



BEA Liquid Data for WebLogic™

Administration Guide

Release: 1.0.1
Document Date: October 2002
Revised: December 2002

Copyright

Copyright © 2002 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop, BEA Liquid Data for WebLogic, and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Administration Guide

Part Number	Date	Software Version
N/A	October 2002	1.0
N/A	December 2002	1.0.1

Contents

About This Document

What You Need to Know	x
e-docs Web Site	x
How to Print the Document	xi
Related Information	xi
Contact Us!	xii
Documentation Conventions	xii

1. Overview of Liquid Data Administration

Working with WebLogic Domains	1-2
Starting the Server and Running the Administration Console	1-2
Configuring Access to Data Sources	1-3
Data Source Descriptions	1-3
Supported Data Source Types	1-3
Managing the Server Repository	1-4
Implementing Security	1-5
Configuring Query Results Caching	1-5
Configuring Custom Functions	1-6
Importing and Exporting Server Configurations	1-6
Deploying Liquid Data in a Production Environment	1-6
Generating Web Services from Stored Queries	1-7
Ongoing Liquid Data Management Tasks	1-7

2. Creating Liquid Data Domains

Understanding WebLogic Domains and Administration	2-2
Understanding the Relationship of Liquid Data to WebLogic Domains	2-2
Creating a New Domain	2-3

Adding Liquid Data to a Domain	2-3
3. Starting and Stopping the Server	
Starting WebLogic Server and the Liquid Data Server	3-1
You Must Start the Server in the Appropriate WebLogic Domain	3-2
Starting the Server	3-3
Stopping the Server	3-3
Next Steps	3-4
4. Using the WLS Administration Console	
Using the Administration Console to Manage Liquid Data	4-2
Starting the Administration Console	4-2
Overview of the Administration Console	4-3
Finding the Liquid Data Node in the Navigation Tree	4-5
5. Configuring Liquid Data Server Settings	
Configuring Server Settings	5-1
Modifying Server Settings	5-4
6. Viewing and Accessing All Configured Data Sources	
Viewing All Configured Data Sources	6-2
Configuring Secure Access to Data Source Descriptions	6-3
Removing Data Source Descriptions	6-5
Distributing Data Source Descriptions to Other Liquid Data Servers	6-6
7. Configuring Access to Relational Databases	
Connection Pool URLs and Driver Names for JDBC Data Sources	7-2
Creating a JDBC Connection Pool	7-2
Creating a JDBC Data Source	7-5
Creating a Relational Database Data Source Description	7-7
Summary of Configured Data Sources	7-11
Modifying a Relational Database Source Description	7-11
Modifying Data Source Description Settings	7-12
Modifying JDBC Connection Pools or JDBC Data Sources	7-12
Removing a Relational Database Source Description	7-13

8. Configuring Access to XML Files

Creating an XML File Data Source Description	8-1
Summary of Configured Data Sources	8-3
Modifying an XML File Data Source Description	8-4
Removing an XML File Data Source Description	8-5

9. Configuring Access to Web Services

Creating a Web Service Data Source Description	9-1
Summary of Configured Data Sources	9-3
Modifying a Web Service Data Source Description	9-4
Removing a Web Service Data Source Description	9-5

10. Configuring Access to Application Views

Adding Liquid Data to a WebLogic Platform or WebLogic Integration Domain ..	10-2
Starting the Liquid Data Server in the WebLogic Platform or WebLogic Integration Domain	10-3
Defining an Application View Using the WebLogic Integration Application View Console	10-3
Configuring an Application View Data Source Description	10-4
Creating an Application View Data Source Description	10-4
Summary of Configured Data Sources	10-6
Modifying an Application View Data Source Description	10-7
Removing an Application View Data Source Description	10-8

11. Configuring Access to Data Views

Creating a Data View Data Source Description	11-2
Summary of Configured Data Sources	11-4
Modifying a Data View Data Source Description	11-4
Removing a Data View Data Source Description	11-5

