default setting is 20. The minimum setting is 1. If the specified value is invalid, then the server will use the default value of 20. **Note:** The maximum threads value that you specify here *does not* affect the WebLogic Server server thread pool. The value specified here applies only to the thread pool created and used by the Liquid Data query engine for processing requests on application view, web service, or custom function data sources.
Maximum Threads Per Query	Maximum number of threads allowed for a single query. Use this to limit the number of threads spawned by a single query. No matter what the settings for the Maximum Number of Threads Per Query, the actual number of threads used will not exceed the maximum number of threads specified in Maximum Threads. The default setting is 4. The minimum setting is 1. If the specified value is invalid, then the server will use the default value of 4. **Note:** The maximum threads value that you specify here *does not* affect the WebLogic Server server thread pool. The value specified here applies only to the thread pool created and used by the Liquid Data query engine for processing requests on application view and web service data sources.
Cache Results	Enables or disables (default) the caching of query results for stored queries. • To enable results caching, enable (check) this check box. • To disable results caching, clear (uncheck) this check box. For more information about caching, see "Enabling the Results Cache" on page 22-4.

Table 5-1 Liquid Data Server Configuration Settings (Continued)

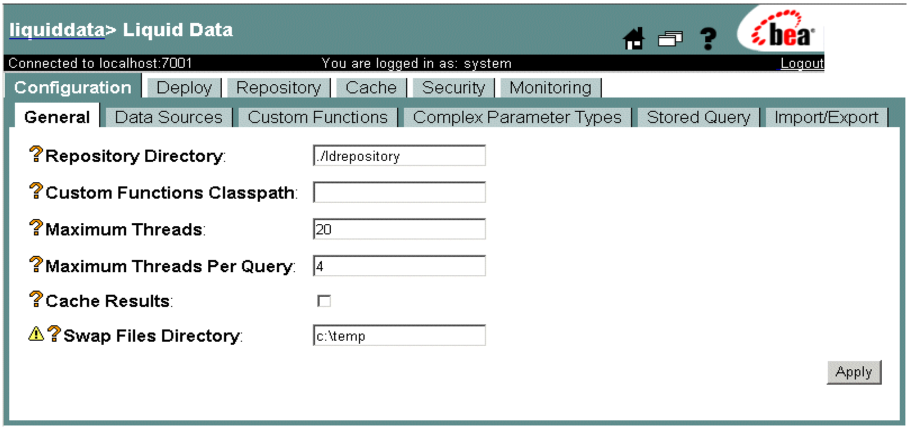
Field	Description
Swap Files Directory	Path of location of swap files that Liquid Data uses to manage large amounts of intermediate results data returned from a stored or ad-hoc query. If the Large Results flag is selected for the query, then Liquid Data uses swap files to temporarily store intermediate results on disk. Default is `temp`. This location is a subdirectory of the *`BEA_HOME`*`/user_projects/`*`domain_name`* directory. If the specified directory does not exist, Liquid Data will create it automatically. **Note:** If you change this setting, you must *reboot* the Liquid Data server in order for the change to take effect.

To Configure Liquid Data Server Settings

Perform the following to configure Liquid Data server settings:

1. In the left pane of the Administration Console, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the General tab.

Figure 5-2 Configuring Liquid Data Server Settings



Repository Directory: ./ldrepository

Custom Functions Classpath:

Maximum Threads: 20

Maximum Threads Per Query: 4

Cache Results: □

Swap Files Directory: c:\temp

4. If you have logged into the Administration Console with a username that belongs to the `LDAdmin` group, then fill in the fields described in Table 5-1, "Liquid Data Server Configuration Settings," on page 5-2.

5. If you have logged into the Administration Console with a username that belongs to the `LDAdmin` group, then click Apply to apply any changes.

6. If you changed the swap files directory, reboot the Liquid Data server for the change to take effect.

Modifying Server Settings

To modify server settings:

1. In the left pane, click the Liquid Data node.

2. In the right pane, click the Configuration tab.

3. Click the General tab.

4. Modify the fields, described in Table 5-1, "Liquid Data Server Configuration Settings," on page 5-2, as needed.

5. Click Apply.

CHAPTER 6

Viewing and Accessing All Configured Data Sources

A *data source* is a source of information that can be queried. Liquid Data supports querying the following types of data sources: relational databases (RDBMSs) via JDBC, XML files, Web Services, application views, and data views (which are the dynamic results of queries stored along with the queries that produce them).

This chapter describes how to view and access configure BEA Liquid Data for WebLogic data sources using the All Data Sources configuration tab on the Liquid Data node in the Administration Console. It includes the following sections:

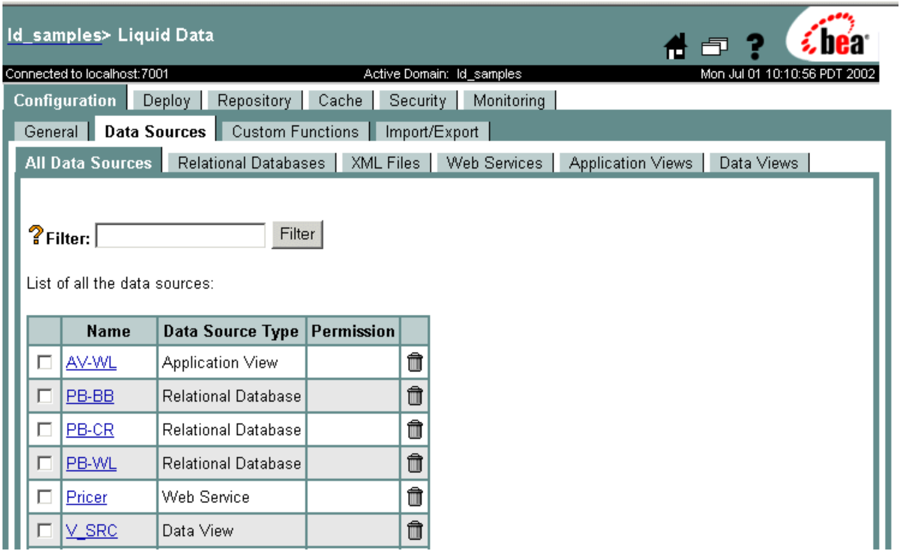
Viewing All Configured Data Sources

To view all currently configured data sources:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the All Data Sources tab.

   The tab shows a list of all data sources currently configured on the Liquid Data server to which you are connected.

Figure 6-1 Viewing All Configured Data Sources



Connected to localhost:7001 Active Domain: ld_samples Mon Jul 01 10:10:56 PDT 2002

Filter:

List of all the data sources:

	Name	Data Source Type	Permission	
□	AV-WL	Application View		🗑
□	PB-BB	Relational Database		🗑
□	PB-CR	Relational Database		🗑
□	PB-WL	Relational Database		🗑
□	Pricer	Web Service		🗑
□	V_SRC	Data View		🗑

5. You can filter on a name or a partial name, as described in Table 6-2, “Viewing All Configured Data Sources,” on page 6-3.
6. To view or modify the configuration for a specific data source, click on the data source name.

The Administration Console displays the configuration tab for that data source.

Table 6-2 Viewing All Configured Data Sources

Field	Description
Filter	A simple filter that limits the list of displayed data source descriptions by name. The filter is case-sensitive. For example, to search for all data sources with source description names starting with the letters `PB`, type `PB` into the Filter field and then click Filter. The tab is refreshed showing only files that begin with the letters `PB`. The Filter field is case-sensitive, does not accept special characters, and does not accept wildcards.
List of All Data Sources	Shows a linked list of all configured data sources by default (when you first click on the All Data Sources tab on the Liquid Data node). After you've filtered on a name or a partial name, shows a subset of data sources based on what you are filtering for. To get back the full list of all data sources, you need to click off of this tab, and then click the All Data Sources tab again. The full list will be re-displayed.

For more information about editing data source descriptions, see the chapter associated with that data source type. For example, to configure access for relational databases, see "Creating a Relational Database Data Source Description" on page 7-6.

Configuring Secure Access to Data Source Descriptions

You can configure security for each data source description by assigning security policies. For more information about Liquid Data security, see Chapter 19, "Security in Liquid Data."

Permissions determine the tasks that users can perform on data sources in the Data View Builder, Liquid Data applications, and the Administration Console. Users must be logged into the applicable tool with the following permissions:

Table 6-3 Permissions Required for Data Sources

Method (Access Level)	Description
`All`	Allows read, modify, and execute access to the object.
`Read Configuration`	Browse or view the contents of an item, or download from the repository.

Table 6-3 Permissions Required for Data Sources (Continued)

Method (Access Level)	Description
`Modify Configuration`	Create, modify, rename, or delete files or directories, or upload items to the directory. This level implies `read` access.
`Execute Query`	Allows execute permission on the object. Whether a user can actually execute a query (either stored or ad-hoc) is determined dynamically at runtime based on whether a user has execute access to all of the resources the query requires.

To Assign Security Policies to a Data Source

Perform the following steps to assign security policies to a data source:

1. Open the Liquid Data Administration Console and click the Data Sources tab.
2. Navigate to the data source to which you want to assign a security policy.
3. Click Define Security Policy or Edit Security Policy on the data source to which you want to assign security. If the object already has a policy defined, the link is labeled “Edit Security Policy”; if the object has no defined policy, the link is labeled “Define Security Policy.”
4. Assign security policies as needed to the data source. For details, see “Assigning Security Policies to Liquid Data Objects” on page 19-13.

Removing Data Source Descriptions

You can remove a data source description that you no longer need. Removing a data source description does not remove the actual data source to which it refers. You can remove the data source description using the All Data Sources tab on the Liquid Data node or using the summary tab associated with the data source type.

Note: You must log in with `modify` access privledges before you can remove a data source description. For more information, see “Defining Liquid Data Roles and Groups” on page 19-2.

To remove a data source description:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.

4. On the All Data Sources tab or the tab associated with the data source type, select (check) the check box next to the data source description that you want to remove.

5. Click on the trash can next to the data source description.

6. When prompted, click Yes to confirm removal.

   The Administration Console removes the selected data source description.

Note: Removing the data source description does *not* remove the underlying data source to which the Liquid Data data source description pointed. The process of removing the actual data source varies depending on how the data source is set up. For example, you can use the Repository Tab on the Liquid Data node in the Administration Console to remove source XML files from the repository, as described in "Deleting Folders and Files in the Server Repository" on page 18-13.

Distributing Data Source Descriptions to Other Liquid Data Servers

Each Liquid Data server instance must have its own set of data source descriptions. Rather than entering data source descriptions manually on each Liquid Data server, you can simply copy the data source description from one server to another. The Liquid Data node provides an Import / Export tab that you can use to export the data source description to a file that you can then import on other Liquid Data servers. For more information, see Chapter 17, "Importing and Exporting Liquid Data Configurations."

CHAPTER 7

Configuring Access to Relational Databases

Before a BEA Liquid Data for WebLogic query can access data in a relational database, the relational database must be configured as a Liquid Data data source. Once configured according to the instructions in this chapter, relational databases with data source descriptions will show up as data sources in any Liquid Data EJB client, such as the Data View Builder, that connects to this Liquid Data server.

Configuring a relational database source description for Liquid Data consists of several discrete tasks: configuring a WebLogic Server (WLS) JDBC connection pool, then configuring the relational database as a WLS JDBC data source that uses that JDBC connection pool, and finally adding a Liquid Data source description for the relational database that uses the JDBC resources.

The following sections are included:

Connection Pool URLs and Driver Names for JDBC Data Sources

To configure JDBC connection pools for your data sources, you need to provide the URL to your database in the appropriate format for the database type and the full package name of the JDBC driver used by the database. Formats for the database URL and driver class name vary depending on the type of database you are using.

The following table provides URL formats and driver class names for supported databases.

Table 7-1 Connection Pool URL Formats and Driver Class Names for Supported Databases

Database	URL Format	Driver Class Name
PointBase	`jdbc:pointbase://`*`<hostname>`*`:`*`<portnum>`*`/LDDB`	`com.pointbase.jdbc.jdbcUniversalDriver`
Oracle	`jdbc:oracle:thin:@`*`<hostname>`*`:`*`<portnum>`*`:`*`SID`*	`oracle.jdbc.driver.OracleDriver`
Microsoft SQL Server	`jdbc:weblogic:mssqlserver4:CRM@`*`<hostname>`*`:`*`<portnum>`*	`weblogic.jdbc.mssqlserver4.Driver`
Sybase	`jdbc:sybase:Tds:`*`<hostname>`*`:`*`<portnum>`*`/`*`<dbname>`*	`com.sybase.jdbc.SybDriver`
DB2	`jdbc:db2://`*`<hostname>`*`/`*`<dbname>`*	`COM.ibm.db2.jdbc.net.DB2Driver`
Informix	`jdbc:informix-sqli://`*`<hostname>`*`:`*`<portnum>`*`/`*`<database_name>`*`:INFORMIXSERVER=`*`<hostname>`*	`com.Informix.jdbc.IfxDriver`

Creating a JDBC Connection Pool

For complete information on how to create Java Database Connectivity (JDBC) Connection Pools in WebLogic Server, see JDBC (http://edocs.bea.com/wls/docs81/ConsoleHelp/jdbc.html) in the WebLogic Server documentation.

In order to add a relational database data source description to Liquid Data, you first need to create a JDBC connection pool in WLS for the data source to use. Creating a JDBC connection pool consists of first creating the pool and then deploying it on a target server.

To create and deploy a JDBC connection pool:

1. In the left pane, expand the Services node.
2. Expand the JDBC node.

3. Click on Connection Pools.

   A table of existing connection pools, if any, is shown.

4. Click the Configure a new JDBC connection pool text link.

5. On the General tab, provide information about the JDBC connection pool you want to create as described in the following table.

Table 7-2 JDBC Connection Pool Configuration

Field	Description
Name	Name of the JDBC connection pool. This can be any name by which you choose to identify the pool. JDBC Data Source that uses this pool must use the exact pool name used here. For example, we can create a connection pool name for a Wireless data source called `MyWireless_POOL`.
URL	URL for the database where your data source resides. The URL is passed to the driver to create the physical database connections. For our example Oracle database, we use the following URL: `jdbc:oracle:thin:@<hostname>:1521:SID` **Note:** The required URL format varies depending on database type. See "Connection Pool URLs and Driver Names for JDBC Data Sources" on page 7-2 for a complete list of URL formats and drivers for each supported database type.
Driver Classname	Full package name of the JDBC 2-tier driver class used to create the physical connections between the WebLogic Server and the DBMS for this Connection Pool. For our Oracle example, we use the following Oracle driver classname: `oracle.jdbc.driver.OracleDriver` **Note:** Driver class names vary depending on database type. See "Connection Pool URLs and Driver Names for JDBC Data Sources" on page 7-2 for a complete list of URL formats and drivers for each supported database type.

Table 7-2 JDBC Connection Pool Configuration

Field	Description
Properties	Sets the list of properties passed to the 2-tier JDBC Driver to use when creating physical database connections. The list consists of attribute=value tags, separated by semi-colons. WLS Administration Console view will automatically reformat properties and add other details about the specified database when you click Create. The following examples are based on our Broadband, CRM, and Wireless scenario: • `user=broadband; password=broadband` • `user=crm; password=crm` • `user=wireless; password=wireless` (You can add these as comma separated attribute=value pairs as shown above or one per attribute=value pair per line separated by a return.)
Password	Optional—use only if you are implementing security. Password attribute passed to the tier-2 JDBC driver when creating physical database connections; If set, this value overrides any password defined in Properties. The value is stored in an encrypted form in the `config.xml` file and when displayed on the administration console. Use this method to avoid storing passwords in clear text in `config.xml`.

6. Click Create.

   The new JDBC connection pool you created is shown in the table.

7. In the table of connection pools, click on the name of the new JDBC connection pool you just created.

   The Configuration and Monitoring tabs for that pool are displayed.

8. Click on the Configuration tab.

9. Click on the Connections tab and set the Maximum Capacity.

   Note: This is not a required step, but to facilitate running and testing the data sources, we recommend re-setting Maximum Capacity on the connection pool to some number greater than 1. For complete information on how to create Java Database Connectivity (JDBC) Connection Pools in WebLogic Server, see JDBC (http://edocs.bea.com/wls/docs81/ConsoleHelp/jdbc.html) in the WebLogic Server documentation.

10. Click on the Targets tab.

The name of your Liquid Data server should be listed under Available Servers.

11. Select the Liquid Data server in Available and click the right arrow button to move the server into the Chosen list.
12. Click Apply.

Creating a JDBC Data Source

For complete information on how to create a JDBC data source in WebLogic Server, see JDBC (http://edocs.bea.com/wls/docs81/ConsoleHelp/jdbc.html) in the WebLogic Server documentation.

Once you have created a JDBC connection pool, the next step in configuring a Liquid Data relational database data source is to create a JDBC data source in WLS using the JDBC connection pool that you just configured.

Creating a JDBC data source consists of first creating the data source and then deploying it on a target server. You will need to configure this new data source to use the JDBC connection pool you just created.

Table 7-3 WebLogic Server JDBC Data Source Configuration

Field	Description
Name	Name of the data source. This can be any name you choose to use for the data source, such as `MyWirelessDS`.
JNDI Name	JNDI path to where this data source is bound. Applications that look up the JNDI path will get a `javax.sql.DataSource` instance that corresponds to this data source. This can be any name you choose to use for the data source, such as `MyWirelessDS`.
Pool Name	Name of the connection pool the data source is associated with. The pool name you provide here must match exactly the name of the connection pool you created in the previous task ("Creating a JDBC Connection Pool" on page 7-2). For our example, the pool name is `MyWireless_POOL`.

To create and deploy a JDBC data source:

1. In the left pane, expand the Services node.
2. Expand the JDBC node.
3. Click on Data Sources.

A table of existing data sources, if any, is shown.

4. Click the Configure a new JDBC Data Source connection pool text link.
5. On the General tab, provide information about the JDBC data source you want to create as described in Table 7-2, "JDBC Connection Pool Configuration," on page 7-3. (The fields described in the table are required—other optional fields are also displayed on this tab.)
6. Click Create.

   The new JDBC data source you created is shown in the table.
7. In the table of JDBC data sources, click on the name of the new JDBC data source you just created.

   The Configuration and Monitoring tabs for that JDBC data source are displayed.
8. Click on the Configuration tab.
9. Click on the Targets tab.

   The name of your Liquid Data server should be listed under Available Servers.
10. Select the Liquid Data server in Available and click the right arrow button to move the server into the Chosen list.
11. Click Apply.

Creating a Relational Database Data Source Description

Once you have created a JDBC connection pool and JDBC data source for the relational database, you can create a data source description that tells Liquid Data how to connect to the relational database.

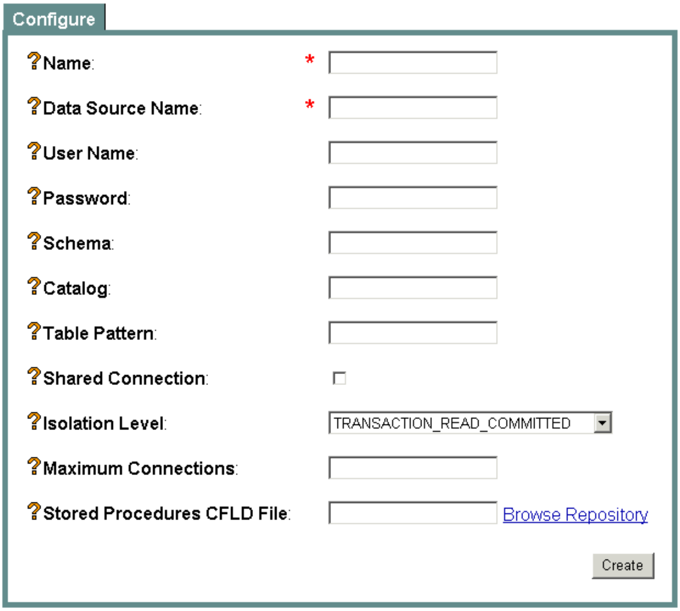
Note: You must log in with `modify` access before you can add a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To create a data source description for a relational database:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the Relational Databases tab.
5. Click the Configure a new Relational Database source description text link.

The configuration tab for creating a new relational database Liquid Data source description is displayed.

Figure 7-4 Configuring a Liquid Data Source Description for a Relational Database



Configure

?Name:	*	
?Data Source Name:	*	
?User Name:		
?Password:		
?Schema:		
?Catalog:		
?Table Pattern:		
?Shared Connection:		□
?Isolation Level:		TRANSACTION_READ_COMMITTED ▾
?Maximum Connections:		
?Stored Procedures CFLD File:		Browse Repository

6. Fill in the fields as described in the following table:

   Note: All names and values that you provide are case-sensitive.

Table 7-5 Liquid Data Relational Database Data Source Description

Field	Description
Name	Logical name of the database—it is a Liquid Data source description used to register the relational database data source with the Liquid Data server. You can choose any meaningful name.
Data Source Name	Data source name, which must match the JNDI name for the JDBC data source created in "Creating a JDBC Data Source" on page 7-5.

Table 7-5 Liquid Data Relational Database Data Source Description (Continued)

Field	Description
User Name	An optional field specifying a WebLogic user name. When this field is blank, the WebLogic user name authenticated in the application (for example, a web application that uses Liquid Data to access data) is passed down to the JDBC connection pool. When you specify a WebLogic user name in this field, the specified name is passed down to the JDBC connection pool instead of the application user name. This is useful if the JDBC connection pool has a security policy associated with it and you do not want to configure your JDBC connection pool security policies with your application user name credentials. When you use this field, only the specified WebLogic user needs access to the JDBC connection pool for database query access.
Password	An optional field specifying the WebLogic password corresponding to the user name specified in the Liquid Data Relational Data Source User Name field.

Table 7-5 Liquid Data Relational Database Data Source Description (Continued)

Field	Description
Schema	Name of the schema (or schemas) you want to use for this Liquid Data data source. While this field is not required, BEA recommends that you specify a schema for your databases. The schema will limit the scope of the schema elements available to liquid data. On some databases, if you do not specify a schema, you might also run into JDBC or database limits such as the maximum number of open cursors. If you run into those types of limit, you must either specify a schema or increase the number of cursors for the JDBC and/or database configuration. You can specify multiple schemas by entering comma-separated schema names, as in the following example which specifies both the `WIRELESS` and `BROADBAND` schemas: `WIRELESS,BROADBAND` Specifying multiple schema names allows you to join across those schema (or select from tables in both schema) in a single query. For example, if you create a data source referencing both the `WIRELESS` and `BROADBAND` schema, and then create a query that uses elements in both schema, Liquid Data generates a single query to the database server. Conversely, if you create two separate data sources, one referencing the `WIRELESS` schema and one referencing the `BROADBAND` schema, and you create a query referencing tables in both schema, Liquid Data sends separate queries to each schema. The requirements for setting the schema name vary depending on the relational database you are using: • **Oracle**—Schema name corresponds to the name of the Oracle schema, which is typically the name of an Oracle user ID. If you do not specify a schema, all of the schema available to the Oracle user ID configured in the JDBC connection pool are shown. • **PointBase**—Schema name corresponds to a database name. The schema is required for PointBase (without it, no schema elements are available to Liquid Data). • **Microsoft SQL Server**—Schema name corresponds to the *catalog* owner, such as `dbo`. Same as the database owner. Schema name must match the catalog or database owner for the database to which you are connecting. • **DB2**—Schema name corresponds to the catalog owner of the database, such as `db2admin`. (DB2 often has many databases.) • **Sybase**—Schema name corresponds to the database owner. Schema name must match the database owner for the database to which you are connecting. • **Informix**—Not needed for Informix data sources.