12. Deploying Liquid Data Components

Liquid Data Components to Deploy	12-2
Navigating to the Deploy Tab	12-2

13. Configuring Access to Custom Functions

About Custom Functions	13-2
Use Cases for Custom Functions.....	13-2
Components of Custom Functions	13-2
Administration Tasks for Custom Functions	13-3
Creating a Custom Function Description	13-4
Summary of Configured Custom Function Groups.....	13-5
Configuring Secure Access to Custom Function Descriptions	13-6
Modifying a Custom Function Description	13-7
Removing a Custom Function Description	13-8

14. Importing and Exporting Liquid Data Configurations

About Liquid Data Configurations	14-1
What Liquid Data Imports and Exports.....	14-2
What Liquid Data Does Not Import or Export.....	14-3
WebLogic Server Specific Configuration Information.....	14-3
Files Added to the Liquid Data Server Repository	14-3
Repository Name.....	14-4
File Swap Configuration	14-4
Navigating to the Import/Export Tab	14-4
Exporting a Liquid Data Configuration.....	14-5
Importing a Liquid Data Configuration.....	14-7

15. Managing the Liquid Data Server Repository

About the Liquid Data Server Repository	15-2
Contents and Organization of the Server Repository	15-2
Server Repository Location.....	15-3
Server Repository File System Hierarchy	15-3
Considerations for Evolving the Repository	15-4
Navigating to the Repository Tab.....	15-5
Browsing the Server Repository.....	15-6
Working with Folders and Files in the Server Repository	15-7
Downloading Files From the Server Repository	15-7
Uploading Files to the Server Repository	15-8
Creating Sub-Folders.....	15-10

Copying and Pasting Files in the Server Repository	15-11
Renaming Folders and Files in the Server Repository	15-12
Deleting Folders and Files in the Server Repository	15-13
Configuring Secure Access to Items in the Server Repository	15-13
Creating Data Views from Stored Queries	15-16
Configuring the Results Cache for Stored Queries	15-17

16. Implementing Security

Overview of Liquid Data Security	16-2
Administration Console Security	16-2
Data Access Security	16-3
Query Security	16-3
Data Source Security	16-4
Repository Directory and File Security	16-5
Liquid Data Server Security Implementation Process	16-5
Initial Security Setup	16-6
Defining a Compatibility Security Realm	16-7
Enabling Liquid Data Secure Mode	16-8
Configuring Liquid Data Users	16-9
Configuring Liquid Data Groups	16-10
Group to Add for Liquid Data	16-11
Configuring Groups	16-11
Assigning ACLs to Liquid Data Resources	16-13
Liquid Data Resources Requiring ACL Configuration	16-13
Access Levels	16-14
How ACLs Affect Access	16-14
Assigning Permissions, Users, and Groups to ACLs	16-14
Integrating Liquid Data Security With BEA Other Software	16-16
Web Services and Liquid Data Security	16-17
WebLogic Integration	16-17
Application Integration and Liquid Data Security	16-17
Business Process Management and Liquid Data Security	16-18
B2B Integration and Liquid Data Security	16-18
WebLogic Portal and Liquid Data Security	16-18
WebLogic Workshop and Liquid Data Security	16-19

17. Monitoring the Server

Monitoring Liquid Data Server Statistics	17-1
Monitoring the Server Log	17-3
Monitoring a WebLogic Domain	17-3
Using Other Monitoring Tools	17-3

18. Configuring the Query Results Cache

Understanding Results Caching.....	18-1
Setting up the Results Cache Database.....	18-2
Step 1: Install and Configure the Database Server.....	18-3
Step 2: Run the SQL Script to Create the Cache Database	18-3
Step 3: Create the JDBC Data Source for the Cache Database.....	18-3
Enabling the Results Cache	18-4
Configuring Results Caching for Stored Queries	18-5
Creating the Cache Policy	18-5
Editing the Cache Policy	18-7
Removing the Cache Policy	18-8
Flushing the Cache	18-9

19. Generating and Publishing Web Services

Viewing a Demo	19-2
About Web Services	19-2
Creating a New Web Service from a Stored Query.....	19-2
Modifying a Web Service.....	19-5
Deleting a Web Service	19-5
Testing a Generated Web Service.....	19-6
Managing the Deployment of a Generated Web Service	19-7
Invoking Published Web Services	19-8

Index

About This Document

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

This document covers the following topics:

- 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 9, “Configuring Access to Web Services”
- Chapter 10, “Configuring Access to Application Views”
- Chapter 11, “Configuring Access to Data Views”
- Chapter 12, “Deploying Liquid Data Components”
- Chapter 13, “Configuring Access to Custom Functions”
- Chapter 14, “Importing and Exporting Liquid Data Configurations”
- Chapter 15, “Managing the Liquid Data Server Repository”
- Chapter 16, “Implementing Security”

-
- Chapter 17, “Monitoring the Server”
 - Chapter 18, “Configuring the Query Results Cache”
 - Chapter 19, “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 1.0 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.

Convention	Item
<i>italics</i>	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. <i>Examples:</i> <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
monospace boldface text	Identifies significant words in code. <i>Example:</i> <pre>void commit ()</pre>
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> <pre>String <i>expr</i></pre>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> <pre>LPT1 SIGNON OR</pre>
{ }	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. <i>Example:</i> <pre>buildobjclient [-v] [-o name] [-f <i>file-list</i>]... [-l <i>file-list</i>]...</pre>

Convention	Item
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
...	<p>Indicates one of the following in a command line:</p> <ul style="list-style-type: none">■ 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 <p>The ellipsis itself should never be typed.</p> <p><i>Example:</i></p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
.	Indicates the omission of items from a code example or from a syntax line.
.	The vertical ellipsis itself should never be typed.
.	

1 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](#)

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 *Deploying Liquid Data*.

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 ACLs. 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 9, “Configuring Access to Web Services”
Application views	Chapter 10, “Configuring Access to Application Views”
Data views	Chapter 11, “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 15, “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 realms, users and groups, Access Control Lists (ACLs) and permissions, and so on. Liquid Data uses ACLs 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.

At a minimum, you must set up basic security according to the instructions in [“Initial Security Setup” on page 16-6](#). If you want to use Liquid Data security in this deployment, you must enable secure mode 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, you need to explicitly configure secure access to data source descriptions, the server repository, stored queries, and custom function descriptions, as described in [Chapter 16, “Implementing Security.”](#)

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 18, “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 13, “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 14, “Importing and Exporting Liquid Data Configurations,”](#) and also “Copying a Server Configuration to Another Server” in “[Deployment Tasks](#)” in *Deploying Liquid Data*.

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 12, “Deploying Liquid Data Components.”](#) For detailed information about deploying Liquid Data in various types of WebLogic domains, see “[Deployment Tasks](#)” in *Deploying Liquid Data*.

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 19, “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. For detailed information on management and monitoring, see [Chapter 17, “Monitoring the Server.”](#) For detailed information about tuning Liquid Data performance, see “[Tuning Performance](#)” in *Deploying Liquid Data*.

1 *Overview of Liquid Data Administration*

2 Creating Liquid Data Domains

This section 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 a 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 *Deploying Liquid Data*.

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 [“Configuring and Managing WebLogic Server Domains”](#) in the WebLogic Server *Administration Guide*.

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 starting from scratch and do not have an existing domain to which you want to add Liquid Data, you first need to create a domain using the WebLogic Platform Configuration Wizard. For more information, see [“Creating a New WebLogic Domain”](#) in *Using the Configuration Wizard* in the WebLogic Platform documentation.

Adding Liquid Data to a 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 [Chapter 12, “Deploying Liquid Data Components,”](#) as well as [“Deployment Tasks”](#) in *Deploying Liquid Data*.

3 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 preconfigured domains are shown in [“You Must 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/docs70/adminguide/startstop.html) (<http://edocs.bea.com/wls/docs70/adminguide/startstop.html>) in the [BEA WebLogic Server Administration Guide](http://edocs.bea.com/wls/docs70/adminguide/index.html) (<http://edocs.bea.com/wls/docs70/adminguide/index.html>).