Table 7-5 Liquid Data Relational Database Data Source Description (Continued)

Field	Description
Catalog	Optional or required (depending on the RDBMS you are using)—Name of the catalog you want to use for this Liquid Data data source. Leave blank to use all catalogs or if the RDBMS system you are using does not support the notion of catalogs. The requirements for setting the catalog name vary depending on the RDBMS you are using: • **Oracle**—Leave blank; do not specify a catalog parameter. • **PointBase**—Leave blank; do not specify a catalog parameter. PointBase has only one catalog called PointBase. • **Microsoft SQL Server**—Catalog name is the database name. • **DB2**—Leave blank; do not specify a catalog parameter. • **Sybase**—Catalog name is the database name. • **Informix**—Leave blank; do not specify a catalog parameter. The user name specified in the JDBC Connection Pool must have sufficient privileges in the RDBMS to use this catalog. (See "Creating a JDBC Connection Pool" on page 7-2.)

Table Pattern	Optional—A pattern used to filter the tables by name.
	Special characters are:
	• _ An underscore character is used to match any single character.
	• % A percent sign is used to match of zero or more characters.
	You can also enter a comma separated list to specify multiple filter patterns. For example, the following list:
	`CUSTOMER, %PROD%, ORDERS_`
	would match the following tables:
	`CUSTOMER, PRODUCT, PRODUCTS, NEWPRODUCTS, ORDERS1, ORDERS9`
	but would not match the following tables:
	`CUSTOMERS, PRO_DUCT, ORDERS_1`

Shared Connection	Toggle to set *shared connection* on (checked) or off (cleared). • **On**—If you have a shared connection, it means that all the EJB instances share a single JDBC connection per data source. • **Off**—Without a shared connection, the Liquid Data server can use multiple JDBC connections per data source.

Table 7-5 Liquid Data Relational Database Data Source Description (Continued)

Field	Description
Isolation Level	Sets the transaction isolation level. • **On**—If Shared Connection is selected (checked), setting the transaction isolation level has no effect. The Liquid Data server always uses the JDBC default (`TRANSACTION_READ_COMMITTED`). • **Off**—If Shared Connection is not selected (cleared), you can select one of the following transaction isolation levels: `TRANSACTION_READ_UNCOMMITTED`—The transaction can view uncommitted updates from other transactions. `TRANSACTION_READ_COMMITTED`—The transaction can view only committed updates from other transactions. This is the default setting. `TRANSACTION_REPEATABLE_READ`—Once the transaction reads a subset of data, repeated reads of the same data return the same values, even if other transactions have subsequently modified the data. `TRANSACTION_SERIALIZABLE`—Simultaneously executing this transaction multiple times has the same effect as executing the transaction multiple times in a serial fashion. For information on JDBC transaction isolation levels, see “transaction-isolation” and “isolation-level” in weblogic-ejb-jar.xml Document Type Definitions (under the subheading “5.1 weblogic-ejb-jar.xml Deployment Descriptor Elements”) in the WebLogic Server documentation.
Maximum Connections	Optional—Specifies maximum number of connections the Liquid Data server can use to access this data source. The default is 0, which indicates that you are not limiting the number of connections. Has no effect if the value of Shared Connections is set to on (checked). **Note:** If the number of concurrent queries to the RDBMS data source exceeds the value in Maximum Connections (unless the value is 0, specifying no maximum), additional query executions will fail with a “No Connection Available” response.
SQL Call Description File	If you have stored procedures or you want to configure your own SQL queries as data sources you can access through Liquid Data, you must create and specify a SQL Call Description File (SCDF) that defines the stored procedures and/or SQL query. The SCDF file must reside in the `<ld_repository>/sql_calls` directory. For details on configuring access to your stored procedures and SQL queries, see Defining Stored Procedures and SQL Queries in *Building Queries and Data Views*.

7. Click Create.

The Administration Console displays the new relational database data source description in the summary table.

Note: You must configure access to this data source description, as described in "Configuring Secure Access to Data Source Descriptions" on page 6-3.

Summary of Configured Data Sources

The summary table shows a list of configured data sources of a particular type and a subset of configuration information for quick scanning. From the summary list, you can do the following:

- Find the entry for a particular data source in the table on the tab of the Liquid Data Administration Console corresponding to the data source type (for example, Relational Databases).
- Click on the data source name to modify its configuration.
- If you want to set a security policy for the data source (and security in enabled for your Liquid Data domain), click Define Security Policy or Edit Security Policy on the data source to which you want to assign security. If the object already has a policy defined, the link is labeled "Edit Security Policy"; if the object has no defined policy, the link is labeled "Define Security Policy." For more information, see "Configuring Secure Access to Data Source Descriptions" on page 6-3.
- Remove an existing data source by clicking the trash can next to it.

Note: You can also view all data sources from the All Data Sources configuration tab on the Liquid Data node in the Administration Console, as described in Chapter 6, "Viewing and Accessing All Configured Data Sources."

Modifying a Relational Database Source Description

You can changed the configured settings in a data source description for a relational database. For example, you might want to make a simple change to the row prefetch settings, or you might want to make more fundamental changes, such as changing the JDBC connection pool and data source used by the Liquid Data source, or changing the target servers or clusters in which the data source is deployed.

For most configuration changes, you will need to verify the operation of any queries that depend on the changed data source configuration. To make fundamental changes in underlying JDBC connection pools and data sources, you will also need to ensure that you set up the new JDBC connection pools and data sources first, before you re-assign the existing Liquid Data sources to them. For more information, see "Creating a JDBC Connection Pool" on page 7-2 and "Creating a JDBC Data Source" on page 7-5.

Modifying Data Source Description Settings

Note: You must log in with `modify` access before you can modify a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To make simple changes to the data source description settings for a relational database, such as the row prefetch settings:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the Relational Databases tab.

   A table of configured Liquid Data data sources is shown.
5. Click on the data source description that you want to modify.
6. Change the settings as needed.
7. Click Apply.
8. Verify the operation of any existing queries that depend on the data source configuration you just changed.

Modifying JDBC Connection Pools or JDBC Data Sources

If you are making changes to JDBC connection pools or JDBC data sources:

1. Un-deploy the JDBC connection pool or JDBC data source you are modifying by selecting the pool or data source you want to modify, clicking on the associated Targets tab, moving the server from Chosen to Available, and clicking Apply.
2. Make any changes to JDBC connection pools or data sources as needed (or create new ones) and re-deploy these. For more information, see "Creating a JDBC Connection Pool" on page 7-2 and "Creating a JDBC Data Source" on page 7-5.

Removing a Relational Database Source Description

You can remove a data source description that you no longer need. Removing a data source description does not remove the actual relational database to which it refers, nor does it remove the associated WebLogic Server JDBC Data Source or JDBC connection pool.

Note: You must log in with `modify` access before you can remove a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To remove a data source description for a relational database:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the Relational Databases tab.

   A table of configured Liquid Data data sources is shown.
5. Find the data source that you want to remove and click the trash can next to it.
6. When prompted, click Yes to confirm removal.

   The Administration Console removes the selected data source description.

CHAPTER 8

Configuring Access to XML Files

Before a BEA Liquid Data for WebLogic query can access data in an XML file, the XML file must be configured as a Liquid Data data source. The XML file must also be added to the xml_files folder of the Liquid Data Server repository, as described in "Uploading Files to the Server Repository" on page 18-8. Once configured according to the instructions in this chapter, XML files with data source descriptions will show up as data sources available for use in any EJB client, such as the Data View Builder, that connects to this server.

The following sections are included:

- Creating an XML File Data Source Description
- Summary of Configured Data Sources
- Modifying an XML File Data Source Description
- Removing an XML File Data Source Description

Creating an XML File Data Source Description

To access an XML file from Liquid Data, you must first create a data source description that tells Liquid Data how to find the XML file.

Note: You must log in with `modify` access before you can create a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

Table 8-1 Liquid Data XML Flat File Data Source Description

Field	Description	Required
Name	Logical name of the XML file.	Yes.
Data File	XML data file. One of the following formats: • Name of the file that resides in the Liquid Data server repository. Enter the file name, or click Browse Repository to select it. If you have not done so already, save the XML file you want to use as a Liquid Data data source in the server repository. For more information, see Chapter 18, "Managing the Liquid Data Server Repository." • File URL, such as: `file:///D:/bea7a/weblogic700/liquiddata/docs/data.xml` • HTTP URL, such as: `http://bea.com/data.xml`	Yes, unless you check the Dynamic Data Source button.
Schema	Schema for the XML file in the server repository. Enter the file name, or click Browse Repository to select it.	Yes.
Namespace URI	Identifies the target namespace of the schema file. Example: `urn:schemas-bea-com:ld-cptSample`	Optional but recommended. If used, Schema Root Element Name must also be supplied.

Table 8-1 Liquid Data XML Flat File Data Source Description (Continued)

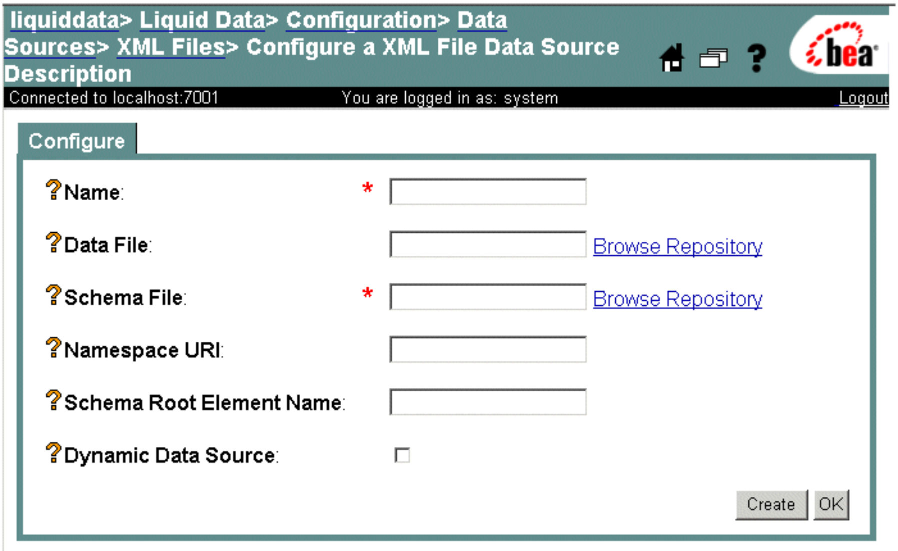
Field	Description	Required
Schema Root Element Name	Identifies a unique root element in the schema file. Many schemas have only a single root. In cases where there are multiple root elements, only elements under the identified root will available as an XML data source. For example, the sample schema `CustomerOrderReport` described in the Liquid Data Getting Started document has only a single root, `CustomerOrder`.	Yes.
Dynamic Data Source	Check this box if the data source is to be supplied at query runtime. If it is checked, you must specify a second parameter for the `xf:document` function in the XQuery with a file name. The name has the same rules as those stated for the data file.	Optional.

To create a data source description for an XML file:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the XML File tab.
5. Click the Configure a new XML file source description text link.

   The configuration tab for creating a new XML file Liquid Data source description is displayed.

Figure 8-2 Configuring a Liquid Data Source Description for an XML File


liquiddata> Liquid Data> Configuration> Data Sources> XML Files> Configure a XML File Data Source Description
bea
Connected to localhost:7001
You are logged in as: system
Logout
Configure
?Name: *
?Data File:
Browse Repository
?Schema File: *
Browse Repository
?Namespace URI:
?Schema Root Element Name:
?Dynamic Data Source: □
Create
OK

6. Fill in the fields as detailed in Table 8-1, "Liquid Data XML Flat File Data Source Description," on page 8-2.

7. Click Create.

   The Administration Console displays the new XML file data source description in the summary table.

Note: You must configure access to this data source description, as described in "Configuring Secure Access to Data Source Descriptions" on page 6-3. In addition, you can configure access to any underlying XML files in the repository, as described in "Configuring Secure Access to Items in the Server Repository" on page 18-13.

Summary of Configured Data Sources

The summary table shows a list of configured data sources of a particular type and a subset of configuration information for quick scanning. From the summary list, you can do the following:

- Navigate to the configuration for a particular data source by clicking on it in the table.
- If you want to set a security policy for the data source (and security in enabled for your Liquid Data domain), click Define Security Policy or Edit Security Policy on the data source to which you want to assign security. If the object already has a policy defined, the link is labeled "Edit Security Policy"; if the object has no defined policy, the link is labeled "Define Security Policy."

For more information, see "Configuring Secure Access to Data Source Descriptions" on page 6-3.

- Remove an existing data source by clicking the trash can next to it.
- If there is a known configuration problem for a data source, a red mark appears in the Status column of the summary table. For details on the data source status, see "Checking the Status of Liquid Data Resources" on page 21-1.

Note: You can also view all data sources from the All Data Sources configuration tab on the Liquid Data node in the Administration Console, as described in Chapter 6, "Viewing and Accessing All Configured Data Sources."

Modifying an XML File Data Source Description

You can modify an existing XML file data source description.

Note: You must log in with `modify` access before you can modify a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To modify an existing data source description for an XML file:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the XML File tab.

   A table of configured Liquid Data XML files is shown.
5. Click on the XML file for which you want to modify the source description.
6. Change the settings as needed.
7. Click Apply.
8. Verify the operation of any existing queries that depend on the data source configuration you just changed.

Removing an XML File Data Source Description

You can remove a data source description that you no longer need. Removing a data source description does not remove the actual XML file to which it refers. To explicitly remove the XML file from the repository, see "Deleting Folders and Files in the Server Repository" on page 18-13.

Note: You must log in with `modify` access before you can remove a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To remove a data source description for an XML file:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the XML File tab.

   A table of configured Liquid Data XML data sources is shown.
4. Find the XML file that you want to remove and click the trash can next to it.
5. When prompted, click Yes to confirm removal.

   The Administration Console removes the selected data source description.

CHAPTER 9

Configuring Access to Delimited Files

This chapter describes how to configure access to delimited files in Liquid Data. Delimited files are typically exported from a database, spreadsheet, or an application like a spreadsheet. A common format for delimited files is a comma separated values (CSV) file. This chapter contains the following sections:

- Using Delimited Files in Liquid Data
- Configuring Delimited Files

Using Delimited Files in Liquid Data

Delimited files are text files that contain a pre-defined character as a data separator. You can create data sources in Liquid Data that access these files for use in queries. You can then combine this data with any other data sources by creating queries that use the delimited file data source.

Creating Delimited Files From an Application

You can create delimited file with any pre-defined character as a data separator. Many applications, including database and spreadsheet applications, provide built in features to generate comma separated value (CSV) files. You can then use the CSV files (or files with any separator character) as data files to query against with Liquid Data.

Typically, you can use the Save As feature in a database or spreadsheet application (Microsoft Excel, for example) to save a file as a delimited or CSV file.

Delimited File Configuration Field Descriptions

The following table describes the fields in the Configure Delimited Files Data Source Description screen of the Liquid Data Administration Console.

Table 9-1 Delimited File Configuration Fields

Field	Description
Name	The name of the Delimited File data source in the repository.
Data File	Name of a delimited data file in the repository. Enter the delimited file name or click Browse Repository to select it.
Schema	Optional. Identifies the name of the schema associated with the delimited file. You can find the `.xsd` file in the *`LDrepository`*`/schema` folder. Either enter the schema file name or click Browse Repository to browse the schema folder.
Separator	The character used to separate the fields in a delimited file. The default is a comma character (,).
Remove Quotes	Specifies if the quotation marks (") in the data file are removed when the data is read. Check this box unless you want the quotation marks to display as part of the data.

Table 9-1 Delimited File Configuration Fields (Continued)

Field	Description
Has Header	Specifies if the delimited data file has a header row. When this box is checked, the first row is treated as a header row and not a data row. If there is no schema file, then the values in the header row are used for the schema elements of the Delimited File Data Source.
Dynamic Data Source	Check this box if the data source is to be supplied at query runtime. If it is checked, you must specify a second parameter for the `xf:document` function in the XQuery with a file name. The name has the same rules as those stated for the data file.

Configuring Delimited Files

Perform the following steps to configure a delimited file data source description.

1. Create a delimited file and save it in the `delimited_files` folder of the Liquid Data repository. For details about the Liquid Data repository, see "Managing the Liquid Data Server Repository" on page 18-1.

2. Start the Administration Console (for details, see "Starting the Administration Console" on page 4-2).

3. In the left pane of the Administration Console, expand the node for your domain and click the Liquid Data node.

4. In the right pane, click the Data Sources tab in the Liquid Data Administration console.

5. In the right pane, navigate to the Delimited Files tab (below the Data Sources tab).

6. Click the Configure a New Delimited File Data Source Description link (or to modify an existing one, click the name of the data source).

7. Enter values for the Name and Data File fields. optionally, enter values for the other fields. For a description of the fields, see "Delimited File Configuration Field Descriptions" on page 9-2.

Configuring Access to Web Services

Before a BEA Liquid Data for WebLogic query can access data in a Web Service, the Web Service must be configured as a Liquid Data data source. Once configured according to the instructions in this chapter, Web Services with data source descriptions will show up as data sources available for use in any EJB client, such as the Data View Builder, that connects to this server.

The following sections are included:

- Creating a Web Service Data Source Description
- Summary of Configured Data Sources
- Modifying a Web Service Data Source Description
- Removing a Web Service Data Source Description

Creating a Web Service Data Source Description

To access a Web Service from Liquid Data, you must first create a data source description that tells Liquid Data how to connect to the Web Service.

Note: You must log in with `modify` access before you can create a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

Table 10-1 Liquid Data Web Service Data Source Description Configuration

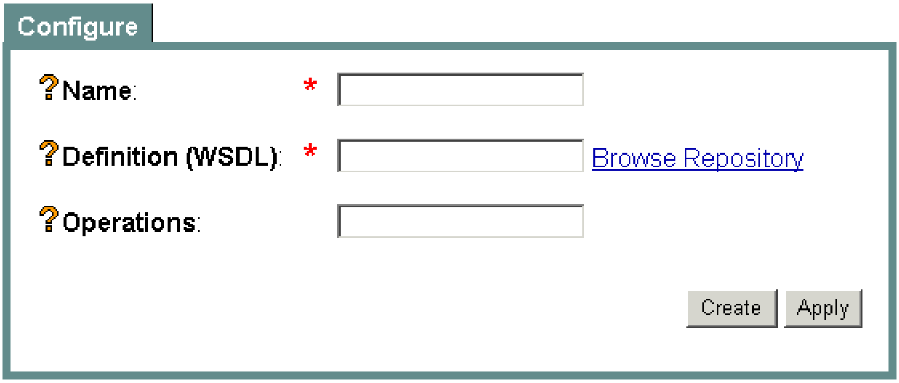
Field	Description
Name	Logical name of the Web Service. Web Service data source names must start with an alphabetic character (a-z or A-Z).
Definition (WSDL)	Uniform Resource Locator (URI) of the Web Service definition. This can point to a local WSDL file in the repository (enter the URI or click Browse Repository to select the file) or to a network accessible shared drive.
Operations	Optional—Filter of Web Service operations to make available to Liquid Data queries. Multiple filters are separated by commas. For example: `getSalesPrices,getSalesDiscount`

To create a data source description for a Web Service:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the Web Services tab.
5. Click the Configure a new Web Service source description text link.

   The configuration tab for creating a new Web Service Liquid Data source description is displayed.

Figure 10-2 Configuring a Liquid Data Source Description for a Web Service



6. Fill in the fields described in Table 10-1, "Liquid Data Web Service Data Source Description Configuration," on page 10-2.

7. Click Create.

   The Administration Console displays the new Web Service data source description in the summary table.

Note: You must configure access to this data source description, as described in "Configuring Secure Access to Data Source Descriptions" on page 6-3.

Summary of Configured Data Sources

The summary table shows a list of configured data sources of a particular type and a subset of configuration information for quick scanning. From the summary list, you can do the following:

- Navigate to the configuration for a particular data source by clicking on it in the table.
- If you want to set a security policy for the data source (and security in enabled for your Liquid Data domain), click Define Security Policy or Edit Security Policy on the data source to which you want to assign security. If the object already has a policy defined, the link is labeled "Edit Security Policy"; if the object has no defined policy, the link is labeled "Define Security Policy." For more information, see "Configuring Secure Access to Data Source Descriptions" on page 6-3.
- Remove an existing data source by clicking the trash can next to it.

Note: You can also view all data sources from the All Data Sources configuration tab on the Liquid Data node in the Administration Console, as described in Chapter 6, "Viewing and Accessing All Configured Data Sources."

Modifying a Web Service Data Source Description

Note: You must log in with `modify` access before you can modify a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To modify the settings on an existing Web Service data source description:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the Web Services tab.

   A table of configured Liquid Data data sources is shown.
5. Click on the data source you want to modify.
6. Change the settings as needed.
7. Click Apply.
8. Verify the operation of any existing queries that depend on the data source configuration you just changed.

Removing a Web Service Data Source Description

You can remove a data source description that you no longer need. Removing a data source description does not remove the actual Web Service to which it refers.

Note: You must log in with `modify` access before you can delete a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To remove a Web Service data source description from the Liquid Data Server:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the Web Services tab.

A table of configured Liquid Data data sources is shown.

5. Find the data source that you want to remove and click the trash can next to it.
6. When prompted, click Yes to confirm removal.

   The Administration Console removes the selected data source description.

Note: Removing a Web Service data source description from Liquid Data does not remove the underlying Web Service.

Configuring Access to Application Views

Before a BEA Liquid Data for WebLogic query can access data in an application view, the application view must be configured as a Liquid Data data source. Once configured according to the instructions in this chapter, application views with data source descriptions will show up as data sources available for use in any EJB client, such as the Data View Builder, that connects to this server.

Note: Liquid Data can use the services of an application view but not its events. Application view events have no effect on Liquid Data.

Before you can add an application view as a Liquid Data data source, you must first do the following:

- Install and configure WebLogic Integration.
- In the server startup file for the Liquid Data server, you must add the path to the Application Integration `wlai-client.jar` file (such as `%`*`WLI_HOME`*`%\lib\wlai-client.jar`) to the *end* of the Liquid Data classpath (`LDCLASSPATH`). For instructions, see "Deployment Tasks" in the *Deployment Guide*.
- Deploy each adapter for which you will define application views.
- Use the WebLogic Integration Application View Console to define the application views you want to use as Liquid Data data sources, as described in "Defining an Application View" in *Using Application Integration* in the WebLogic Integration documentation.

The rest of this section describes how to configure the application view as a Liquid Data data source via the WebLogic Administration Console, and how to modify or remove a Liquid Data application view data source description.

For complete information on how to use WLI Application Integration—including application views, see Using Application Integration (http://edocs.bea.com/wli/docs81/aiuser/index.html) in the WebLogic Integration documentation.

The following sections are included here:

- Adding Liquid Data to a WebLogic Platform or WebLogic Integration Domain
- Starting the Liquid Data Server in the WebLogic Platform or WebLogic Integration Domain
- Configuring an Application View Data Source Description
  - Creating an Application View Data Source Description
  - Summary of Configured Data Sources
  - Modifying an Application View Data Source Description
  - Removing an Application View Data Source Description

Adding Liquid Data to a WebLogic Platform or WebLogic Integration Domain

Before you can start using Liquid Data with Application Integration, you must have a WebLogic Platform or WebLogic Integration domain configured to work with Liquid Data. There are two major steps to create such a domain: (1) creating the WebLogic Platform or WebLogic Integration domain, and (2) adding Liquid Data to it.

If you are starting from scratch, you must first create a WebLogic Platform or WebLogic Integration domain using the WebLogic Platform Configuration Wizard and the WebLogic Integration (WLI) configuration templates. For instructions, see "Creating Domains Using the Configuration Wizard" in the WebLogic Platform documentation.

Once you have WebLogic Platform or WebLogic Integration domain, you need to add Liquid Data to that domain by running the Liquid Data utility for deploying Liquid Data to a new domain. For information on how to run this utility, see "Adding Liquid Data to an Existing Domain" on page 2-2 in Chapter 2, "Creating Liquid Data Domains."

Starting the Liquid Data Server in the WebLogic Platform or WebLogic Integration Domain

Once you have configured a WebLogic Platform or WebLogic Integration domain and added Liquid Data to it, you are ready to start the Application Integration server. In a command window, navigate to the domain directory (`BEA_HOME/user_projects/domain_name`) in which you configured Application Integration with Liquid Data and run the `startWebLogic.cmd` (on Windows) or `startWebLogic.sh` (on UNIX) to start the server.

Defining an Application View Using the WebLogic Integration Application View Console

The first step in creating an application view data source for Liquid Data is to define an application view in the WebLogic Integration Application View Console. For detailed instructions on defining an application view, see "Defining an Application View" in *Using Application Integration* in the WebLogic Integration documentation.

Note: The Liquid Data installation includes an option to install the Application Integration component. If you chose this option during installation, the WebLogic Integration Application View Console and related software is installed on your system and accessible. For information about Liquid Data installation options see the Liquid Data *Installation Guide*.

Configuring an Application View Data Source Description

Once you have defined an application view using the WebLogic Integration Application View Console as described in "Defining an Application View Using the WebLogic Integration Application View Console" on page 11-3, you are ready to add the application view as a Liquid Data data source. You do this by using the WebLogic Administration Console to configure a Liquid Data source description for the application view you just defined. The process from this point on is similar to configuring any other Liquid Data data source.

To continue with Liquid Data Server application view configuration, start the WebLogic Server Administration Console for your Liquid Data server if you have not done so already. (See "Starting the Administration Console" on page 4-2. To connect to a local server running on your machine use the following URL in the address bar of a Web browser `http://localhost:7001/console/`).

Creating an Application View Data Source Description

To access an application view from Liquid Data, you must first create a data source description that tells Liquid Data how to find the application view.

Notes: You must log in with `modify` access before you can create a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

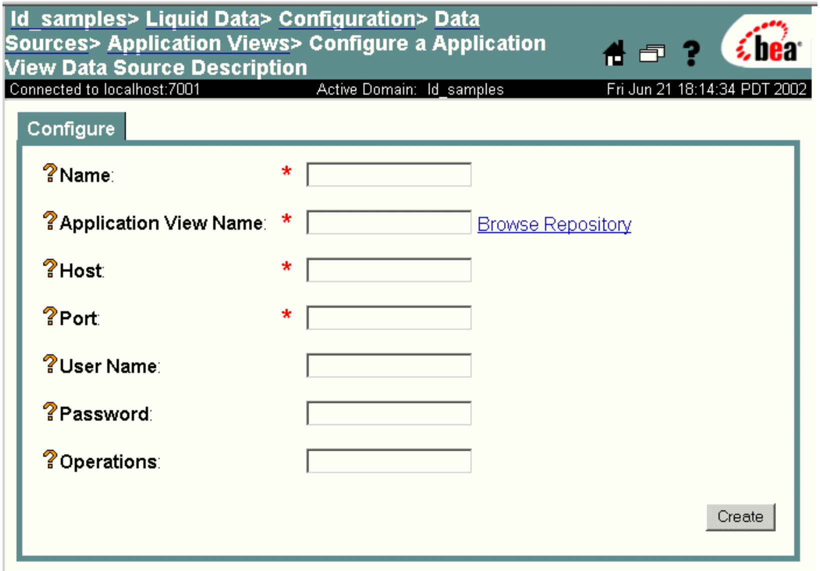
Liquid Data can use the services of an application view but not its events. Application view events have no effect on Liquid Data.

To create a data source description for an application view:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the Application View tab.
5. Click the Configure a new Application View source description text link.

   The configuration tab for creating a new application view Liquid Data source description is displayed.

Figure 11-1 Configuring a Liquid Data Source Description for an Application View



ld_samples> Liquid Data> Configuration> Data Sources> Application Views> Configure a Application View Data Source Description

Connected to localhost:7001 Active Domain: ld_samples Fri Jun 21 18:14:34 PDT 2002

Configure

?Name: *

?Application View Name: * Browse Repository

?Host: *

?Port: *

?User Name:

?Password:

?Operations:

Create

6. Fill in the fields as detailed in Table 11-2.

7. Click Create.

   The Administration Console displays the new application view data source description in the summary table.

Note: You must configure access to this data source description, as described in "Configuring Secure Access to Data Source Descriptions" on page 6-3.

Table 11-2 Liquid Data Application View Data Source Description

Field	Description
Name	Logical name of the Liquid Data application view data source. You can use any name you want. For use in Liquid Data, the name must start with an alphabetic character (a-z or A-Z).
Application View Name	Name of the application view as defined in the WebLogic Integration Application View Console.

Table 11-2 Liquid Data Application View Data Source Description (Continued)

Field	Description
Host	Host name or IP address of the system on which the application view is running. For more information about this field and the remaining fields in this table, see "Defining an Application View Using the WebLogic Integration Application View Console" on page 11-3.
Port	Listen port used by the system on which the application view is running.
User Name	WebLogic Server username used for the application view. This identifies a user who has access to the application view instance.
Password	WebLogic Server password used for the application view. This is the password for a user who has access to the application view instance.
Operations	Optional—Filter of the application view. You can specify that only a subset of the operations provided by the application view be available to Liquid Data.

Summary of Configured Data Sources

The summary table shows a list of configured data sources of a particular type and a subset of configuration information for quick scanning. From the summary list, you can do the following:

- Navigate to the configuration for a particular data source by clicking on it in the table.
- If you want to set a security policy for the data source (and security in enabled for your Liquid Data domain), click Define Security Policy or Edit Security Policy on the data source to which you want to assign security. If the object already has a policy defined, the link is labeled "Edit Security Policy"; if the object has no defined policy, the link is labeled "Define Security Policy." For more information, see "Configuring Secure Access to Data Source Descriptions" on page 6-3.
- Remove an existing data source by clicking the trash can next to it.

Note: You can also view all data sources from the All Data Sources configuration tab on the Liquid Data node in the Administration Console, as described in Chapter 6, "Viewing and Accessing All Configured Data Sources."

Modifying an Application View Data Source Description

You can modify an existing application view data source description. If you want to modify the actual application view definition in Application Integration, you need to do this through the WebLogic Integration Application View Console as described in "Defining an Application View Using the WebLogic Integration Application View Console" on page 11-3.

Note: You must log in with `modify` access before you can modify a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To modify the data source description for an application view:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.
4. Click the Application View tab.

   A table of configured Liquid Data data sources is shown.
5. Click on the data source you want to modify.
6. Change the settings as needed.
7. Click Apply.
8. Verify the operation of any existing queries that depend on the data source configuration you just changed.

Removing an Application View Data Source Description

You can remove a data source description that you no longer need. Removing a data source description does not remove the actual application view to which it refers.

Note: You must log in with `modify` access before you can remove a data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To remove a data source description for an application view:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Data Sources tab.

4. Click the Application View tab.

   A table of configured Liquid Data application view data sources is shown.

5. Find the application view that you want to remove and click the trash can next to it.

6. When prompted, click Yes to confirm removal.

   The Administration Console removes the selected data source description.

Configuring Access to Data Views

Before a BEA Liquid Data for WebLogic query can access data in a data view, the data view must be configured as a Liquid Data data source. Data views are derived from stored queries. Only one data view can be created from a stored query. You can also deploy data views directly from the Data View Builder. For instructions on how to create data views, see *Building Queries and Data Views* and "To Create a Data View from a Stored Query" on page 16-6.

Once configured, data views with data source descriptions will show up as data sources available for use in any EJB client, such as the Data View Builder, that connects to this server. For more information, see "Using Data Views as Data Sources" in *Building Queries and Data Views*.

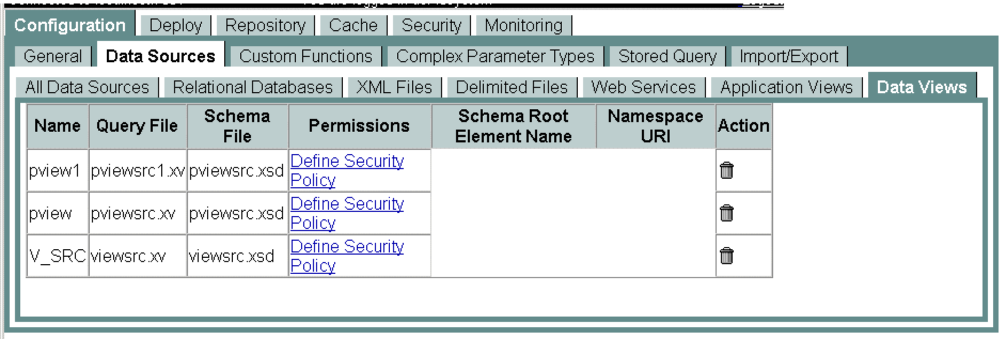
The following sections are included:

- Data View Data Sources Page of the Administration Console
- Creating a Data View Data Source Description
- Deleting a Data View Data Source Description
- Defining Security Policy for a Data View

Data View Data Sources Page of the Administration Console

The Data View tab of the Liquid Data Administration Console displays a list of data views configured in Liquid Data. You configure the data views from the Stored Queries tab of the Liquid Data Administration Console.

Figure 12-1 Data Views Tab of the Liquid Data Administration Console



Name	Query File	Schema File	Permissions	Schema Root Element Name	Namespace URI	Action
pview1	pviewsrc1.xv	pviewsrc.xsd	Define Security Policy			🗑
pview	pviewsrc.xv	pviewsrc.xsd	Define Security Policy			🗑
V_SRC	viewsrc.xv	viewsrc.xsd	Define Security Policy			🗑

From the Data View tab, you can perform the following:

- If you want to set a security policy for the data source (and security in enabled for your Liquid Data domain), click Define Security Policy or Edit Security Policy on the data source to which you want to assign security. If the object already has a policy defined, the link is labeled “Edit Security Policy”; if the object has no defined policy, the link is labeled “Define Security Policy.” For more information, see “Configuring Secure Access to Data Source Descriptions” on page 6-3.
- Remove an existing data source by clicking the trash can next to it.

Note: You can also view all data sources from the All Data Sources configuration tab on the Liquid Data node in the Administration Console, as described in Chapter 6, “Viewing and Accessing All Configured Data Sources.”

Creating a Data View Data Source Description

To access a data view as a data source from Liquid Data, you must first create a data source description. The data source description provides Liquid Data with the necessary information to access the query defined in the data view and use it as a data source for other queries.

Note: You can create and deploy a Data View directly from the Data View Builder instead of going to the Administration Console. For details, see *Building Queries and Data Views*.

You create new data view data sources from the stored query tab of the Liquid Data Administration Console. To create a data view data source description, see "To Create a Data View from a Stored Query" on page 16-6.

Deleting a Data View Data Source Description

You can delete a data source description that you no longer need. Deleting a data source description does not remove the stored query to which it refers; it removes only the data view description from the repository.

Note: You must have access to the console and `modify` security permissions on the data view to delete the data view data source description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To remove a data source description for a data view:

1. Start the Administration Console (for details, see "Starting the Administration Console" on page 4-2).
2. In the left pane of the Administration Console, expand the node for your domain and click the Liquid Data node.
3. In the right pane, click the Data Sources tab.
4. Click the Data Views tab.

   A table of configured Liquid Data data views is shown (see Figure 12-1).
5. Find the data view that you want to remove and click the trash can in the Action column (the last column in the table).
6. When prompted, click Yes to confirm removal.

   The Administration Console removes the selected data source description and deletes the file from the Liquid Data repository.

Defining Security Policy for a Data View

You can define a security policy for each data source from the Data Views tab of the Liquid Data Administration Console.

Perform the following steps to define a security policy on a Liquid Data data view data source.

1. Start the Administration Console (for details, see "Starting the Administration Console" on page 4-2).
2. In the left pane of the Administration Console, expand the node for your domain and click the Liquid Data node.
3. Navigate to the Liquid Data —> Data Sources —> Data Views tab of the Liquid Data Administration Console.
4. Click Define Security Policy or Edit Security Policy on the data view to which you want to assign security. If the data view already has a policy defined, the link is labeled "Edit Security Policy"; if it no defined policy, the link is labeled "Define Security Policy."
5. Assign a security policy to the data view, as described in "Assigning Security Policies to Liquid Data Objects" on page 19-13.
6. Click Apply on the security policy screen.

CHAPTER 13

Deploying Liquid Data Components

This chapter describes how you can deploy BEA Liquid Data for WebLogic components using the Deploy tab on the Liquid Data node in the Administration Console. It contains the following sections:

- Liquid Data Components to Deploy
- Navigating to the Deploy Tab

Administrators can use the Deploy tab to shut down applications (in the form of JAR, WAR, or EAR files) on a WebLogic Server without interrupting other running applications. Administrators can also upgrade an application by undeploying it, substituting an updated JAR, WAR, or EAR file, and then redeploying the updated application on target servers.

Note: This chapter describes how to deploy core Liquid Data software components only. In most if not all cases, however, the same results can be achieved automatically when creating a Liquid Data domain or extending an existing domain for Liquid Data through the BEA WebLogic Configuration Wizard. There may be additional tasks required to deploy other Liquid Data components. See "Deployment Tasks" in the *Deployment Guide* for details.

Liquid Data Components to Deploy

The `LDS.ear` file contains the following deployable components:

Table 13-1 Contents of the LDS.ear file

Component	Description	Deployment Target
`ejb_query.jar`	EJB that contains all query-related classes, including the Query Processor and stateless session bean. For more information, see the Liquid Data Javadoc.	For a multi-node and clustered deployment, deploy to each Managed Server.
`ejb_qbc.jar`	EJB for Data View Builder to obtain configuration information from the Liquid Data Server.	For a multi-node and clustered deployment, deploy to each Managed Server.
`XMediator.war`	Initializes the Liquid Data Server.	For a multi-node and clustered deployment, deploy to each Managed Server.
`ldconsole.war`	Liquid Data configuration tabs that appear in the WebLogic Server Administration Console.	Always deploy to an Administration Server.
`cacheEjb.jar`	EJB that manages results caching for stored queries that are configured for results caching. For more information, see "Configuring the Query Results Cache" on page 22-1.	For a multi-node and clustered deployment, deploy to each Managed Server.
`ldcacheListener.war`	Listener application that listens to notifications from the MBean for changes to the global cache status (enabled or disabled), changes to a stored query that is cached, and changes to the cache policy for a stored query.	For a multi-node and clustered deployment, deploy to each Managed Server.

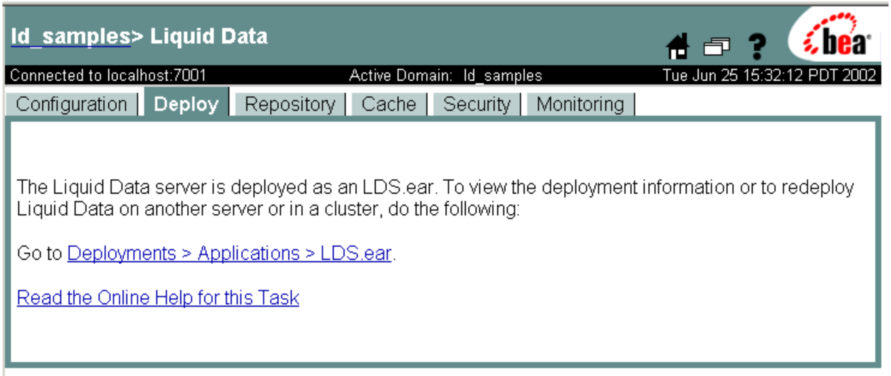
Navigating to the Deploy Tab

To navigate to the Deploy tab:

1. In the left pane of the Administration Console, click the Liquid Data node.
2. In the right pane, click the Deploy tab.

The Administration Console displays the Liquid Data Deploy tab.

Figure 13-2 Deploy Tab on the Liquid Data Node in the Administration Console



ld_samples> Liquid Data

Connected to localhost:7001 Active Domain: ld_samples Tue Jun 25 15:32:12 PDT 2002

The Liquid Data server is deployed as an LDS.ear. To view the deployment information or to redeploy Liquid Data on another server or in a cluster, do the following:

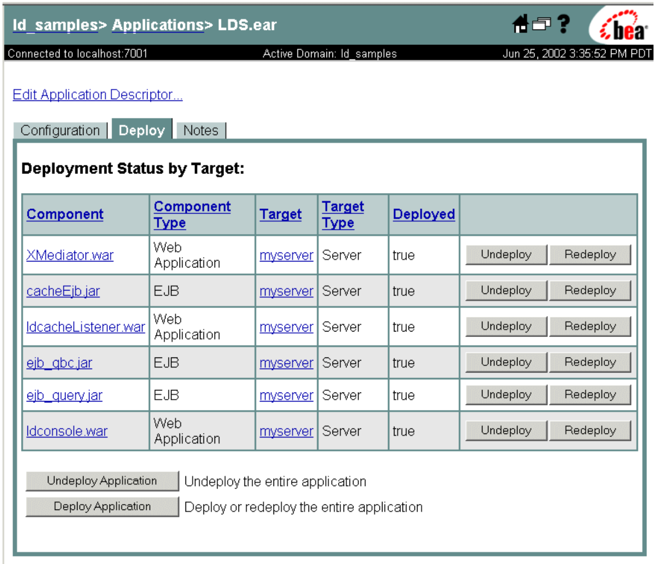
Go to Deployments > Applications > LDS.ear.

Read the Online Help for this Task

3. Click on the link to go to Deployments > Applications > LDS.ear.

   The Administration Console displays the WebLogic Server Deploy tab showing the deployment status of the Liquid Data enterprise archive file (`LDS.ear`) as deployed in WebLogic Server.

Figure 13-3 WebLogic Deploy Tab



ld_samples> Applications> LDS.ear

Connected to localhost:7001 Active Domain: ld_samples Jun 25, 2002 3:35:52 PM PDT

Edit Application Descriptor...

Deployment Status by Target:

Component	Component Type	Target	Target Type	Deployed	
XMediator.war	Web Application	myserver	Server	true	Undeploy Redeploy
cacheEjb.jar	EJB	myserver	Server	true	Undeploy Redeploy
ldcacheListener.war	Web Application	myserver	Server	true	Undeploy Redeploy
ejb_qbc.jar	EJB	myserver	Server	true	Undeploy Redeploy
ejb_query.jar	EJB	myserver	Server	true	Undeploy Redeploy
ldconsole.war	Web Application	myserver	Server	true	Undeploy Redeploy

Undeploy Application — Undeploy the entire application

Deploy Application — Deploy or redeploy the entire application

4. Select target servers and deploy or undeploy as needed. For instructions, see “Deployment Tasks” in the *Deployment Guide*.

Configuring Access to Custom Functions

Custom functions are user-defined functions that performed specialized tasks. Before a BEA Liquid Data for WebLogic query can access a custom function, the custom function must be configured on the Custom Functions tab on the Liquid Data node in the Administration Console. This chapter describes how to configure access to custom functions. It includes the following sections:

- About Custom Functions
- Administration Tasks for Custom Functions
- Creating a Custom Function Description
- Summary of Configured Custom Function Groups
- Configuring Secure Access to Custom Function Descriptions
- Modifying a Custom Function Description
- Removing a Custom Function Description

For additional information, see "Using Custom Functions" in *Invoking Queries Programmatically*.

About Custom Functions

The Data View Builder provides a set of standard functions for use in creating data views and queries using various types of joins and mappings. You can also extend the Data View Builder by creating custom functions to perform specialized tasks.

This section includes the following parts:

- Use Cases for Custom Functions
- Components of Custom Functions

Use Cases for Custom Functions

Custom functions allow you to perform specialized operations that are not available in standard functions. There are many possible use cases for custom functions in Liquid Data. The following list provides just a few examples of what custom functions can do:

- Process a column in a database that contains data requiring specialized interpretation. For example, a `name_title` column might contain numeric codes (1, 2, 3, and so on) that represent text (Mr., Mrs., Ms., Dr., and so on) rather than the text itself. A custom function could be created to decode the data in the column and return the text instead.
- Calculate a special mathematical formula, equation, or operation.
- Invoke stored procedures on a JDBC data source via the query EJB.

Components of Custom Functions

A custom function is implemented in Java code and declared in a custom functions library definition (CFLD) file. For detailed information about these tasks, see "Using Custom Functions" in *Invoking Queries Programmatically*.

Once implemented and declared in the Liquid Data Server repository, a *custom function description* must be created for each custom function. A custom function description defines the following information:

- logical name of the custom function as declared in the CFLD file
- name of the CFLD file in which the custom function is declared

Once configured according to the instructions in this section, custom functions with custom function descriptions will show up as functions available for use in any Data View Builder client that connects to this server.