You Must Start the Server in the Appropriate WebLogic Domain

You must start the Liquid Data server in the appropriate WebLogic domain. A preconfigured 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 *Deploying Liquid Data* and “[Creating a New WebLogic Domain](#)” in *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 Preconfigured Domain and Start Commands for Samples Server

Platform	Windows and UNIX Paths to Start Samples Server in Each Domain
Windows	<ul style="list-style-type: none">■ Start—>Programs—>Liquid Data for WebLogic 1.0—><i>Launch Liquid Data Server</i>Or■ <code>WL_HOME\liquiddata\samples\config\ld_samples\startWebLogic.cmd</code>
UNIX	<code>WL_HOME/liquiddata/samples/config/ld_samples/startWebLogic.sh</code>

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 *Deploying Liquid Data*.

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 preconfigured 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 *Deploying Liquid Data*.

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.”](#)

4 Using the WLS Administration Console

This topic 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 add 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 7.0—>Liquid Data for WebLogic 1.0—>Start 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.

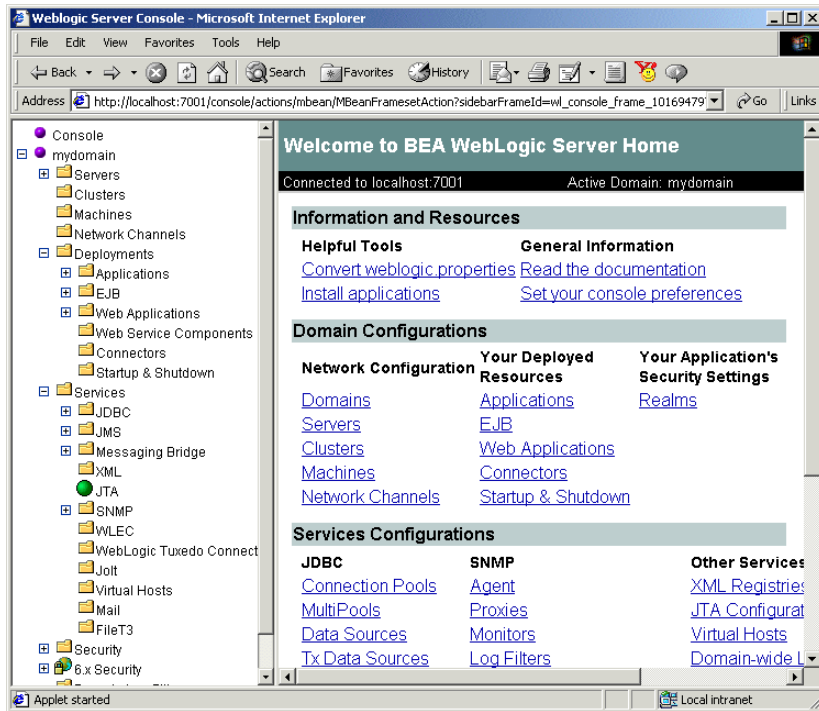
You can expand and collapse nodes in the tree by clicking on the `+` and `-` signs next to the nodes as follows:

- 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 and Using the Administration Console](http://edocs.bea.com/wls/docs70/adminguide/overview.html#start_admin_console) (http://edocs.bea.com/wls/docs70/adminguide/overview.html#start_admin_console) in the [BEA WebLogic Server Administration Guide](http://edocs.bea.com/wls/docs70/adminguide/index.html)(<http://edocs.bea.com/wls/docs70/adminguide/index.html>).