Administration Tasks for Custom Functions

To configure custom functions, administrators perform the following tasks:

1. Add the JAR file containing the custom function implementation to the `custom_lib` folder in the Liquid Data Server repository, as described in "Uploading Files to the Server Repository" on page 18-8.

2. Add the path to the JAR file in the Custom Functions Classpath field on the General tab on the Liquid Data node, as described in Chapter 5, "Configuring Liquid Data Server Settings."

3. Add the CFLD file containing the custom function declaration to the `custom_functions` folder in the Liquid Data Server repository, as described in "Uploading Files to the Server Repository" on page 18-8.

4. For each custom function, create a custom function description, as described in "Creating a Custom Function Description" on page 14-3.

5. Assign security policies to the custom function description, as described in "Configuring Secure Access to Custom Function Descriptions" on page 14-5, and also assign security policies to the JAR and CFLD files in the Liquid Data Server repository, as described in "Configuring Secure Access to Items in the Server Repository" on page 18-13.

Creating a Custom Function Description

Note: You must log in with `modify` access before you can create a custom function description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To create a custom function description for a group of custom functions:

1. Make sure that the JAR file containing custom function implementations resides in the `custom_lib` folder in the Liquid Data Server repository, and that the CFLD file declaring the custom functions resides in the `custom_functions` folder in the Liquid Data Server repository.

2. In the left pane, click the Liquid Data node.

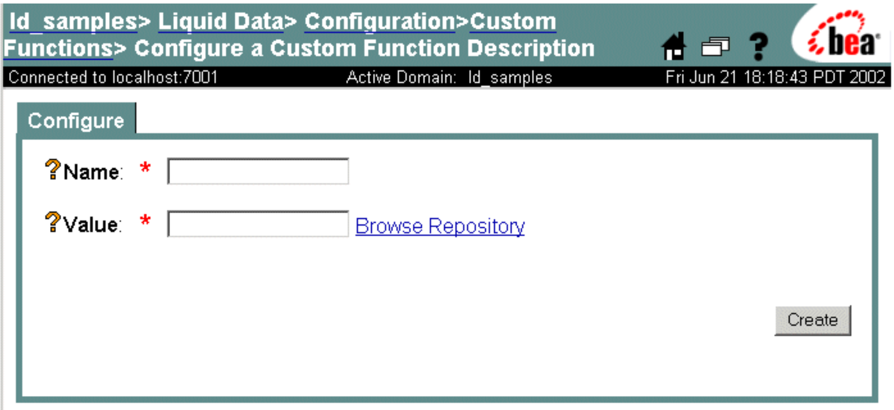
3. In the right pane, click the Configuration tab.

4. Click the Custom Functions tab.

5. Click the Configure a new Custom Function description text link.

   The Administration Console displays the configuration tab for creating a new custom function description.

Figure 14-1 Configuring a Custom Function Description



ld_samples> Liquid Data> Configuration>Custom Functions> Configure a Custom Function Description

Connected to localhost:7001 Active Domain: ld_samples Fri Jun 21 18:18:43 PDT 2002

6. Enter the information described in the following table:

Table 14-2 Custom Function Description Information

Field	Description
Name	Logical name of the group of custom functions declared in the custom functions library definition (CFLD) file. Custom function names must start with an alphabetic character (a-z or A-Z).
Value	File name of the CFLD file that declares this custom function in an XML format. This file usually resides in the `custom_functions` folder in the Liquid Data Server repository, which is described in "Server Repository File System Hierarchy" on page 18-4. Either type the CFLD file name or click Browse Repository to browse the `custom_functions` folder and select it.

7. Click Create.

   The Administration Console displays the new custom function description in the summary table.

Summary of Configured Custom Function Groups

The summary table on the Custom Functions tab on the Liquid Data node shows a list of custom function groups that have been configured with custom function descriptions on the current server. From the summary list, you can perform the following tasks:

- Navigate to the custom function description for a particular custom function group by clicking on it in the table.
- If you want to set a security policy for the custom function (and security in enabled for your Liquid Data domain), click Define Security Policy or Edit Security Policy on the data source to which you want to assign security. If the object already has a policy defined, the link is labeled "Edit Security Policy"; if the object has no defined policy, the link is labeled "Define Security Policy." For more information, see "Configuring Secure Access to Data Source Descriptions" on page 6-3.
- Remove an existing custom function group from the Liquid Data function library by clicking the trash can at the far right of the selected function.

Configuring Secure Access to Custom Function Descriptions

Note: You must log in with `modify` access before you can assign security policies to a custom function description. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

You can configure security for each custom function description using security policies. You need to assign `execute` permissions to users who are authorized to execute queries that use particular custom functions. For more information about Liquid Data security, see "Defining Liquid Data Roles and Groups" on page 19-2

Note: In the Repository tab on the Liquid Data node, you can assign `modify` and `read` access to CFLD files and JAR files associated with custom functions. The security policies assigned in the Custom Functions tab determine whether a user can execute a query in which a custom function is used. The security policies in the Repository tab determine whether the user logged into the Administration Console can modify or read the CFLD or JAR files in the repository.

To Assign Security Policies to a Custom Function Description

Perform the following steps to assign security policies to a data source:

1. Open the Liquid Data Administration Console and click the Custom Functions tab.

2. For the custom function to which you want to assign a security policy, click Define Security Policy or Edit Security Policy on the data source to which you want to assign security. If the object already has a policy defined, the link is labeled “Edit Security Policy”; if the object has no defined policy, the link is labeled “Define Security Policy.”

3. Assign security policies as needed to the custom functions. For details, see “Assigning Security Policies to Liquid Data Objects” on page 19-13.

Modifying a Custom Function Description

Note: You must log in with `modify` access before you can modify a custom function description. For more information, see “Defining Liquid Data Roles and Groups” on page 19-2.

You can modify a custom function description to change the logical name of the custom function group or the name of the CFLD file in which the custom function group is declared.

To modify a custom function description:

1. In the left pane, click the Liquid Data node.

2. In the right pane, click the Configuration tab.

3. Click the Custom Functions tab.

   The Administration Console displays a table of custom function descriptions.

4. Click on the custom function description that you want to modify.

5. Change the settings as needed.

6. Click Apply.

Removing a Custom Function Description

Note: You must log in with `modify` access before you can remove a custom function description. For more information, see “Defining Liquid Data Roles and Groups” on page 19-2.

You can remove a custom function description that you no longer need. To remove a custom function description:

1. In the left pane, click the Liquid Data node.

2. In the right pane, click the Configuration tab.

3. Click the Function Library tab.

The Administration Console displays a table of custom function descriptions.

4. Find the custom function description that you want to remove and click on the trash can on the far right column for that function.
5. When prompted, click Yes to confirm removal.

   The Administration Console removes the selected custom function description.

Note: Removing a Liquid Data custom function description does not remove the JAR or CFLD files associated with the custom function group from your system.To remove these files from the Liquid Data Server repository, see "Deleting Folders and Files in the Server Repository" on page 18-13.

CHAPTER 15

Configuring Access to Complex Parameter Types

Complex parameter types (CPTs) are user-defined data structures that enable the execution of Liquid Data queries against dynamic content. Such content is known as *runtime source*, *data stream*, *real-time data, in-flight XML documents*, and so forth.

This chapter describes how to configure access to complex parameter types in the Administration Console. It includes the following sections:

- Creating a Complex Parameter Type Description
- Managing Complex Parameter Types

You will find other important aspects of using CPTs described in:

- Using Complex Parameter Types in *Building Queries and Data Views*
- Setting Complex Parameter Types in the *Application Developer's Guide*

Creating a Complex Parameter Type Description

Before you can use a BEA Liquid Data for WebLogic query with a CPT in the Data View Builder or programmatically, you must:

- Identify a schema that represents the data that will be retrieved using the complex parameter type. For details, see Using Complex Parameter Types in Queries in *Building Queries and Data Views*.
- Configure the CPT data source through the Administration Console.

Table 15-1 Complex Parameter Type Description Information

Field	Description	Required
Alias	Sets the logical name for the complex parameter type being defined.	Yes.
Schema	Identifies the schema associated with the complex parameter type. You can find the `.xsd` file in the *<LDrepository>* schema folder, described in "Server Repository File System Hierarchy" on page 18-4. Either enter the schema file name or click Browse Repository to browse the `schema` folder.	Yes.
Namespace UR	Identifies the target namespace of the schema file. Example: `urn:schemas-bea-com:ld-cptSample`	Optional but recommended. If used, Schema Root Element Name must also be supplied.
Schema Root Element Name	Identifies a unique root element in the schema file. Many schemas only have a single root. In cases where there are multiple root elements, the CPT will only use the identified root element and its sub-elements. For example, the sample schema `CustomerOrderReport` described in the Liquid Data Getting Started document has only a single root, `CustomerOrder`.	Optional but recommended. If used, Namespace URI must also be supplied.

To create a complex parameter type description using the Administration Console:

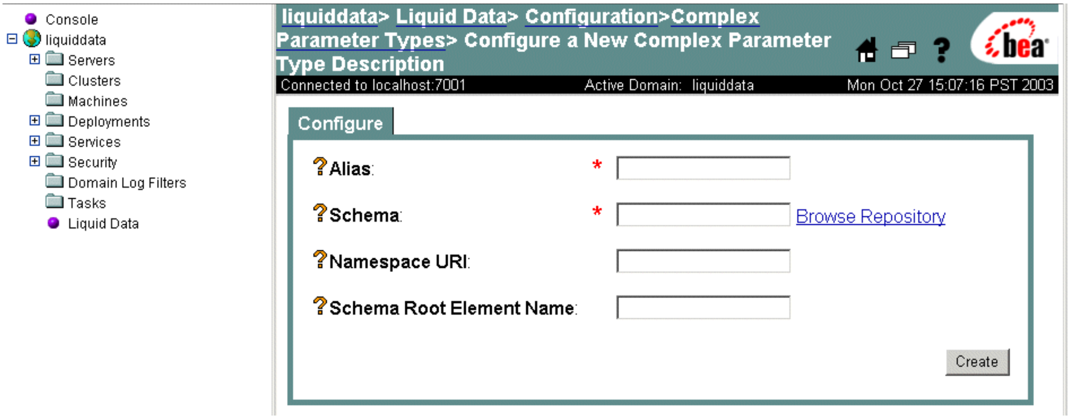
1. In the left pane, click the Liquid Data node.

2. In the right pane, click the Configuration tab (probably already selected).

3. Click the Complex Parameter Types tab.

4. Click the Configure a New Complex Parameter Type Description link.

Figure 15-2 Configuring a Complex Parameter Type Description



liquiddata> Liquid Data> Configuration>Complex Parameter Types> Configure a New Complex Parameter Type Description

Connected to localhost:7001 Active Domain: liquiddata Mon Oct 27 15:07:16 PST 2003

Alias: *

Schema: *

Namespace URI:

Schema Root Element Name:

5. Enter the information described in Table 15-1, “Complex Parameter Type Description Information,” on page 15-2.

6. Click Create.

The new description appears in the Administration Console summary table.

Managing Complex Parameter Types

When you click the Complex Parameter Types tab, you will see a list of CPTs that are configured on the current server in the summary table. From the summary list, you can perform the following tasks:

- Navigate to and optionally edit a particular CPT description by clicking on its name.
- Delete an existing CPT definition from the Liquid Data function library by clicking the trash can at the far right of the selected definition.

Modifying a Complex Parameter Type Configuration

Follow these steps to modify a CPT configuration:

1. In the left pane, click the Liquid Data node.

2. In the right pane, click the Configuration tab (probably already selected).
3. Click the Complex Parameter Types tab.

   The table of complex parameter type descriptions appears.
4. Click the CPT description alias name that you want to modify.
5. Change entries as needed.
6. Click Apply.

Removing a Complex Parameter Type Configuration

You can remove a CPT description that you no longer need. To do so:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab (probably already selected).
3. Click the Complex Parameter Type tab.
4. Find the CPT description that you want to remove and click the trash can icon on the far right column for that description.
5. When prompted, click Yes to confirm removal.

Note: Removing a Liquid Data CPT description does not remove the associated schema. For information on removing such files from the Liquid Data Server repository, see "Deleting Folders and Files in the Server Repository" on page 18-13.

CHAPTER 16

Configuring Stored Queries

This chapter describes how to configure stored queries in Liquid Data. It contains the following sections:

- Stored Queries in Liquid Data
- Stored Query Configuration Tasks
  - To Configure (Deploy) a Stored Query
  - To Create a Data View from a Stored Query
  - To Create a Web Service from a Stored Query
  - To Delete a Stored Query Configuration

You can also create, test, and deploy stored queries directly from the Data View Builder. For instructions on how to deploy a stored query from the Data View Builder, see “Deploying a Query” in *Building Queries and Data Views*.

Stored Queries in Liquid Data

A stored query in Liquid Data is a file in the Liquid Data repository that contains the XQuery text to submit a query to the Liquid Data server. Stored queries are stored in the following location:

LDRepository/stored_queries

where LDRepository is the root-level directory of the Liquid Data repository. The location of the Liquid Data repository is specified in the General tab of the Liquid Data Administration Console.

You can deploy a stored query either from the Liquid Data Administration Console or from the Data View Builder. For details on using the Data View Builder to deploy stored queries, see *Building Queries and Data Views*. For details on deploying a stored query from the Liquid Data Administration Console, see "Stored Query Configuration Tasks" on page 16-5.

This section describes stored queries and includes the following subsections:

- Stored Queries Tab of Liquid Data Administration Console
- Available for Liquid Data Control or EJB API
- Stored Query Configuration Field Descriptions

Stored Queries Tab of Liquid Data Administration Console

You use the Stored Query tab of the Liquid Data Administration Console to configure stored queries. The Stored Query tab displays all of the queries stored in the repository and allows you to configure the following for each query:

- A schema file for the query
- Generate a Web Service from the stored query (schema file required)
- Generate a Data View from the stored query (schema file required)
- Specify a security policy for the stored query (security must be enabled)
- Delete the stored query definition

If the stored_queries directory of the repository has subdirectories, any queries in a subdirectory is displayed in the stored query tab with a dot (.) between each subdirectory and the query name. For example, if you have the following file in your Liquid Data repository:

LDRepository/stored_queries/rtl/electronics/orders.xq

The entry in the Stored Query tab appears as follows:

```
rtl.electronics.orders.xq
```

If you configure a security policy on stored queries that resides in subdirectories of the `stored_queries` repository folder, they inherit any security policy that is configured from the folder(s) in which they reside. Stored queries do not inherit security policy from a security resource group. In the previous example, the stored query `orders.xq` inherits security policy from the `rtl` repository folder and from the `electronics` repository folder. For details on configuring a security policy on a repository folder, see "Configuring Secure Access to Items in the Server Repository" on page 18-13 and "Assigning Security Policies to Liquid Data Objects" on page 19-13.

Note: Because you can generate Web Services from stored queries, stored queries must have names that conform to the XML Schema specification. For example, the names must begin with an alphabetic character (letter)—not a number, and they must not have any special characters such as hyphens. For more information, see "Naming Conventions for Stored Queries" and the W3C XML Schema document at http://www.w3.org/XML/Schema.

Available for Liquid Data Control or EJB API

You can access stored queries programmatically with the Liquid Data EJB API. If you are using the Liquid Data Control in WebLogic Workshop, the queries must have a schema file associated with them. The Liquid Data Control requires the schema file because it generates an `XMLBean` Java classes corresponding to the schema definitions. The `XMLBean` compiler requires the schema for the file. The generated `XMLBean` Java classes allow an interface to the data, which makes it easier to display data from a Liquid Data query in an application.

Stored Query Configuration Field Descriptions

The following table describes the fields in the Configure Stored Query screen of the Liquid Data Administration Console.

Table 16-1 Stored Query Configuration Fields

Field	Description
Name	The name of the stored query saved in the repository. The name includes any subdirectories with a dot (.) between the directory hierarchies. For an example of the dot notation, see "Stored Queries in Liquid Data" on page 16-2.
Schema	Identifies the name of the schema file associated with the query. You can find the `.xsd` file in the *LDrepository*/`schema` folder. Either enter the schema file name or click Browse Repository to browse the schema folder.
Namespace URI	Identifies the full target namespace of the Stored Query schema file. Example: `urn:schemas-bea-com:myQuery` Note: If a namespace is provided, the schema root element name must also be supplied. If it is not provided, Liquid Data determines the namespace from the schema file.
Schema Root Element Name	Identifies a unique root element in the schema file. Many schemas only have one root. In cases where there are more than one root, only sub-elements of the identified root will be part of the schema used in the stored query. You must specify the root element if there is more than one root element to ensure that Liquid Data uses the correct element for the schema. Note: If the schema root element name is entered, a namespace must also be provided.

Stored Query Configuration Tasks

This section includes procedures for the following tasks related to stored queries:

- To Configure (Deploy) a Stored Query
- To Create a Data View from a Stored Query
- To Create a Web Service from a Stored Query
- To Delete a Stored Query Configuration

Note: You can also create, test, and deploy stored queries directly from the Data View Builder. For instructions on how to deploy a stored query from the Data View Builder, see "Deploying a Query" in *Building Queries and Data Views*.

To Configure (Deploy) a Stored Query

Perform the following steps to configure a stored query in Liquid Data.

1. Create a query and save it in the `stored_queries` folder of the Liquid Data repository. For details about the Liquid Data repository, see "Managing the Liquid Data Server Repository" on page 18-1.

2. Start the Administration Console (for details, see "Starting the Administration Console" on page 4-2).

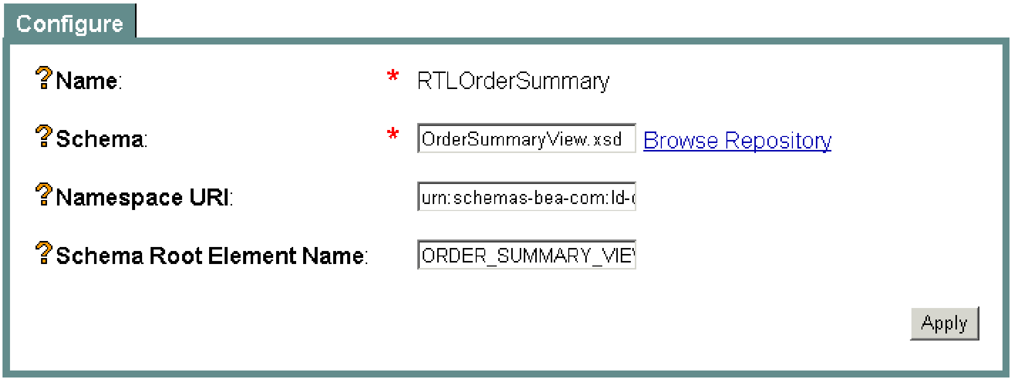
3. In the left pane of the Administration Console, expand the node for your domain and click the Liquid Data node.

4. In the right pane, navigate to the Stored Query tab.

   A table containing links to all of the query files in the `stored_queries` folder of the Liquid Data repository appears.

5. Click the link corresponding to the query you want to configure.

Figure 16-2 Edit Stored Query Screen



Configure

Name: * RTLOrderSummary

Schema: * OrderSummaryView.xsd Browse Repository

Namespace URI: urn:schemas-bea-com:ld-(

Schema Root Element Name: ORDER_SUMMARY_VIE\

6. Enter a schema file and (optionally) a Namespace URI and a Schema Root Element. For a description of the fields, see "Stored Query Configuration Field Descriptions" on page 16-4.

7. Click Apply to save your configuration changes.

To Create a Data View from a Stored Query

Perform the following steps to use the Administration Console to create a Data View from a stored query.

Note: You can create and deploy a Data View directly from the Data View Builder instead of going to the Administration Console. For details, see *Building Queries and Data Views*.

1. Start the Administration Console (for details, see "Starting the Administration Console" on page 4-2).

2. In the left pane of the Administration Console, expand the node for your domain and click the Liquid Data node.

3. In the right pane, navigate to the Stored Query tab.

4. The stored query must have a schema configured (and, optionally, a namespace and root element node) before you can create a data view from it. For details on adding a schema file to the stored query configuration, see "To Configure (Deploy) a Stored Query" on page 16-5.

5. Find the entry for your stored query and click the Create Data View link.

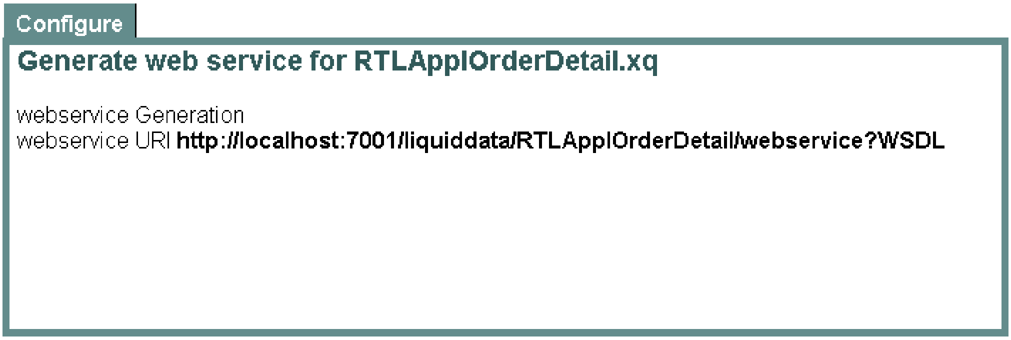
6. Enter a name for the Data View on the Configure a Data View Data Source screen and click Create.

To Create a Web Service from a Stored Query

Perform the following steps to create a Web Service from a stored query:

1. Start the Administration Console (for details, see "Starting the Administration Console" on page 4-2).
2. In the left pane of the Administration Console, expand the node for your domain and click the Liquid Data node.
3. In the right pane, navigate to the Stored Query tab.
4. The stored query must have a schema configured (and, optionally, a namespace and root element node) before you can create a web service from it. For details on adding a schema file to the stored query configuration, see "To Configure (Deploy) a Stored Query" on page 16-5.
5. Find the entry for your stored query and click the Generate Web Service link. A screen similar to the following appears when the web service generation is complete.

Figure 16-3 Successful Web Service Generation



Generate web service for RTLApplOrderDetail.xq

webservice Generation
webservice URI http://localhost:7001/liquiddata/RTLApplOrderDetail/webservice?WSDL

To Delete a Stored Query Configuration

When you delete the stored query definition (by clicking the trash can icon), the Administration Console deletes the association between the stored query file and the schema definition. If you have created a Data View and a Web Service from the query, those objects are not deleted when you click the trash can. To delete those objects, you must go to the Data View and Web Services tabs of the Liquid Data Administration Console.

Perform the following steps to delete a stored query configuration:

1. Start the Administration Console (for details, see "Starting the Administration Console" on page 4-2).
2. In the left pane of the Administration Console, expand the node for your domain and click the Liquid Data node.
3. In the right pane, navigate to the Stored Query tab.
4. Find the entry for your stored query and click the trash can icon.

The definition for the stored query is deleted, but the stored query file remains in the repository. If you click the link for the stored query, the Configure Stored Query screen appears and you can reconfigure it.

Importing and Exporting Liquid Data Configurations

This chapter describes how to copy BEA Liquid Data for WebLogic configurations between Liquid Data servers. It contains the following sections:

- About Liquid Data Configurations
- Navigating to the Import/Export Tab
- Exporting a Liquid Data Configuration
- Importing a Liquid Data Configuration

The exporting and importing process transfers only a portion of the total Liquid Data server configuration. If you want to copy a complete server configuration to another server, see “Copying a Server Configuration to Another Server” in “Deployment Tasks” in the *Deployment Guide*.

About Liquid Data Configurations

This section describes what does and does not get imported or exported in Liquid Data configurations. It contains the following sections:

- What Liquid Data Imports and Exports
- What Liquid Data Does Not Import or Export

What Liquid Data Imports and Exports

A Liquid Data configuration consists of the information described in the following table. A Liquid Data import or export results in all of these settings being imported or exported, respectively:

Table 17-1 Liquid Data Configuration Information

Type of Information	Description
Liquid Data server configuration	General settings for the Liquid Data server and the query engine. For more information on server configuration, see Chapter 5, "Configuring Liquid Data Server Settings." **Note:** The name of the server repository is not exported.
Data source descriptions	Data source descriptions that contain the information needed to connect to particular data sources used in Liquid Data queries. Each Liquid Data server instance must have its own set of data source descriptions. For more information on data source descriptions, see Chapter 6, "Viewing and Accessing All Configured Data Sources."
Results cache settings	Results cache settings, which include the Caching checkbox on the General tab on the Liquid Data node and cache policy settings associated with stored queries on the Cache tab on the Liquid Data node. For more information, see Chapter 22, "Configuring the Query Results Cache."
Custom function descriptions	Custom function descriptions for user-defined functions added to the Liquid Data function library. For more information, see Chapter 14, "Configuring Access to Custom Functions."

Note: For information about what is not included, see "What Liquid Data Does Not Import or Export" on page 17-3.

Rather than entering all of this configuration information manually on each server, you can simply copy a full Liquid Data configuration from one server to another. To copy a Liquid Data configuration, you export the configuration from one Liquid Data server to a file (in XML format), and then import that file on every Liquid Data server where you want to copy it. For example, you can copy the Liquid Data configuration on a development server to a Liquid Data server deployed in a production environment.

What Liquid Data Does Not Import or Export

This Liquid Data import/export feature handles only Liquid Data specific configuration information. This section describes what is *not* included in the import/export.

WebLogic Server Specific Configuration Information

The Liquid Data import/export process does not include WebLogic Server specific configurations defined in the `config.xml` file such as JDBC connection pools, JDBC data sources, or Compatibility Security information. To transfer this configuration information, you will need to either reconfigure these settings via the Administration Console on the new server, or you must save and copy relevant entries in the original WebLogic Server `config.xml` file to the `config.xml` file on the new server. For more information about distributing this information, see "Deployment Tasks" in the *Deployment Guide*.

Files Added to the Liquid Data Server Repository

The Liquid Data import/export process does not include files that have been added to the repository, such as target schema, XML data files, JAR files for custom function libraries, and so on. If you are copying a configuration from one server to another and you want to make the same files accessible in the new Liquid Data server repository, you need to do the following:

1. Perform the import process on the target server.
2. On the source server, users must explicitly download the files to a temporary location using the Repository tab on the Liquid Data node in the Administration Console, as described in "Downloading Files From the Server Repository" on page 18-7.
3. On the target server, upload the files you want in the repository using the Repository tab on the Liquid Data node in the Administration Console, as described in "Uploading Files to the Server Repository" on page 18-8.

Repository Name

The Liquid Data import/export process does not include the name of the server repository.

File Swap Configuration

The Liquid Data import/export process does not include the settings that control how Liquid Data handles file swapping for stored queries. If the Large Results flag is selected for a query, then Liquid Data uses swap files to temporarily store results on disk. You must manually configure file swapping—you cannot import these settings.

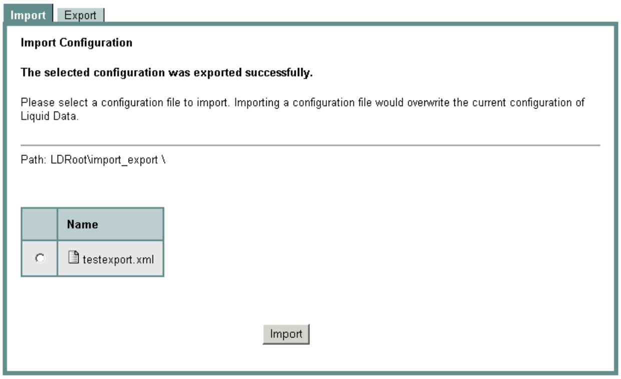
Navigating to the Import/Export Tab

To navigate to the Import / Export Tab on the Liquid Data node:

1. In the left pane of the Administration Console, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Import/Export tab.

   The Administration Console displays the Import/Export tab.

Figure 17-2 Import /Export Tab on the Liquid Data Node



Import Configuration

The selected configuration was exported successfully.

Please select a configuration file to import. Importing a configuration file would overwrite the current configuration of Liquid Data.

Path: LDRoot\import_export \

	Name
○	testexport.xml

Table 17-3 Tabs on the Import/Export Tab

Tab	Description
Import tab	Used to import a Liquid Data configuration, as described in "Importing a Liquid Data Configuration" on page 17-6.
Export tab	Used to export a Liquid Data configuration to a file, as described in "Exporting a Liquid Data Configuration" on page 17-4.

Exporting a Liquid Data Configuration

To copy a Liquid Data configuration to other Liquid Data servers, you must first create the export file that you will subsequently import into the other Liquid Data servers.

Note: You must be logged in with `modify` access to create the target directory and to create a file in that directory.

Table 17-4 Export Configuration Information

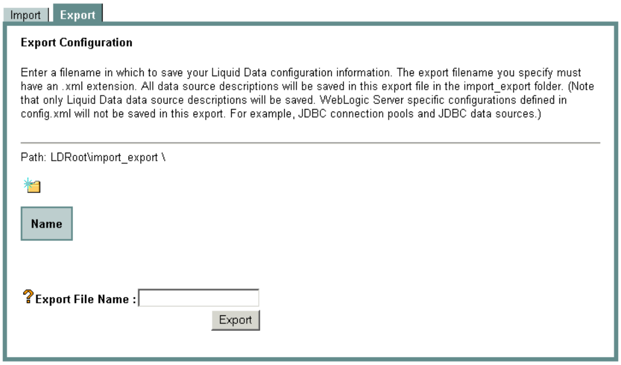
Field	Description
Path	Shows the file system path to the `import_export` folder in the repository root directory, which is the target folder in which Liquid Data creates the export file. For more information about the repository root directory, see Chapter 5, "Configuring Liquid Data Server Settings."
Export File Name	Name of the export file. You must specify a name that is valid on the target file system. Liquid Data automatically assigns an XML extension to the file name.

To export a Liquid Data configuration to a file:

1. On the source Liquid Data server, navigate to the Import / Export tab, as described in "Navigating to the Import/Export Tab" on page 17-4.
2. In the Import / Export tab, click Export.

   The Administration Console displays the Export tab.

Figure 17-5 Export Tab on the Liquid Data Node



Export Configuration

Enter a filename in which to save your Liquid Data configuration information. The export filename you specify must have an .xml extension. All data source descriptions will be saved in this export file in the import_export folder. (Note that only Liquid Data data source descriptions will be saved. WebLogic Server specific configurations defined in config.xml will not be saved in this export. For example, JDBC connection pools and JDBC data sources.)

Path: LDRoot\import_export \

Export File Name :

The Export tab includes the information shown in Table 17-4, "Export Configuration Information," on page 17-5.

3. Navigate the file hierarchy, if needed, and select or create the sub-folder in which you want to create the export file.

   Note: To simplify this process, consider saving to a shared volume to which any target Liquid Data servers have access.

4. Enter the file name of the export file.

5. Click Export.

Liquid Data exports the Liquid Data configuration to the named export file in the `import_export` folder (or a sub-folder).

Importing a Liquid Data Configuration

After you have exported a Liquid Data configuration to a file, you can import it into any other Liquid Data server. The import process is additive for new items and existing items are replaced.

Note: Before you import a Liquid Data configuration, you must have configured the repository root directory on the target server according to the instructions in "Configuring Server Settings" on page 5-2.

When importing a Liquid Data configuration, the Administration Console:

- Adds new data source descriptions and custom function descriptions to the target server.
- Replaces Liquid Data server settings on the new server with imported settings.
- If the source file contains any data source descriptions that have the same name as those already defined on the target Liquid Data server, the Administration Console replaces the data source description on the target server with the information in the source file.

If you want the new repository to include files stored in the previous repository, you need to explicitly upload them according to the instructions in "Uploading Files to the Server Repository" on page 18-8.

To import a Liquid Data configuration:

1. Complete the following tasks on the target Liquid Data server, depending on the type of data sources required:

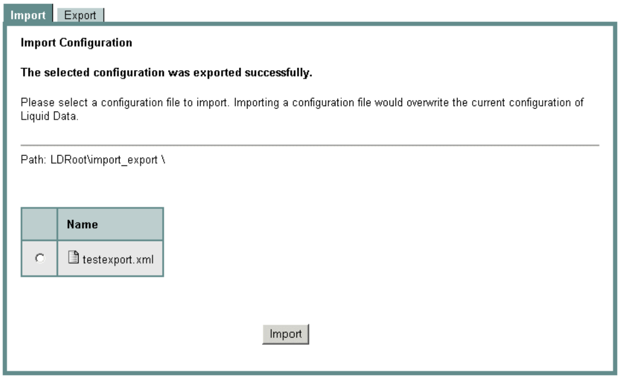
   – For relational databases, you need to configure JDBC connectivity, as described in "Creating a JDBC Connection Pool" on page 7-2. As an alternative to using the Administration Console, you can copy the relevant sections from the `config.xml` file in the source Liquid Data server and paste them into the `config.xml` file on the target Liquid Data server.

- For XML files, Web Services, and data views, you need to copy the repository to the target server, as described in “Copying a Server Configuration to Another Server” in “Deployment Tasks” in the *Deployment Guide*.
- For application views, if Liquid Data serves as the Application Integration server, then you need to reconfigure the application view using the WebLogic Integration Console, as described in “Defining an Application View Using the WebLogic Integration Application View Console” on page 11-3.

2. Upload the export file to the `import_export` folder (or a sub-folder) in the repository, as described in “Uploading Files to the Server Repository” on page 18-8.
3. On the target Liquid Data server, navigate to the Import / Export tab, as described in “Navigating to the Import/Export Tab” on page 17-4.
4. In the Import / Export tab, click Import.

   The Administration Console displays the Import tab.

Figure 17-6 Import Tab on the Liquid Data Node



Import Configuration

The selected configuration was exported successfully.

Please select a configuration file to import. Importing a configuration file would overwrite the current configuration of Liquid Data.

Path: LDRoot\import_export \

	Name
○	testexport.xml

5. Select the source file that you want to import.
6. Click Import.

   Liquid Data imports the configuration settings from the selected import file.

Managing the Liquid Data Server Repository

This chapter describes how to manage the BEA Liquid Data for WebLogic repository (also called the *server repository*). It contains the following sections:

- About the Liquid Data Server Repository
- Navigating to the Repository Tab
- Browsing the Server Repository
- Downloading Files From the Server Repository
- Uploading Files to the Server Repository
- Creating Sub-Folders
- Working with Folders and Files in the Server Repository
- Configuring the Results Cache for Stored Queries

You use the Administration Console to configure and manage the server repository. You must log into the Administration Console with sufficient permissions to perform the necessary operations in the file system on which the server repository resides. For more information, see Chapter 19, "Security in Liquid Data."

About the Liquid Data Server Repository

This section describes the server repository. It contains the following parts:

- Contents and Organization of the Server Repository
- Server Repository Location
- Server Repository File System Hierarchy
- Considerations for Evolving the Repository

Contents and Organization of the Server Repository

The server repository is the central location for storing and sharing the following Liquid Data information:

- Stored queries
- Data views
- XML files
- Source and target schemas
- Web Service WSDL files
- Generated Web Services
- Custom function libraries
- Delimited files
- SQL stored procedures and other SQL queries (SQL Calls)

The server repository provides a file system structure that organizes this information by category. Information is stored in separate folders in various formats. For example, stored queries are saved as `XQ` files in the `stored_queries` folder. You use the Administration Console to manage these folders and files, as well as to configure the server repository location.

Warning: Use the Repository tab on the Liquid Data node in the Administration Console—*not* file system commands or tools—to manage folders and files in the server repository.

Server Repository Location

By default, the server repository resides on a shared file system of the host Liquid Data server in the following location:

```
domainRootDir/repositoryRootDir
```

where

- *domainRootDir* is the directory in which your Liquid Data domain files are located
- *repositoryRootDir* is the root directory of the server repository

For example, the Liquid Data sample server repository (installed with a full Liquid Data installation) is located in the following directory:

```
BEA_HOME/weblogic81/samples/domains/liquiddata/ldrepository
```

If you use the domain configuration wizard to create a Liquid Data and WebLogic Platform domain, the default repository directory created is as follows:

```
BEA_HOME/user_projects/domains/platform/domainName/ldrepository
```

You use the General tab on the Liquid Data node in the Administration Console to configure the root directory of the server repository. You can specify a relative path (relative to the current domain directory) or a fully-qualified path. If you specify a location that has an existing server repository, the existing server repository is *not* overwritten. For more information, see Chapter 5, "Configuring Liquid Data Server Settings."

You configure only one server repository per Liquid Data deployment. The server repository must reside on a shared volume so that others can access it. In a clustered environment, all managed Liquid Data servers must be configured to point to the same server repository on a shared volume, such as on the local file system of the Administration Server host machine. For more information, see "Clustered Deployments" in "Deployment Tasks" in the *Deployment Guide*.

Server Repository File System Hierarchy

The server repository root directory contains the following folders:

Table 18-1 File System Hierarchy of the Server Repository

Folder	Contents
`custom_functions`	Custom functions library definition files (`CFLD`) containing declarations of custom functions in an XML format. For more information, see Using Custom Functions in the *Application Developer's Guide* and Chapter 14, "Configuring Access to Custom Functions."
`custom_lib`	Java Archive (JAR) files containing the Java implementations of custom functions. For more information, see Using Custom Functions in the *Application Developer's Guide* and Chapter 14, "Configuring Access to Custom Functions."
`data_views`	Stored data view (`XV`) files created using the Data View Builder. For more information, see Chapter 12, "Configuring Access to Data Views" and Designing Queries in *Building Queries and Data Views*.
`delimited_files`	Delimited files such as comma separated value (CSV) files. For more information, see Chapter 9, "Configuring Access to Delimited Files."
`dtds`	Not supported. Source document type definition (`DTD`) files. Source DTD files are associated with the XML data files stored in the `xml_files` folder. For more information, see Chapter 8, "Configuring Access to XML Files."
`import_export`	Exported Liquid Data configuration files. For more information, see Chapter 17, "Importing and Exporting Liquid Data Configurations."
`schemas`	Source and target schema (`XSD`) files. Source schema files are associated with the XML data files stored in the `xml_files` folder. For more information, see Chapter 8, "Configuring Access to XML Files."
`sql_calls`	SQL Call Description Files. These files configure access to stored procedures and other SQL queries in a relational data source. For more information, see the stored procedure chapter of *Building Queries and Data Views*.
`stored_queries`	Stored query (`XQ`) files created using the Data View Builder. For more information, see Designing Queries in *Building Queries and Data Views*.
`web_services`	Web Services Description Language (`WSDL`) files for Web Services used as data sources. For more information, see Chapter 10, "Configuring Access to Web Services."

Table 18-1 File System Hierarchy of the Server Repository (Continued)

Folder	Contents
`web_services_gen`	Application archive (EAR) files of Web Services that have been published through Liquid Data. For more information, see Chapter 23, "Generating and Publishing Web Services."
`xml_files`	XML data files used as data sources for views and queries. For more information, see Chapter 8, "Configuring Access to XML Files."

Considerations for Evolving the Repository

The server repository uses the underlying file system of the host machine. The server repository does not provide advanced features, however, such as file locking mechanisms or version control.

In a shared development environment, therefore, consider the implications of deleting, moving, or renaming files to which others or the Liquid Data Server require access. If possible, make server repository changes during idle periods to avoid file contention problems. In addition, consider implementing a third-party source control system to provide locking and version control for repository folders and files.

When deploying Liquid Data in a production environment, you can add items to the server repository without interrupting the run-time state of the system.

In general, the best approach is to populate and refine the server repository in a development environment, create a staging environment for testing and, when the repository is stable, then switch the staging server from development to production mode.

In a clustered environment, all managed Liquid Data servers must be configured to point to the same server repository on a shared volume, such as on the local file system of the Administration Server host machine. For more information, see "Clustered Deployments" in "Deployment Tasks" in the *Deployment Guide*.

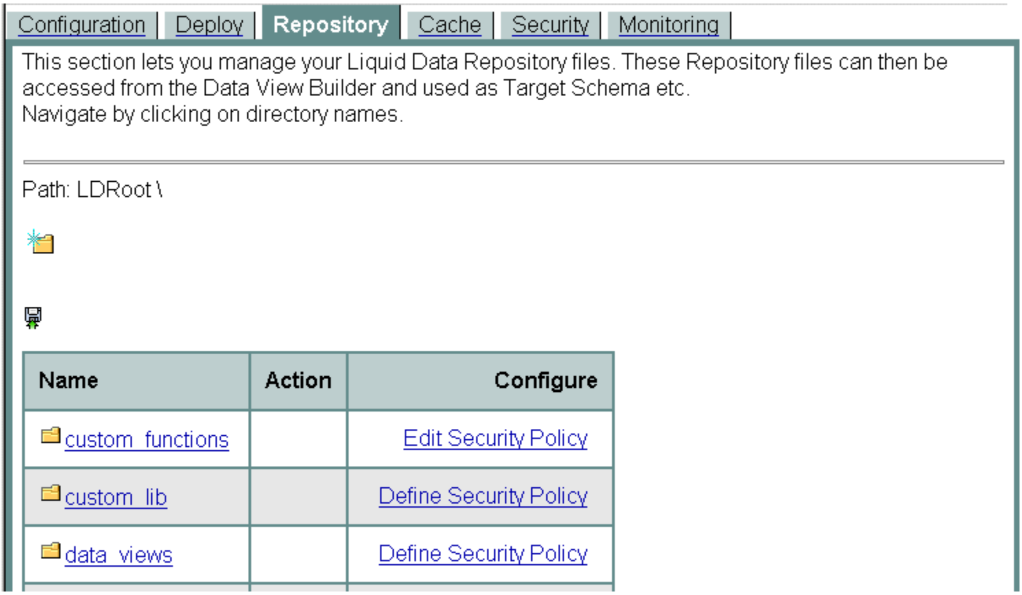
Navigating to the Repository Tab

To navigate to the Repository tab on the Liquid Data node:

1. In the left pane of the Administration Console, click the Liquid Data node.
2. Click the Repository tab.

The Administration Console displays the contents of the root directory of the server repository. For more information, see "Server Repository Location" on page 18-3.

Figure 18-2 Repository Tab on the Liquid Data Node



This section lets you manage your Liquid Data Repository files. These Repository files can then be accessed from the Data View Builder and used as Target Schema etc.
Navigate by clicking on directory names.

Path: LDRoot \

Name	Action	Configure
custom_functions		Edit Security Policy
custom_lib		Define Security Policy
data_views		Define Security Policy

Browsing the Server Repository

You browse the server repository by navigating the file system hierarchy. When you click the Repository tab, the Administration Console displays the root directory of the server repository.

Note: You must log into the Administration Console with at least `read` access to browse items in the server repository. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To navigate to a folder on the Repository tab:

- In the list of items in the current folder, click the name of the folder that you want to view.

To navigate to a parent folder on the Repository tab:

- Click the Up to Parent Folder icon or, in the directory path, click the name of the folder that you want to view.

To perform an operation on a folder or file in the repository:

- Click the appropriate icon (such as the trash can icon) or hyperlink (such as Define Security Policy) that appears on the same row as the item you want to modify.

Working with Folders and Files in the Server Repository

This section describes how to work with folders and files in the server repository. It contains the following parts:

- Downloading Files From the Server Repository
- Uploading Files to the Server Repository
- Copying and Pasting Files in the Server Repository
- Renaming Folders and Files in the Server Repository
- Deleting Folders and Files in the Server Repository
- Configuring Secure Access to Items in the Server Repository

Note: In this section, the term *item* refers to both folders and files.

Downloading Files From the Server Repository

You can download server repository files, stored on a remote server, to a local system. You might want to download files to retrieve a local copy for editing purposes. After changing the local copy of the file, you can then upload it to the remote server again, as described in "Uploading Files to the Server Repository" on page 18-8.

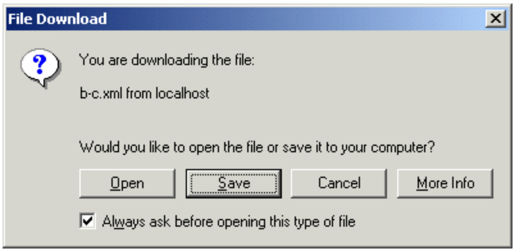
Note: You must log into the Administration Console with at least `read` access to download files from the server repository. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To download a file from the server repository:

1. Navigate to the server repository folder in which the file you want to download resides.
2. Next to the file that you want to download, click the Download icon.

   The Administration Console asks you whether you want to save the file.

Figure 18-3 Download Action Prompt



You are downloading the file:

b-c.xml from localhost

Would you like to open the file or save it to your computer?

☑ Always ask before opening this type of file

3. Click Save.

   The Administration Console displays a File Save As window.

4. Navigate to the target directory.

5. Specify a different file name, if you want.

6. Click Save.

   The Administration Console downloads the selected file to the target location.

Uploading Files to the Server Repository

You can upload files to the server repository that you have created or modified locally, such as XML schemas or custom function library definition (`CFLD`) files.

Note: You must log into the Administration Console with at least `modify` access to upload files to the server repository. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To upload a file to the server repository:

1. Navigate to the target repository folder to which you want to upload files.

2. Click the Upload icon.

   The Administration Console prompts you to specify the name of a local file to upload (see Table 18-4).