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

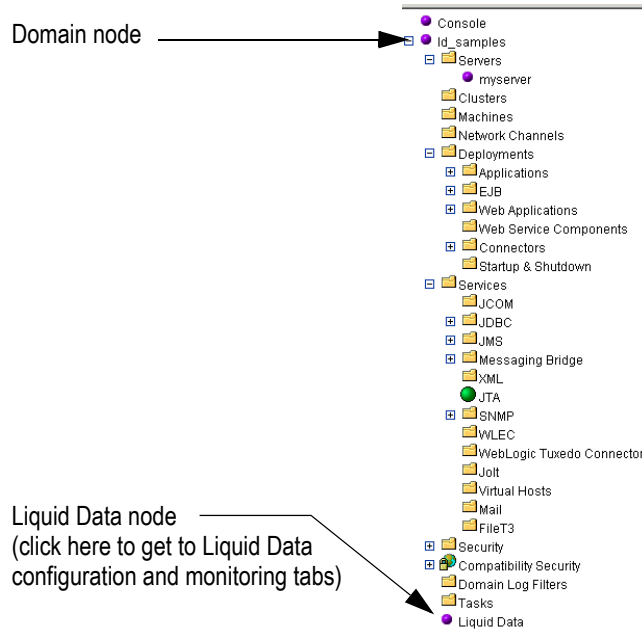


Finding the Liquid Data Node in the Navigation Tree

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



5 Configuring Liquid Data Server Settings

This topic 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.

To configure Liquid Data server settings:

1. In the left pane of the Administration Console, click the Liquid Data node.

5 Configuring Liquid Data Server Settings

2. In the right pane, click the Configuration tab.
3. Click the General tab.

Figure 5-1 Configuring Liquid Data Server Settings

The screenshot shows the 'Id_samples> Liquid Data' administration console. The top status bar indicates 'Connected to localhost:7001', 'Active Domain: Id_samples', and the date 'Tue Oct 08 16:23:10 PDT 2002'. The 'Configuration' tab is selected, with sub-tabs for 'General', 'Data Sources', 'Custom Functions', and 'Import/Export'. The 'General' sub-tab is active, displaying several configuration fields: 'Repository Directory' (set to 'ldrepository'), 'Custom Functions Classpath' (empty), 'Maximum Threads' (set to '20'), 'Security Mode' (checkbox), 'Cache Results' (checkbox), and 'Swap Files Directory' (set to 'temp'). An 'Apply' button is located at the bottom right of the configuration area.

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 the following table.

Table 5-1 Liquid Data Server Configuration Settings

Field	Description
Repository Directory	<p>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 15-3.</p> <p>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.</p> <p>If the specified directory does not exist, Liquid Data will create it automatically.</p>

Table 5-1 Liquid Data Server Configuration Settings

Field	Description
Custom Functions Classpath	<p>Classpath for libraries used by custom functions that do <i>not</i> reside in the <code>custom_lib</code> folder in the server repository.</p> <p>Use semicolons to separate paths. File pathnames or URLs can be used. For example:</p> <pre>c:/path/cf1.jar;c:/path2/cf2.jar</pre> <p>For information on how to configure custom function descriptions, see Chapter 13, “Configuring Access to Custom Functions” and “Server Repository File System Hierarchy” on page 15-3.</p>
Maximum Threads	<p>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.</p> <p>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.</p> <p>Note: The maximum threads value that you specify here <i>does not</i> 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.</p>
Security Mode	<p>Enables or disables (default) the Liquid Data security mode.</p> <ul style="list-style-type: none"> ■ To enable Liquid Data security, select (check) this check box. Once security is enabled, you must configure secure access to Liquid Data resources. ■ To disable Liquid Data security, clear (uncheck) this check box. <p>For more information, see “Enabling Liquid Data Secure Mode” on page 16-8.</p>
Cache Results	<p>Enables or disables (default) the caching of query results for stored queries.</p> <ul style="list-style-type: none"> ■ To enable results caching, enable (check) this check box. ■ To disable results caching, clear (uncheck) this check box. <p>For more information about caching, see “Enabling the Results Cache” on page 18-4.</p>

Table 5-1 Liquid Data Server Configuration Settings

Field	Description
Swap Files Directory	<p>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.</p> <p>If the Large Results flag is selected for the query, then Liquid Data uses swap files to temporarily store intermediate results on disk.</p> <p>Default is <code>temp</code>. This location is a subdirectory of the <code>BEA_HOME/user_projects/domain_name</code> directory.</p> <p>If the specified directory does not exist, Liquid Data will create it automatically.</p> <p>Note: If you change this setting, you must <i>reboot</i> the Liquid Data server in order for the change to take effect.</p>

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**.