3. Do one of the following:

   – Enter the file name of the file to upload.

     or

- Click browse, navigate to the source folder, and select the file that you want to upload.

Note: You cannot select a folder.

4. Click Upload.

   The Administration Console uploads the selected file to the selected directory.

Table 18-4 Uploading a File To the Repository

Field	Description
Local File	Name of the file to upload.

Creating Sub-Folders

You can create sub-folders in any folder in the server repository. For example, you might create sub-folders in the `stored_queries` directory to define a hierarchy of stored queries for different types of users. You could create a sub-folder named `hr_queries` to contain stored queries for confidential personnel data, and another folder named `sales_queries` to contain stored queries for sales data. Once created, you can assign separate security policies to each folder to ensure that only authorized users can access these queries, as described in "Assigning Security Policies to Liquid Data Objects" on page 19-13.

Note: You must log into the Administration Console with at least `modify` access to create a folder. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To create a folder in the server repository:

1. Navigate to the repository folder to which you want to add a folder.
2. Click the Create New Folder icon.

   The Administration Console prompts you to specify a folder name (see Table 18-5).
3. Enter the name of the new folder.
4. Click Create.

   The Administration Console creates the specified folder.

Table 18-5 Creating a New Folder in the Repository

Field	Description
New Folder Name	New name for the folder. The name must comply with the naming standards of your file system. For a folder name, do not use slashes (/) or periods (.) in the name.

Copying and Pasting Files in the Server Repository

You can copy files from one location in the server repository and paste them in a different location or in the same location with different names.

Note: You must log into the Administration Console with at least `read` access to the source folder and `modify` access to the target folder. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To copy and paste a file:

1. Browse the server repository and find the item that you want to copy.
2. Next to the file that you want to copy, click the Copy button.

   The Administration Console displays a Paste link.
3. Browse the server repository and select the target folder where you want to paste the selected item.
4. Click Paste.

   The Administration Console prompts you to specify the target file name (see Table 18-6).
5. Click Paste.

   The Administration Console pastes the selected item in the target location.
   - If the target file already exists, Liquid Data notifies you that you must specify a different file name.
   - If you pasted the file in a different location, the name must be unique in the target location.

Table 18-6 Pasting a Copied File in the Repository

Field	Description
File Name	Name for the target file, including extension. The name must comply with the naming standards of your file system. For a folder name, do not use slashes (/) or periods (.) in the name.

Renaming Folders and Files in the Server Repository

You can rename files or folders in the server repository. You might want to rename items if, for example, you wanted to assign new names to files or folders that you copied from another location.

Note: You must log into the Administration Console with at least `modify` access to rename an item. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To rename an item:

1. Browse the server repository and select the item that you want to rename.

   Note: You cannot rename any of the default repository folders—only sub-folders that have been created within them, according to the instructions in "Creating Sub-Folders" on page 18-10.

2. Next to the folder or file to rename, click Rename.

   The Administration Console prompts you to specify a different name (see Table 18-7).

3. Enter the new name of the item.

4. Click Rename.

The Administration Console renames the selected item with the name you specified.

Table 18-7 Renaming a Folder or File in the Repository

Field	Description
New Name	New name for the selected folder or file. The name must comply with the naming standards of your file system. • For a folder name, do not use slashes (/) or periods (.). • For a file name, do not use slashes (/) and use periods only to denote the extension. **Note:** Do not change the filename extension for files, such as stored queries or data views.

Deleting Folders and Files in the Server Repository

You can delete folders and files from the server repository that you no longer need. You can delete only *empty* folders, so if you want to delete a folder, you must first delete its contents.

Notes: You must log into the Administration Console with at least `modify` access to delete an item. For more information, see "Defining Liquid Data Roles and Groups" on page 19-2.

To delete items from the server repository:

1. Browse the server repository and select the folder or file that you want to delete.

   Note: You cannot delete any of the default repository folders—only sub-folders that have been created within them, which is described in "Creating Sub-Folders" on page 18-10.

2. Click Delete.

The Administration Console deletes the specified file or folder and removes it from the file system.

Notes: If you delete a stored query for which caching is enabled, Liquid Data also deletes any cached results. For more information, see Chapter 22, "Configuring the Query Results Cache."

Configuring Secure Access to Items in the Server Repository

You must explicitly configure security for all items to which you want to limit access in the server repository. To configure security, you assign permissions to folders and files using security policies. You might want to assign security policies to folders, individual files, or for sub-folders in the default folders. For details on using security in Liquid Data, see Chapter 19, "Security in Liquid Data."

Permissions determine the tasks that users can perform on server repository items in the Data View Builder and the Administration Console. Users must be logged in with the following permissions:

Table 18-8 Permissions Required for Server Repository Items in the Administration Console

Method (Access Level)	Description
`All`	Allows read, modify, and execute access to the object.
`Read Configuration`	Browse or view the contents of an item, or download from the repository.

Table 18-8 Permissions Required for Server Repository Items in the Administration Console (Continued)

Method (Access Level)	Description
`Modify Configuration`	Create, modify, rename, or delete files or directories, or upload items to the directory. This level implies `read` access.
`Execute Query`	Allows execute permission on the object. Whether a user can actually execute a query (either stored or ad-hoc) is determined dynamically at runtime based on whether a user has execute access to all of the resources the query requires.

To Assign Security Policies to a Repository Item

Perform the following steps to assign security policies to a repository item:

1. Open the Liquid Data Administration Console and click the Repository tab.
2. Browse the server repository to find the folder or file to which you want to assign a security policy.
3. Click Define Security Policy or Edit Security Policy on the directory or object to which you want to assign security. If the object already has a policy defined, the link is labeled “Edit Security Policy”; if the object has no defined policy, the link is labeled “Define Security Policy.”
4. Assign security policies as needed to the repository object or directory. For details, see “Assigning Security Policies to Liquid Data Objects” on page 19-13.

Configuring the Results Cache for Stored Queries

You can configure the result cache for individual stored queries. Before you configure results caching, it must be explicitly enabled in the Cache Results field on the General tab on the Liquid Data node, as described in Chapter 5, “Configuring Liquid Data Server Settings.” For more information about the results cache and the Cache tab on the Liquid Data node, see Chapter 22, “Configuring the Query Results Cache.”

CHAPTER 19

Security in Liquid Data

This chapter describes the security infrastructure in BEA Liquid Data for WebLogic, and includes information on how to activate the security system and how to set security attributes on Liquid Data objects. It contains the following sections:

- Overview of Liquid Data Security
  - Defining Liquid Data Roles and Groups
  - Creating Liquid Data Security Resource Groups
  - Configuring Security Policies on Liquid Data Objects
- Initializing Liquid Data Security
- Defining Liquid Data Security Resource Groups
- Assigning Security Policies to Liquid Data Objects
- Integrating Liquid Data Security with Other BEA Software

Overview of Liquid Data Security

The security infrastructure in Liquid Data extends the security policies in WebLogic Server to include Liquid Data objects such as data sources and stored queries in addition to setting up security roles, groups, and users. These security policies allow Liquid Data administrators to set up rules which dynamically determine whether a given user:

- Can access a particular object
- Has read/write/execute permissions on the object or a subset of those permissions

Although Liquid Data is always enabled, by default Liquid Data objects do not have any security policies configured. Therefore an object is generally accessible unless a more restrictive policy for the object is configured.

You use the Liquid Data node of the WebLogic Administration Console to configure security in Liquid Data, which includes the following tasks:

- Setting up security roles, groups, and users
- Configuring security policies for access to Liquid Data objects

Note: Compatibility security, which uses the WebLogic 6.1 security infrastructure, is not supported in Liquid Data 8.1.

Note: The Liquid Data security model simply extends the WebLogic Server security model, therefore the information in the WebLogic Server security documentation applies to Liquid Data.

- For details of the WebLogic Server security infrastructure, see “Managing WebLogic Security” in the WebLogic Server documentation.
- For details of how you can apply a security policy to a WebLogic Server resource, see “Securing WebLogic Resources” in the WebLogic Server documentation.

Defining Liquid Data Roles and Groups

Based on the role granted to a particular user or group, Liquid Data node of the WebLogic Administration Console access to an object can be restricted or a combination of read/write/execute permissions applied.

Domains created or extended for Liquid Data by the WebLogic Configuration Wizard automatically set up Liquid Data security roles and groups, including the required LDAdmin role.

Under some circumstances, however, it may be necessary or desirable to manually create Liquid Data security management. You can do this through the Security node of the WebLogic Administration Console.

The following sections are included:

- Basic Administration Console Access Requirements
- Using Liquid Data Pre-Defined Security Roles and Groups
- Manually Creating Liquid Data Groups and Roles

Basic Administration Console Access Requirements

To use the Liquid Data node of the WebLogic Administration Console, a user or a group to which the user belongs must be able to access the Administration Console.

Only a user or group assigned the LDAdmin role has unrestricted access to the Liquid Data node of the WebLogic Administration Console. A more restrictive role called LDConsole is also available in domains created using the Liquid Data WebLogic Configuration Wizard.

For more details on Liquid Data roles and groups see “Using Liquid Data Pre-Defined Security Roles and Groups” on page 19-3.

If you are not using one of the Liquid Data domains available from the WebLogic Configuration Wizard, you can create your own roles and groups. Since the LDAdmin role is required to administer Liquid Data, creating an LDAdmin role and assigning a user or group to that role is a requirement. See “Manually Creating Liquid Data Groups and Roles” on page 19-5.

Using Liquid Data Pre-Defined Security Roles and Groups

When you create or extend a domain for Liquid Data using the WebLogic Configuration Wizard, the following roles and groups are automatically pre-defined:

- LDAdmin Role
- LDConsole Role
- LDAdministrators Group
- LDConsoleUsers Group

This arrangement, described in Table 19-1, provides for granting users and groups appropriate levels of access to the Liquid Data node of the WebLogic Administration Console.

Note: In some situations — such as when data source security handled externally to the WebLogic Platform — you will need to define your roles and groups manually. See "Manually Creating Liquid Data Groups and Roles" on page 19-5.

Table 19-1 Pre-Defined Roles and Groups in Liquid Data Domains.

Name	Description
LDAdmin role	The LDAdmin role is required to administer Liquid Data. The LDAdmin role can: • Create, modify, and delete data source descriptions. • Create, modify, and delete directories and files in the Liquid Data repository. As a member of the LDAdministrator's group, the LDAdmin has complete access (Read, Write, and Execute) to all Liquid Data resources, as well as unrestricted access to the Liquid Data node of the WebLogic Administration Console. As pre-defined in the WebLogic Configuration Wizard, the LDAdmin role is assigned to the LDAdministrators group. You can assign this role to individual users or groups to have the same effect as adding them to the LDAdministrators group. **Note:** For the most efficient management, BEA recommends granting security roles to groups rather than individual users.
LDConsole role	The optional LDConsole role is designed for users needing access to the Liquid Data node of the WebLogic Administration Console, but who can only access resources to which they have permissions. The LDConsole role can access the Liquid Data node of the WebLogic Administration Console, but can only read and modify resources to which they have appropriate permissions (granted or revoked via security policy, for example). As pre-defined in the WebLogic Configuration Wizard, the LDConsole role is assigned to the LDConsoleUsers group. You can assign this role to individual users or groups to have the same effect as adding them to the LDConsoleUsers group. **Note:** For the most efficient management, BEA recommends granting security roles to groups rather than individual users.

Table 19-1 Pre-Defined Roles and Groups in Liquid Data Domains. (Continued)

Name	Description
LDAdministrators group	The LDAdministrators group has the LDAdmin role. Therefore it has complete access (Read, Write, and Execute) to all Liquid Data resources. Any user who is a member of the WebLogic Server Administrators group is automatically a member of the LDAdministrators group.
LDConsoleUsers group	The LDConsoleUsers group has the LDConsole role. Therefore it has access to the Liquid Data console, but can only read, write, or execute resources to which they have permissions. For example, if a user Joe is a member of the LDConsoleUsers group, but is explicitly denied access to the Liquid Data Relational Data Source named `mySource` (through a security policy, for example), then Joe can log into the console but will not be able to see or modify the `mySource` data source.

You add users or groups to the LDAdmin or LDConsole roles through WebLogic Administration Console Security node, as described in "Setting Up Liquid Data Administrator Users" on page 19-8. For details on managing WebLogic security, see "Managing WebLogic Security" in the WebLogic Server documentation.

Manually Creating Liquid Data Groups and Roles

As noted in "Basic Administration Console Access Requirements" on page 19-3, only the unrestricted role named LDAdmin can fully manage Liquid Data. (See "Using Liquid Data Pre-Defined Security Roles and Groups" on page 19-3 for suggested Liquid Data security administration structure.) For details on setting up security groups and roles see "Initializing Liquid Data Security" on page 19-7.

Note: The Liquid Data security model extends the WebLogic Server security model, the information in the WebLogic Server security documentation applies to Liquid Data.

- For details of the WebLogic Server security infrastructure, see "Managing WebLogic Security" in the WebLogic Server documentation.
- For details of how you can apply a security policy to a WebLogic Server resource, see "Securing WebLogic Resources" in the WebLogic Server documentation.

Creating Liquid Data Security Resource Groups

Liquid Data allows you to create *security resource groups*, which are string prefixes for object names that you can configure with a security policy. The security policy is then inherited by any object in which the name is prefixed by the string.

For example, if you create a security resource with the name `accounting` and set up a security policy that allows execute access to the `Accountants` group, then any objects prefixed with the string `accounting.` (the period character is required) automatically is executable only by members of the `Accountants` group.

For the procedure for defining security resources groups, see "Defining Liquid Data Security Resource Groups" on page 19-11.

Configuring Security Policies on Liquid Data Objects

You can configure security policies for the Liquid Data resources such as stored queries, data sources, repository directories, and file security. For more information, see "Assigning Security Policies to Liquid Data Objects" on page 19-13.

Query Security

Query security depends on whether the query is a stored query or an ad-hoc query. For custom functions associated with a query, access is determined by the security policy associated custom function as well as the security policy associated with any data sources used by the custom function (if it uses any data sources).

Stored Queries

For stored queries, access is determined by the security policy associated with the file name of the stored query and/or with the directory and/or subdirectories of the stored query. The security policy is assigned to the stored query in the Administration Console, as described in "Assigning Security Policies to Liquid Data Objects" on page 19-13. At run time, Liquid Data checks that the user who submitted the query request has `execute` permission to the stored query before submitting the query to the Liquid Data server for processing.

Ad Hoc Queries

For ad-hoc queries, access is determined by the security policy associated with the data source(s) that the query attempts to access, as described in "Data Source Security" on page 19-7. The security policy is assigned to the data source in the Administration Console. At run time, the Query Engine checks that the user who submitted the request has `execute` permission to all data sources associated before processing the query.

Data Source Security

For all data sources, access is determined by the security policy associated with the data source description. The security policy is assigned to the data source description using the Administration Console. For more information, see "Configuring Secure Access to Data Source Descriptions" on page 6-3.

For the following types of data sources, additional steps are required to configure data access security.

- For relational database data sources, you define data access security by configuring the database connection pool in the Administration Console and specify security information. For more information, see "Creating a JDBC Connection Pool" on page 7-2 and "Securing WebLogic Resources" in the WebLogic Server documentation.
- For application view data sources, you define data access security by:
  - Configuring connection parameters on the Configure Connection Parameters page when you configure an application view using the Application View Console. For more information, see "Defining an Application View Using the WebLogic Integration Application View Console" on page 11-3.
  - Configuring the application pool username and password in the Administration Console, as described in "Configuring an Application View Data Source Description" on page 11-3.
- For data views used as data sources, access is determined by the security policy associated with the underlying query and data sources, as well as the security policy set on the data view description.

Repository Directory and File Security

You can control access to directories and files in the repository by assigning security policies to individual directories or files using the Administration Console. You can assign security policies to stored queries, data views, XML files, web service definitions, SQL Calls, and custom functions. For more information, see "Configuring Secure Access to Items in the Server Repository" on page 18-13.

Initializing Liquid Data Security

Before you can configure security policies on Liquid Data objects, there are several initialization tasks you need to perform. This section describes those tasks and contains the following sections:

- Making Sure Your Domain Includes Liquid Data
- Setting Up Liquid Data Administrator Users
- Setting Up Liquid Data Console Users

Making Sure Your Domain Includes Liquid Data

You must have Liquid Data enabled in the domain in order to use any part of Liquid Data, including the Liquid Data security. You can use the WebLogic Configuration Wizard to create a Liquid Data domain or to add Liquid Data to an existing domain. For details, see the Liquid Data *Deployment Guide*.

To verify that Liquid Data is enabled in the domain, check that the following exist in your domain:

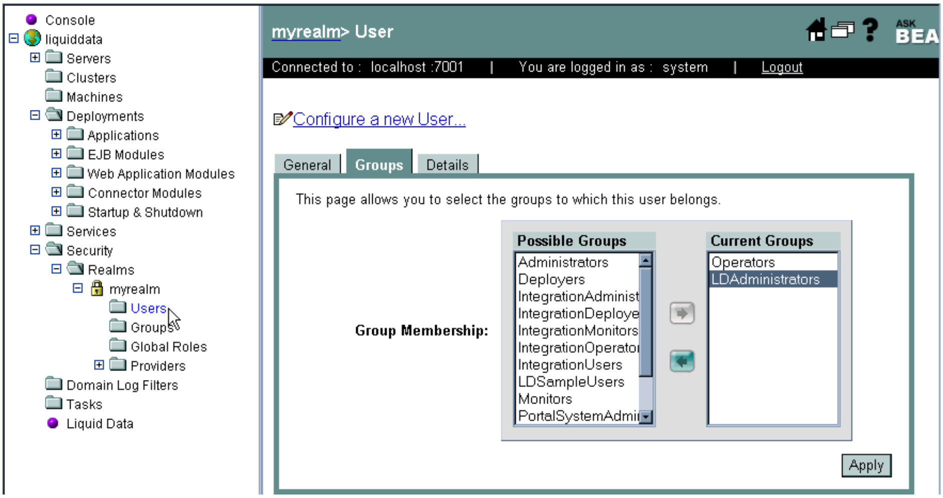
- The Administration Console includes the Liquid Data node for your domain.
- The following groups exist in the domain:
  - LDAdministrators (granted the LDAdmin Role)
  - LDConsoleUsers (granted the LDConsole Role)
- The following roles exist in the domain:
  - LDAdmin
  - LDConsole

If you use the Domain Configuration Wizard to create your Liquid Data domain or to add Liquid Data to an existing domain, these objects are all automatically created.

Setting Up Liquid Data Administrator Users

If you want a user or group to have administrative access to all objects in the Liquid Data Administration Console, you must assign the user or group to have the LDAdmin role.

Figure 19-2 Grant LDAdministrators Group Membership to a User



Connected to : localhost :7001 | You are logged in as : system

This page allows you to select the groups to which this user belongs.

Group Membership:

Current Groups: Operators, LDAdministrators

Users who are members of the WebLogic Server Administrators group automatically are members of the LDAdministrators group, so there is no need to grant those users membership in the LDAdministrators group.

Similarly, if you used the WebLogic Configuration Wizard to create or extend your domain, security setting options allow for extending LDAdmin membership to other users or groups.

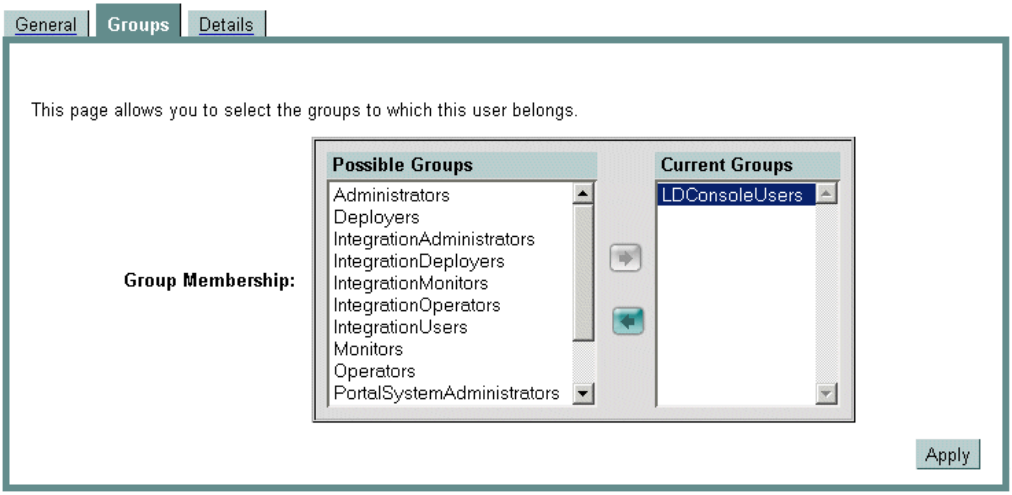
If you are not using the WebLogic Configuration Wizard and you want other users to have access to all Liquid Data resources, then you must:

1. Assume the LDAdmin role and
2. Explicitly assign the user or group to have the LDAdmin role.

Setting Up Liquid Data Console Users

If you want a user or group to be able to log into the Liquid Data console but only have access to objects to which he has permission defined (through a security policy, for example), you can assign the user or group the LDConsole role. If you have created a domain using the WebLogic Configuration Wizard, you can add users or groups to the LDConsole role by adding them to the LDConsoleUsers group.

Figure 19-3 Grant LDConsoleUsers Group Membership to a User



This page allows you to select the groups to which this user belongs.

Group Membership:

Possible Groups	Current Groups
Administrators Deployers IntegrationAdministrators IntegrationDeployers IntegrationMonitors IntegrationOperators IntegrationUsers Monitors Operators PortalSystemAdministrators	LDConsoleUsers

Users who are members of the Administrators group automatically have all of the privileges associated with the LDConsoleUsers group, so there is no need to grant those users membership in the LDConsoleUsers group.

Similarly, if you used the WebLogic Configuration Wizard to create or extend your domain, security setting options allow for extending LDAdministrator membership to other users or groups.

If you are not using the WebLogic Configuration Wizard and you want other users to have access to all Liquid Data resources, then you must:

1. Assume the Administrators role and
2. Explicitly assign the user or group to have the LDConsoleUsers group.

Defining Liquid Data Security Resource Groups

Security resource groups are prefixes to object names that you configure with a security policy. Objects with names beginning with the prefix followed by a dot (`.`) automatically inherit the security policy of the resource group. If you change the security policy of the resource group, the security policy of all objects with the prefix also changes.

Note: Security resource groups only apply to data sources and custom functions; they do not apply to repository folders and they do not apply to stored queries. For information on how security works on repository objects and folders, see "Configuring Secure Access to Items in the Server Repository" on page 18-13.

Nested Levels of Security Resource Groups

You can create multiple levels of security resource groups that inherit security policies from the parent nested resource group. Each of these nested levels must include the name of the inherited resource group as a prefix. If you create such nested resource groups, the permissions of the higher-level parent groups should be a higher level (less restrictive) of permissions than those of the lower-level (more restrictive) groups, otherwise the lower-level groups will try to get permissions that they are implicitly denied by the higher-level groups, making the lower-level groups have no net effect on the security policy.

Example

Consider a security resource group named `marketing` that has `READ` and `EXECUTE` access to the marketing data sources. You can then name the marketing data sources with the `marketing` prefix as follows:

```
marketing.WebService
marketing.OracleDB
marketing.CRM
```

These data sources automatically inherit the security policy of the `marketing` security resource group.

Now add a security resource group named `marketing.9to5`, and assign it a security policy which allows `READ` and `EXECUTE` access between the hours of 9AM and 5PM. Because it is prefixed with the name of the marketing resource group, the `marketing.9to5` resource group inherits the security policy of `marketing`, but restricts access to between the hours of 9AM and 5PM. You can then add new data sources with the `marketing.9to5` prefix (which inherit the security policies of both the `marketing` and the `marketing.9to5` security resource groups) as follows:

```
marketing.9to5.anotherWebService
marketing.9to5.sybaseDB
```

To Define a Security Resource Group

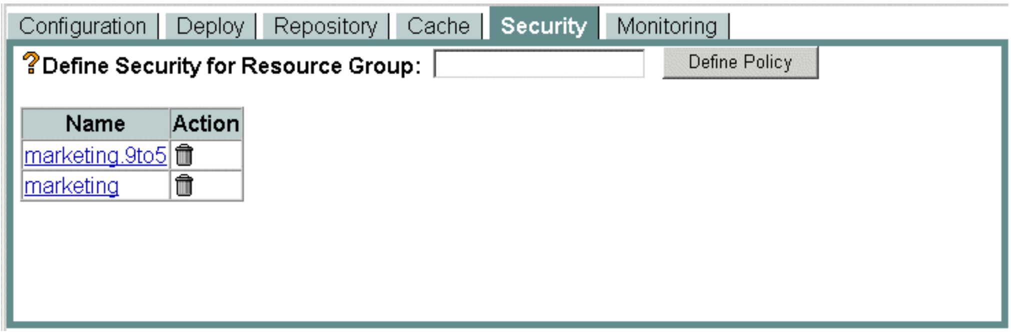
Perform the following steps to define a Liquid Data security resource group:

1. Start the Administration Console (for details, see "Starting the Administration Console" on page 4-2).

2. In the left pane of the Administration Console, expand the node for your domain and click the Liquid Data node.

3. Click the Security tab on the Liquid Data console.

Figure 19-4 Security Tab of Liquid Data Administration Console



Name	Action
marketing.9to5	🗑
marketing	🗑

4. Enter a name for the security resource group and click Define Policy.

5. Assign a security policy to the resource group. For details, see "Assigning Security Policies to Liquid Data Objects" on page 19-13.

When you go back to the Security tab, you will see the new security group. You can now create objects with the security resource group prefix and they will inherit the security policy defined for the security resource group.

Assigning Security Policies to Liquid Data Objects

You can assign a security policy to any Liquid Data object, including data sources, custom functions, repository objects, and stored queries. The security policies you assign are similar to ones you can assign to WebLogic Server objects. For details on setting up WebLogic Server security, see "Managing WebLogic Security" in the WebLogic Server documentation.

This section describes the security policies and provides a general procedure for defining security policy. The following subsections are included:

- Security Policy Methods (Access Levels)
- Conditions for the Security Policies
- To Assign a Security Policy to an Object

Security Policy Methods (Access Levels)

You can assign the following access levels in security policies:

Table 19-5 Security Policy Access Levels for Liquid Data Resources

Method (Access Level)	Description
`All`	Allows read, modify, and execute access to the object.
`Read Configuration`	Browse or view the contents of an item in the Liquid Data node of the WebLogic Administration Console. You can not necessarily see it in the Data View Builder or use it in a query.
`Modify Configuration`	In the Administration Console you can create, modify, rename, or delete files or directories, or upload items to the directory. This level also implies `read` access.
`Execute Query`	Allows execute permission on the object in the Data View Builder and use it in a query or Data View. Whether a user can actually execute a query (either stored or ad hoc) is determined dynamically at runtime based on whether a user has execute access to all of the resources the query requires.
`ConfigureCacheAPIAccess`	The ability to access the `purgeCache` APIs. Only available on the query results cache. For details, see "Configuring the Query Results Cache" on page 22-1

Note: If an attribute is set for a particular configuration that setting will take precedence over the All access level setting for that configuration. For example if All is true but Execute Query is set to No, you would not be able to execute a query on that object.

The security policy you define ensures that only authorized users and groups can perform the following:

- Access and execute a query. Liquid Data verifies user or group access, based on the security policy configured, before executing the query.
- Access specific data source elements (such as particular tables in a database, service calls in an application view, or a web service) for ad-hoc queries or custom functions. Liquid Data verifies user or group access to selected data source elements before executing the ad-hoc query.

For the procedure to set a security policy, see "To Assign a Security Policy to an Object" on page 19-15.

Conditions for the Security Policies

When you build a security policy for an object, you must specify a set of conditions for the policy. The Liquid Data server evaluates the conditions at runtime and grants access to the resource if the conditions are met. You can specify security policies that limit access based the following conditions:

- User name
- Member of group
- Role granted
- Hours of access
- When server is in development mode

The conditions allow you to restrict access to any users, groups, or roles you specify. It also provides the ability to restrict access to a given time period of the day or to when the server is running in development mode.

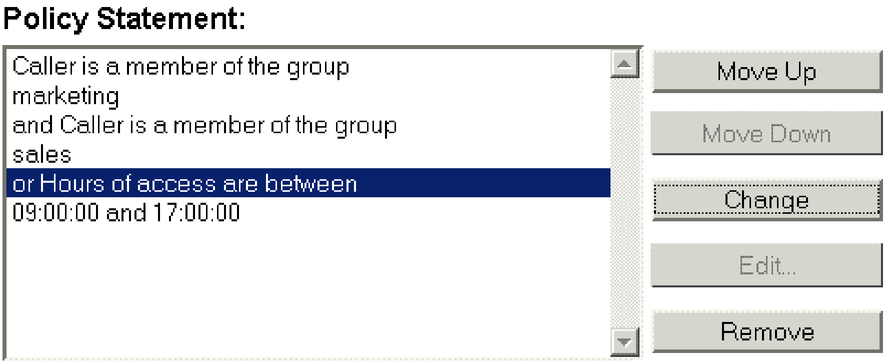
Understanding the AND and OR Condition Logic

When you configure a security policy, you can specify the operator that controls the logic between conditions. The values for the logical operator can be either AND or OR. The AND operator indicates that both conditions (before and after the AND operator) must be true for the condition to be satisfied. The OR operator indicate that either one or the other (or both) condition must be true for the condition to be satisfied.

By moving the conditions up and down in the Policy Statement pane and changing the logical operator between AND and OR, you can create complex and robust policies that are evaluated dynamically at runtime.

To change the logic for a condition from AND to OR (or vice versa), select the condition and click the Change button, as shown in Figure 19-6.

Figure 19-6 Specifying Conditions With a Policy Statement



Policy Statement:

Caller is a member of the group marketing
and Caller is a member of the group sales
or Hours of access are between 09:00:00 and 17:00:00

To Assign a Security Policy to an Object

Perform the following general steps to assign a security policy to an object. Depending on the object, the steps might vary slightly.

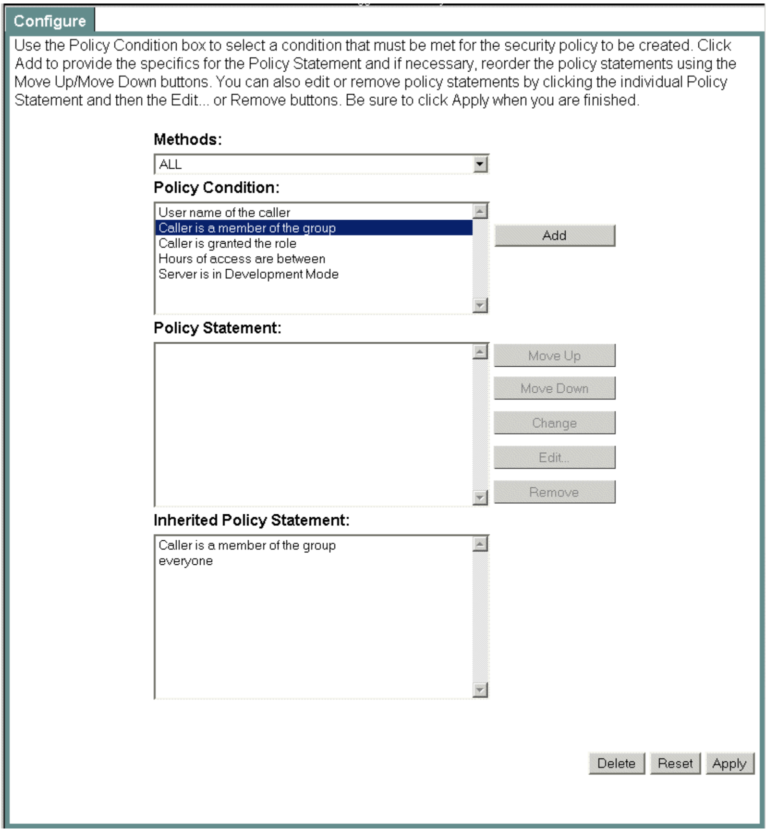
1. Start the Administration Console (for details, see "Starting the Administration Console" on page 4-2).

2. In the left pane of the Administration Console, expand the node for your domain and click the Liquid Data node.

3. Navigate to the Liquid Data object to which you want to apply a security policy. For example, if you are adding a security policy to a stored query, navigate to the Stored Queries tab.

4. Click Define Security Policy or Edit Security Policy on the object to which you want to assign security. If the object already has a policy defined, the link is labeled "Edit Security Policy"; if the object has no defined policy, the link is labeled "Define Security Policy".

Figure 19-7 Security Policy configuration screen



Use the Policy Condition box to select a condition that must be met for the security policy to be created. Click Add to provide the specifics for the Policy Statement and if necessary, reorder the policy statements using the Move Up/Move Down buttons. You can also edit or remove policy statements by clicking the individual Policy Statement and then the Edit... or Remove buttons. Be sure to click Apply when you are finished.

Methods: ALL

Policy Condition:

- User name of the caller
- Caller is a member of the group
- Caller is granted the role
- Hours of access are between
- Server is in Development Mode

Policy Statement:

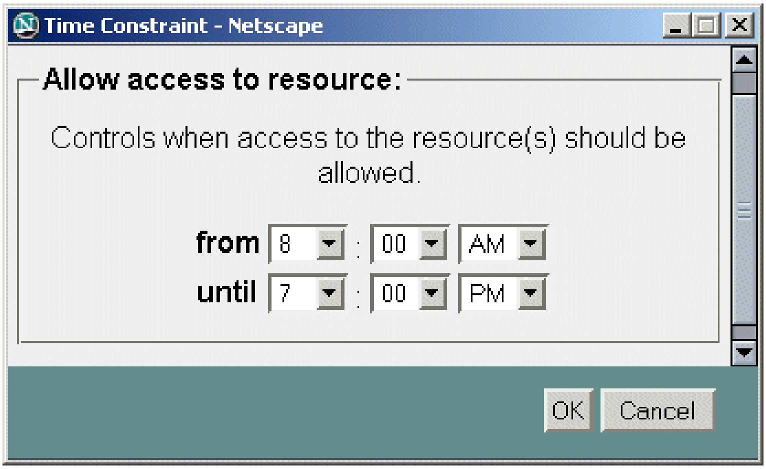
Inherited Policy Statement:

Caller is a member of the group
everyone

5. In the Methods drop-down list, select one of the options. For a description of the options, see "Security Policy Methods (Access Levels)" on page 19-13.

6. For the Policy Condition, select a condition and click Add. For examples of these screens, see Figure 19-8 and Figure 19-9.

Figure 19-8 Security Policy Time Constraints screen



Time Constraint - Netscape

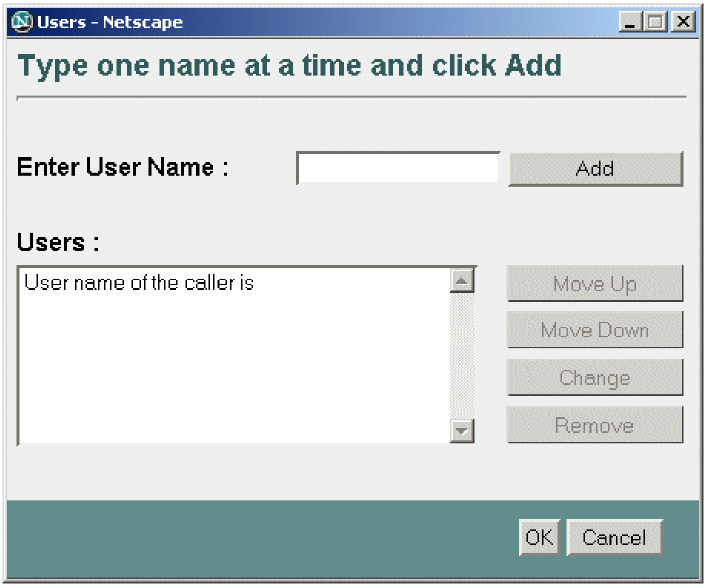
Allow access to resource:

Controls when access to the resource(s) should be allowed.

from 8 : 00 AM

until 7 : 00 PM

Figure 19-9 Security Policy Users screen



Users - Netscape

Type one name at a time and click Add

Enter User Name :

Users :

User name of the caller is

Note: You must enter the name for the User, Group, and Role screens exactly as they are defined in WebLogic Server, including capitalization. You can add any name, even ones with which there is no user, group, or role associated.

7. If you are adding a user condition, enter the name of the user and click OK. If you are adding a group condition, enter the name of the group and click OK. If you are adding the name of a role, click the name of the role and click OK. If you are setting controls on access times, enter the beginning and end times and click OK.

8. Repeat the previous two steps as necessary to add the conditions you require.

9. In the Policy Statement section, modify the policy as desired by selecting conditions and moving them up, down, or changing the logic from AND to OR. For details, see "Understanding the AND and OR Condition Logic" on page 19-14.

10. Note the inherited policies for the object. These are displayed for reference only; you cannot change them.

11. When you are satisfied with your security policy, click Apply.

Integrating Liquid Data Security with Other BEA Software

This section describes how to integrate Liquid Data security with other BEA software. It contains the following sections:

- Web Services and Liquid Data Security
- WebLogic Integration
- Application Integration and Liquid Data Security
- WebLogic Portal and Liquid Data Security
- WebLogic Workshop and Liquid Data Security

In addition to Liquid Data security tasks, integration with these components might involve other security tasks required by these components. For more information, see the documentation associated with the software with which you want to integrate.

Web Services and Liquid Data Security

A WebLogic Server web service is a proxy for the client, so the security or subject context is determined by the client connection. For more information, see "Configuring Security" in *Programming Web Services* in the WebLogic Server documentation.

WebLogic Integration

WebLogic Integration uses WebLogic Server security to protect access to WebLogic Integration business processes and other resources. User access is determined by the roles to which the user is assigned. The WebLogic Administration Console is used to define users, organizations, and roles, and also to map roles to groups in WebLogic Server security. For more information, see "Using WebLogic Integration Security" in *Deploying Solutions* in the WebLogic Integration documentation.

Application Integration and Liquid Data Security

For information about Application Integration security, see "Defining an Application View" and "Using Application Views by Writing Custom Code" in *Using Application Integration* in the WebLogic Integration documentation.

WebLogic Portal and Liquid Data Security

When authenticating users with WebLogic Portal, WebLogic Portal passes the security credentials to Liquid Data. Once the security credentials are passed to Liquid Data, Liquid Data uses its security mechanism to enforce any security policies configured in the Liquid Data domain.

If you want to use the WebLogic Portal security mechanisms as the entry point for user security when Liquid Data and WebLogic Portal are running in separate domains, you can set up your Liquid Data and WebLogic Portal environments as follows:

- For a WebLogic Portal domain to communicate with a separate Liquid Data server domain, both domains must be configured as trusted domains.
- Create a user with the same name in the WebLogic Portal domain and in the Liquid Data domain (the password need not be the same). This user must be authenticated in the portal domain and the credential is then passed to the Liquid Data domain before query execution (based on the security policies of the objects the query is requesting).

There are also other ways of setting up security when Liquid Data and WebLogic Portal are in separate domains. For example, you can create a default user (or several default users) on the Liquid Data domain and then map Portal users to the default user(s). For more information about WebLogic Portal security, see the WebLogic Portal documentation.

If Liquid Data and WebLogic Portal are running in the same domain, the Liquid Data security works the same as in any other domain.

WebLogic Workshop and Liquid Data Security

WebLogic Workshop is a development environment, and developers might need to use the Run As command to test certain security configurations in Workshop. For information about WebLogic Workshop security, see “WebLogic Workshop Security Overview” in the WebLogic Workshop documentation.

Monitoring the Server

This chapter describes how to monitor a running BEA Liquid Data for WebLogic server. It includes the following sections:

- Monitoring Liquid Data Server Statistics
- Monitoring the Server Log
- Monitoring a WebLogic Domain
- Using Other Monitoring Tools

Monitoring Liquid Data Server Statistics

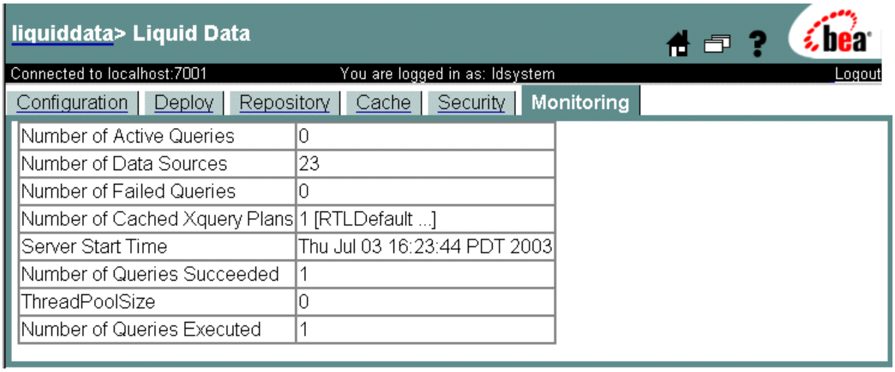
The Liquid Data node in the Administration Console includes a Monitoring tab for monitoring the current status of the Liquid Data Server.

To view the status:

1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Monitoring tab.

   The Administration Console displays a list of managed objects.

Figure 20-1 Monitoring Tab on the Liquid Data Node



Number of Active Queries	0
Number of Data Sources	23
Number of Failed Queries	0
Number of Cached Xquery Plans	1 [RTLDefault ...]
Server Start Time	Thu Jul 03 16:23:44 PDT 2003
Number of Queries Succeeded	1
ThreadPoolSize	0
Number of Queries Executed	1

The Monitoring tab displays statistics that are described in the following table:

Table 20-2 Monitoring Statistics for the Liquid Data Server

Field	Description
Number of Active Queries	Number of active queries.
Number of Data Sources	Number of configured data sources.
Number of Failed Queries	Number of failed queries.
Number of Cached XQuery Plans	Indicates the total number of XQuery plans currently cached in memory. A query with a cached plan does not require query compilation the next time it is run. For example, if you have 10 stored queries and of these you have executed only two queries (one or more times each), then the query cache will have the following entries: 2 (*queryName1*, *queryName2*)
Server Start Time	Date and time when the Liquid Data Server was started.
Number of Queries Succeeded	Number of queries successfully executed.
Thread Pool Size	Size of the thread pool.
Number of Queries Executed	Total number of queries (both successful and unsuccessful) executed since the Liquid Data Server was started.

Monitoring the Server Log

If logging is enabled on your WebLogic Server installation, the server log files contain information about the time spent to compile and execute a query. For more information, see "Server Log" in the WebLogic Server documentation

Custom applications can contain debugging calls to `stdout` that record times when Liquid Data compiles a query, submits the query to a data source for processing, receives the results from the data source, and processes the results. For more information, see "Using WebLogic Logging Services."

Monitoring a WebLogic Domain

You can use the WebLogic Server Administration Console to monitor the health and performance of the domain in which WebLogic is deployed, including such resources as servers, JDBC connection pools, JCA, HTTP, the JTA subsystem, JNDI, and EJBs. For more information, see "Monitoring a WebLogic Server Domain" in *Configuring and Managing WebLogic Server*.

Using Other Monitoring Tools

You can use performance monitoring tools, such as the OptimizeIt and JProbe profilers, to identify Liquid Data application hot spots that result in either high CPU utilization or high contention for shared resources. For more information, see "Tuning WebLogic Server Applications." For a complete list of performance monitoring resources, see "Related Reading" in *WebLogic Server Performance and Tuning*.

Checking the Status of Liquid Data Resources

This chapter describes the status page of the Liquid Data Administration console. It includes the following sections:

- Status Administration Page
- What Determines the Status of a Resource

Status Administration Page

If a resource has a known configuration problem, the problem is highlighted on the summary page for each type of resource. You can then click on the status for a particular resource to find details about the problem.

Status Values

When you view the status for a given resource, one row is displayed for each managed server in a cluster. Table 21-1 shows the values for the status and their definitions. If there are any additional errors, those are also displayed in the status table.

Table 21-1 Resource Status Values

Value	Definition
UNKNOWN	The resource status checking has not occurred. The status will be checked the next tome the resource is accessed (for example, when a client request requires the resource or when an administrator refreshes the status on the Liquid Data Administration Console).
INVALID	The resource has been checked and failed the validation. A message describing the problem is logged and is available in the console status page for the resource.
WARNING	Indicates that the resource validation encountered some problems for part of the resource. This might indicate that certain functions are available but others are not. A message describing the problem is logged and is available in the console status page for the resource.
VALID	The validation check for the resource was successful.

To Check the Status of a Invalid Resources

The summary page for each resource (for example, the stored query page) includes a column displaying the resource status. If the column has a red icon for a given resource, that indicates a problem for that resource. Click the red icon to see the details for the resource status.

What Determines the Status of a Resource

The status of a resource is validated at server startup and when a configuration is created or modified. Table 21-2 shows the logic used during resource validation to determine the status for different resources.

Table 21-2 Validation logic for resources

Liquid Data Resource	Validation Logic
Application Views and Web Services	Until the resource is accessed the first time, the status is UNKNOWN. After it has been accessed, if no functions are loaded, then it is marked INVALID. If one or more functions are loaded but one or more functions do not load, then the data source status is WARNING. Otherwise, it is VALID.
Custom Functions	Attempts to load the functions specified in the CFLD file. If no functions are loaded, then it is marked INVALID. If one or more functions are loaded but one or more functions do not load, then the data source status is WARNING. Otherwise, it is VALID.
SQL Calls and Relational Databases	Connects to the database to get the schema information for the data source. If any errors occur during this process, then the data source is marked INVALID. Otherwise it is VALID.
Delimited Files	If a schema is specified, check the schema file to see if it is a valid XML schema. If the schema is invalid, the status of the data source is INVALID. If there is no schema file, check the first row of the delimited file to infer the schema. If there are errors and the schema cannot be inferred, then the status of the data source is INVALID, otherwise it is VALID.
Stored Queries, XML Files, and Complex Parameter Types	Checks the schema file to see if it is a valid XML schema. If the schema is valid, the status of the data source is VALID, otherwise the status is INVALID.

Note: Even if the status of a resource is VALID, it does not guarantee that the resource will be available. There could be problems in the underlying systems or in other parameters that the Liquid Data server does not detect. The system is designed to highlight problems that the server is aware of, not to exhaustively check for problems.

CHAPTER 22

Configuring the Query Results Cache

This chapter describes how to manage caching for stored queries in BEA Liquid Data for WebLogic. It contains the following sections:

- Understanding Results Caching
- Setting up the Results Cache Database
- Enabling the Results Cache
- Configuring a Security Policy for Purging the Cache Results
- Configuring Results Caching for Stored Queries
- Flushing the Cache
- Purging the Cache Programmatically

Note: Caching is used with stored queries only. Caching does not apply to ad-hoc queries, which are never cached.

Understanding Results Caching

After the Query Processor executes a query, it returns to the client the data that resulted from query execution. If Liquid Data results caching is enabled, the first time a query is run, Liquid Data saves its results into a *query results cache*. The next time the query is run with the same parameters, Liquid Data checks the cache configuration and, if the results have not expired, quickly retrieves the results from the cache rather than re-running the query. In this way, for queries that are executed repeatedly with the same parameters, using the results query cache reduces processing time and enhances overall system performance.

The query results cache is disabled by default and must be enabled according to the instructions in "Enabling the Results Cache" on page 22-4. Once enabled, you can configure the cache for individual stored queries as needed, specifying how long query results are stored in the cache before they expire (time out), and explicitly flushing the query cache. For more information, see "Configuring Results Caching for Stored Queries" on page 22-6.

In general, the results cache should be periodically refreshed to reflect data changes in the underlying data stores. The more dynamic the underlying data, the more frequently the cache should be set to expire. For queries on static data, you can configure the results cache so that it never expires.

If the cache policy expires for a particular query, Liquid Data flushes the cache result automatically on the next invocation.

In the event of a Liquid Data Server shutdown, the contents of the results cache are retained. Upon server restart, the Liquid Data Server resumes caching as before. Upon the first invocation of a cached query, the Liquid Data Server checks the results cache to determine whether the cached results for this query are valid or have expired, and then proceeds accordingly.

Note: The query results cache is stored in a database that you must explicitly configure. For more information, see "Setting up the Results Cache Database" on page 22-2.

Setting up the Results Cache Database

To use results caching, you must first set up the results cache database. To set up the results cache database you need to do the following:

- Step 1: Install and Configure the Database Server
- Step 2: Run the SQL Script to Create the Cache Database
- Step 3: Create the JDBC Data Source for the Cache Database

Step 1: Install and Configure the Database Server

To use results caching, you first need to install and configure the database server according to your vendor's instructions. Liquid Data supports the following relational databases for caching:

- Oracle
- Microsoft SQL Server

Step 2: Run the SQL Script to Create the Cache Database

After you have installed and configured a database server, you need to run a SQL script on your database server that creates the Liquid Data cache database. Liquid Data provides the following scripts (in `%WL_HOME%\liquiddata\server\dbscripts`):

Table 22-1 Cache Database Creation Scripts

Database	Script File Name
Oracle	`ldcache_oracle.ddl`
Microsoft SQL Server	`ldcache_mssqlserver.ddl`

Step 3: Create the JDBC Data Source for the Cache Database

After you have created the Liquid Data cache database, you need to create a JDBC data source in WebLogic Server that points to the Liquid Data cache database. Using the Administration Console, create a JDBC data source with the following name:

```
ldCacheDS
```

For details about creating a JDBC data source, see "Creating a JDBC Data Source" on page 7-5.

Note: If you are using Oracle for the cache database, you must set the Honor Global Transactions setting to `FALSE` (it is set to `TRUE` by default). When you create the Oracle JDBC data source in the WebLogic Administration Console, you must uncheck the Honor Global Transactions box.

Once created, you can enable the result cache, as described in the following section, "Enabling the Results Cache" on page 22-4

Enabling the Results Cache

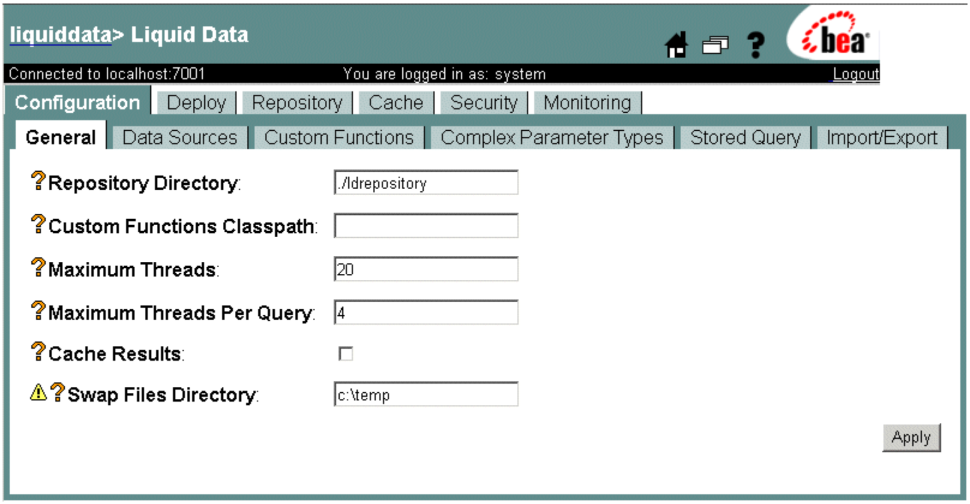
By default, results caching is disabled in Liquid Data. To use results caching, you must explicitly enable it in the General tab on the Liquid Data node in the Administration Console. Before you enable caching, make sure that you have set up the Liquid Data cache database as described in "Setting up the Results Cache Database" on page 22-2.

Note: If caching is enabled but the cache data source has not been configured, an exception is thrown in the Administration Console.

To enable result caching in Liquid Data:

1. In the left pane of the Administration Console, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the General tab.

Figure 22-2 Enabling Results Caching in the General Tab on the Liquid Data Node



Repository Directory: ./ldrepository

Custom Functions Classpath:

Maximum Threads: 20

Maximum Threads Per Query: 4

Cache Results: □

Swap Files Directory: c:\temp

4. On the General tab, check the Cache Results box.
5. Click Apply.

Once caching is enabled, you must explicitly configure the results cache according to the instructions in the following section, "Configuring Results Caching for Stored Queries" on page 22-6.

Configuring a Security Policy for Purging the Cache Results

You can set a security policy on the query cache. The security policy determines if a user can purge results from the cache. The cache security policy is global across the entire cache; that is, if a user has the authority to purge the cache, then that user can purge the cache for any query for which he has access. Depending on the security policy of different queries, the user might have permissions to see some queries and not others.

Security Policy Methods

Table 22-3 describes the different methods you can configure for the cache purging security policy. Set these methods from the Methods drop list on the Define Security Policy page for the cache.

Table 22-3 Security Policy Methods for Purging the Cache

Method	Description
`ALL`	All of the below privileges.
`Read Configuration`	The ability to access the security policy for the cache (through the Edit Security Policy link).
`Modify Configuration`	The ability to access and modify the security policy for the cache (through the Edit Security Policy link).
`ConfigureCacheAPIAccess`	The ability to access the `purgeCache` APIs.

To Configure a Cache Purging Security Policy

To configure a security policy to determine who has permission to purge the cache, perform the following steps:

1. Start the Administration Console (for details, see "Starting the Administration Console" on page 4-2).
2. In the left pane of the Administration Console, expand the node for your domain and click the Liquid Data node.
3. Navigate to the Cache tab on the Liquid Data Administration Console.

4. Click Define Security Policy or Edit Security Policy on the Cache tab. If the cache already has a policy defined, the link is labeled "Edit Security Policy"; if the cache has no defined policy, the link is labeled "Define Security Policy."

5. Assign a security policy for purging the cache and click Apply. For details on assigning security policies, see "Assigning Security Policies to Liquid Data Objects" on page 19-13.

Configuring Results Caching for Stored Queries

Each stored query has its own *cache policy* that determines how Liquid Data manages results caching for that stored query. This section describes how to configure the cache policy for stored queries. It includes the following sections:

- To Create the Cache Policy
- To Edit the Cache Policy
- To Remove the Cache Policy

To Create the Cache Policy

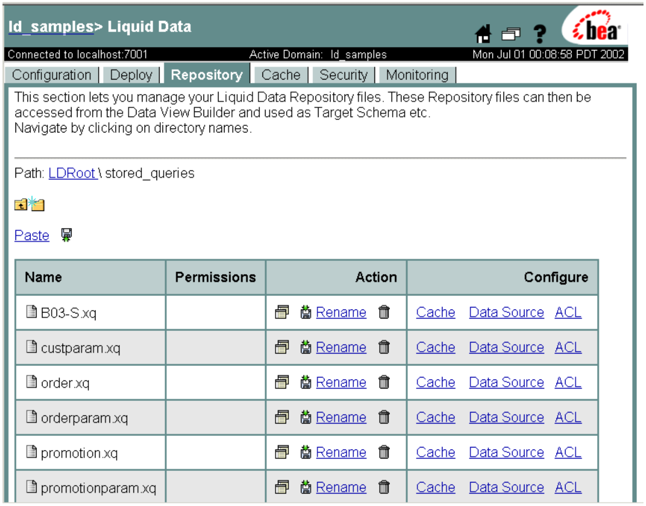
To configure the results cache for a stored query initially:

1. Navigate to the Repository tab on the Liquid Data node, as described in "Navigating to the Repository Tab" on page 18-5.

2. On the Repository tab, click the `stored_queries` folder.

   The Administration Console displays a list of stored queries in the server repository.

Figure 22-4 List of Stored Queries in the Repository


ld_samples> Liquid Data
bea
Connected to localhost:7001
Active Domain: ld_samples
Mon Jul 01 00:08:58 PDT 2002
Configuration
Deploy
Repository
Cache
Security
Monitoring

This section lets you manage your Liquid Data Repository files. These Repository files can then be accessed from the Data View Builder and used as Target Schema etc.
Navigate by clicking on directory names.

Path: LDRoot \ stored_queries

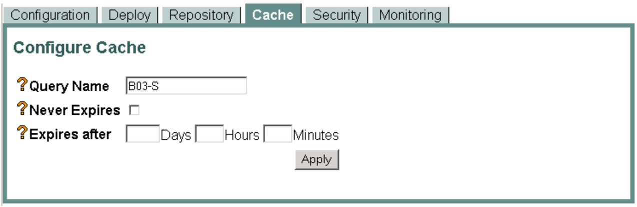
Paste

Name	Permissions	Action	Configure
B03-S.xq		Rename	Cache Data Source ACL
custparam.xq		Rename	Cache Data Source ACL
order.xq		Rename	Cache Data Source ACL
orderparam.xq		Rename	Cache Data Source ACL
promotion.xq		Rename	Cache Data Source ACL
promotionparam.xq		Rename	Cache Data Source ACL

3. Select the stored query for which you want to configure caching.
4. Click Cache.

   The Administration Console displays the Configure Cache tab.

Figure 22-5 Configure Cache Tab



Configure Cache

?Query Name	B03-S
?Never Expires	□
?Expires after	Days Hours Minutes

Apply

5. Enter the information described in the following table.

Table 22-6 Configure Cache Settings

Permission	Task
Query Name	Name of the selected stored query.
Never Expires	If selected (checked), the results cache for the selected stored query never expires. If cleared (not checked), you can configure an expiration time in the Expires After fields.
Expires After	Expiration time, which is calculated from when the query cache is first created. • Days—Number of days, up to 999 days. • Hours—Number of hours, up to 99 hours. • Minutes—Number of minutes, up to 99 minutes. If Days, Hours, and Minutes are all set to zero (0), then the results cache never expires.

Note: The expiration time should reflect the degree to which the data in the underlying data source(s) is expected to change. In general, for more dynamic data (such as real-time data feeds), specify a shorter expiration time. For more static data (such as general product or personnel information), specify a longer expiration time.

6. Click Apply.

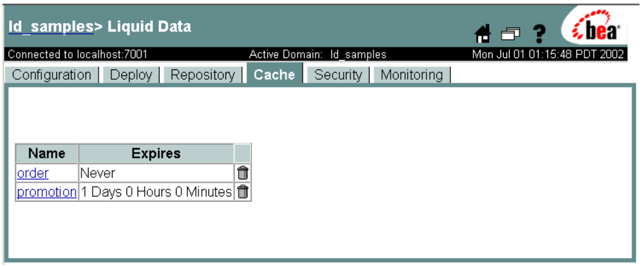
   The Administration Console creates the cache policy and Liquid Data begins caching results for the selected query. Any expiration times are calculated starting with the time that the cache policy was initially created.

To Edit the Cache Policy

To edit the cache policy for a stored query:

1. In the left pane, click Liquid Data.
2. Click the Cache tab.

Figure 22-7 Cache Tab of the Liquid Data Administration Console



ld_samples> Liquid Data

Connected to localhost:7001 Active Domain: ld_samples Mon Jul 01 01:15:48 PDT 2002

Name	Expires	
order	Never	🗑
promotion	1 Days 0 Hours 0 Minutes	🗑

3. Click the name of the stored query whose cache you want to configure.
4. Change the information described in Table 22-6, “Configure Cache Settings,” on page 22-8, as needed.
5. Click Apply.

   The Administration Console updates the selected cache policy and flushes any cached results for this stored query.

To Remove the Cache Policy

To remove the cache policy for a stored query:

1. In the left pane, click Liquid Data.
2. Click the Cache tab (see Figure 22-7).
3. Click the trash can next to the stored query whose cache you want to delete.
4. When prompted, click Yes to confirm removal.

   The Administration Console removes the selected cache policy and flushes any cached results for this query.

   Note: If you delete a stored query for which caching is enabled, Liquid Data also deletes the cached query plan and any cached results. For more information, see “Deleting Folders and Files in the Server Repository” on page 18-13.

Flushing the Cache

Flushing the cache is when all of the entries in the cache are purged from the cache. When the cache is flushed, all queries will execute against their data sources until they are cached again. Liquid Data flushes the cached query result for a given stored query whenever any of the following events occur:

- The stored query is updated or deleted.
- Caching is disabled on the Liquid Data Server.

Liquid Data flushes the cached query result for a given stored query *on the next query invocation* whenever any of the following events occur:

- The query results have expired per the cache policy.
- The cache policy for a stored query is updated or deleted.

Purging the Cache Programmatically

You can use the `com.bea.ldi.cache.ejb` package to purge the cache programmatically. For details on using the EJB API to purge the cache, see the "Using the Cache Purging APIs" chapter in the Liquid Data *Application Developer's Guide* and the Javadoc.

CHAPTER 23

Generating and Publishing Web Services

This chapter describes how to publish Liquid Data stored queries as Web Services. It contains the following sections:

Using the Administration Console, you can publish Liquid Data stored queries as Web Services. Web-based applications can then invoke Liquid Data queries as Web Service clients.

Viewing a Demo

Generate Web Service Demo... If you are looking at this documentation online, you can click the "Demo" button to see a viewlet demo showing how to use the Liquid Data Administration Console to generate a Web Service from a stored query. The viewlet also demonstrates how to test the generated Web Service in BEA WebLogic Workshop. The demo assumes that you have already stored the query you want to use in the Liquid Data Repository.

About Web Services

Web Services are a type of service that can be shared by, and used as components of, distributed Web-based applications. Web Services communicate with clients (both end-user applications or other Web Services) through XML messages that are transmitted by standard Internet protocols, such as HTTP. Web Services endorse standards-based distributed computing. Currently, popular Web Service standards are SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery, and Integration).

Creating a New Web Service from a Stored Query

This section describes how to generate a Web Service from a stored query in the server repository. For each stored query, you can have up to one generated Web Service.

Notes: The names of stored queries to be generated as Web Services must begin with an alphabetic character, followed by other alphabetic characters or numbers. The file name must also have an `.xq` suffix. For more information, see Naming Conventions for Stored Queries in *Building Queries and Data Views*.

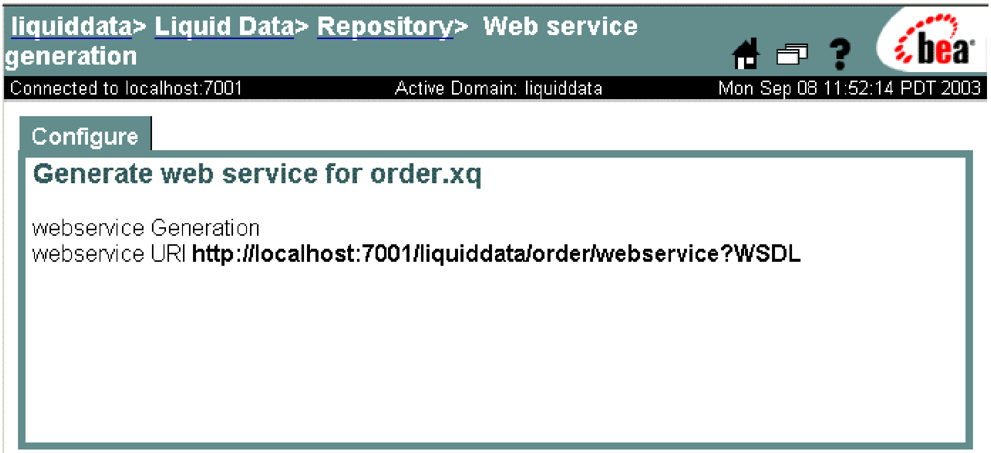
To create a new Web Service from a stored query:

1. In the left pane of the Administration Console, click the Liquid Data node.
2. In the right pane, click the Stored Queries tab.
3. In the list of queries in the Stored Queries tab, find the query with which you want to generate a Web Service.
4. Click Generate Web Service for the query.

   Note: If the Generate Web Service link is not available, then a Web Service has already been generated for this stored query. Also, if the query does not have a schema already associated with it, you must configure one before the Generate Web Service link appears.

5. When the Web Service generation is finished, a screen appears containing the URL of the WSDL file for the Web Service.

Figure 23-1 Web Service Generation Success Screen


liquiddata> Liquid Data> Repository> Web service generation
Connected to localhost:7001
Active Domain: liquiddata
Mon Sep 08 11:52:14 PDT 2003
Configure

Generate web service for order.xq

webservice Generation
webservice URI **http://localhost:7001/liquiddata/order/webservice?WSDL**

The URL of the WSDL of a generated Web Service has the following pattern:

`http://HOSTNAME:PORT/liquiddata/`*`query_name`*`/webservice?WSDL`

For example, if the stored query is named `order.xq`, then the URL of its WSDL is `http://localhost:7001/liquiddata/order/webservice?WSDL`.

If the associated stored query is modified or deleted, then this generated Web Service is deleted automatically. If the stored query has been modified, you need to explicitly create it again using the instructions in this section.

Modifying a Web Service

You cannot directly modify a generated Web Service. If the associated stored query is modified or deleted, the generated Web Service is deleted automatically.

To modify a Web Service:

1. Delete the Web Service according to the instructions in "Deleting a Generated Web Service" on page 23-4.

2. Regenerate the Web Service according to the instructions in "Creating a New Web Service from a Stored Query" on page 23-2.

Deleting a Generated Web Service

You can delete a generated Web Service that you no longer need or that you want to regenerate.

Note: A Web Service is automatically deleted if its associated stored query is subsequently changed or deleted.

To directly delete a Web Service:

1. In the left pane of the Administration Console, click the Liquid Data node.
2. In the right pane, click the Repository tab.
3. Click the `web_services_gen` directory.
4. Scroll to find the Web Service that you want to delete.
5. Click the delete icon next to the Web Service you want to delete.

Testing a Generated Web Service

You can use WebLogic Workshop to test a Web Service that you have generated with the Administration Console.

To test the Web Service:

1. Start WebLogic Workshop.
2. Create a Web Service Project in a WebLogic Workshop Application. (For example, select the top-level folder in the Application file tree, right-click, select New Project, and choose Web Service as the project type.)
3. Create a folder in your Web Service Project.
4. Right click the folder and create a new Java Control.
5. Select Web Service as the type of Java Control.
6. Enter the WSDL URL in the Web Service Control wizard and click Create. Workshop generates the Java Control Extension (`.jcx`) file.
7. Select the new Java Control Extension (`.jcx`) file, right-click, and select Generate Test JWS File. Workshop generates a Java Web Service (`.jws`) file.
8. Select the new Java Web Service (`.jws`) file and click the test button to run the Web Service (or choose Debug —> Start from the menu).

9. In the Test Form, click the startTestDrive button.
10. On the next page, click the Continue This Conversation link.
11. Click the Order button to run the order query in the Web Service.
12. The response from the Web Service appears. The response is the results of the query returned from the Liquid Data server.

Managing the Deployment of a Generated Web Service

When you create a Web Service, the Administration Console automatically deploys the generated `EAR` file to all nodes in the currently active domain. If you subsequently need to manage this `EAR` file, such as undeploying or redeploying it:

1. In the left pane, click Deployments->Applications.
2. Select the `EAR` file from the list.
3. Click the Deploy tab. For additional instructions, see "Deploying Applications and Modules" in the WebLogic Server documentation.

For detailed information about WebLogic Web Services, see Programming WebLogic Web Services in the WebLogic Server documentation.

Finding the Target Schema for a Generated Web Service

If you want to find the target schema for a generated web service, the target schema is stored in the generated `EAR` file. To view the contents of the `EAR` file, open the file with a utility such as WinZip. The `EAR` file is located in the following directory:

```
<ld_repository>/web_services_gen
```

The filename of the `EAR` file is generated based on the filename of the stored query from which the web service was generated. For example, if a stored query is named `order.xq`, the generated web service name is `order.ear`.

Invoking Published Web Services

You invoke Liquid Data Web Services that were generated in the Administration Console using the same approach that you would use for invoking any WebLogic Web Service. For more information, see "Invoking Queries in Web Service Clients" in the *Application Developer's Guide*.

Index