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 topic 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](#)
- [Configuring Secure Access to Data Source Descriptions](#)
- [Removing Data Source Descriptions](#)
- [Distributing Data Source Descriptions to Other Liquid Data Servers](#)

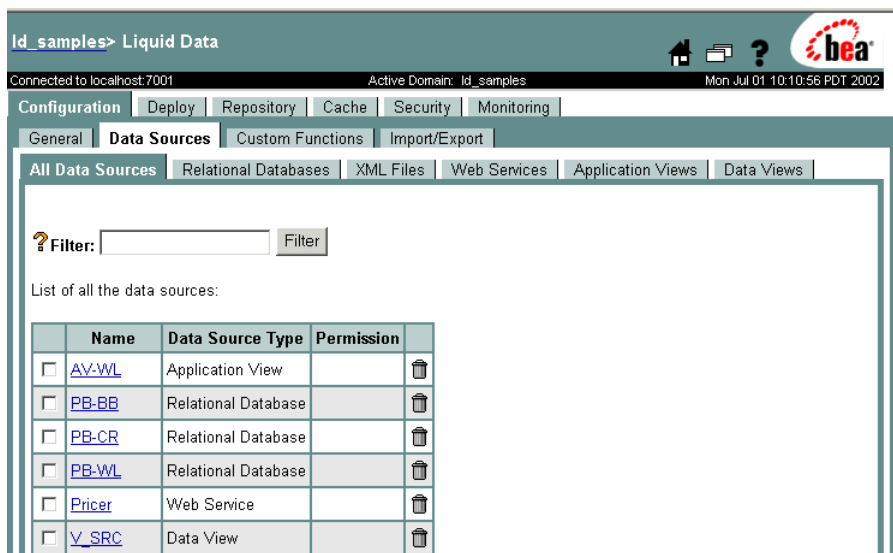
Viewing All Configured Data Sources

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



5. You can filter on a name or a partial name, as described in the following table.

Table 6-1 Viewing All Configured Data Sources

Field	Description
Filter	<p>A simple filter that limits the list of displayed data source descriptions by name. The filter is case-sensitive.</p> <p>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.</p> <p>The Filter field is case-sensitive, does not accept special characters, and does not accept wildcards.</p>
List of All Data Sources	<p>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.</p> <p>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.</p>

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.

For more information about editing data source descriptions, see the topic associated with that data source type. For example, to configure access for relational databases, see [“Creating a Relational Database Data Source Description” on page 7-7](#).

Configuring Secure Access to Data Source Descriptions

If security is enabled on the Liquid Data server, you need to configure security for each data source description by assigning permissions using Access Control Lists (ACLs). Before you assign ACLs, you must define groups, users, and access levels. For more information about Liquid Data security, see [Chapter 16, “Implementing Security.”](#)

6 Viewing and Accessing All Configured Data Sources

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 6-2 Permissions Required for Data Sources

Access Level	Task(s)
read	View a data source description.
modify	Create, edit, or remove data source descriptions.
execute	Run a query that uses the data source.

To assign ACLs to 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 (click) the check box next to the data source description to which you want to assign ACLs.
5. Click Configure ACL.

The Administration Console displays the WebLogic Server ACL configuration page.

Figure 6-2 WebLogic Server ACL Configuration Page



6. To add a new ACL, click on Add a new Permission.
7. The Administration Console displays the Group tab.

Figure 6-3 Group Tab for ACL Configuration

The screenshot shows a web form titled "Group" in a tab. The form contains the following fields:

- ACL:** A text field containing the value "LD.stored_queries.B03-S".
- Permission:** A text field containing the value "MyPermission".
- Grant to Users:** A text field.
- Grant to Groups:** A text field.
- Apply:** A button located at the bottom right of the form.

8. Assign permissions, users, and groups as needed according to the instructions in [“Assigning Permissions, Users, and Groups to ACLs”](#) on page 16-14.

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 before you can remove a data source description. For more information, see [“Administration Console Security”](#) on page 16-2.

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 data view or XML files from the repository, as described in [“Deleting Folders and Files in the Server Repository”](#) on page 15-13.

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 14, “Importing and Exporting Liquid Data Configurations.”](#)

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 topic, 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 topics are included:

- [Creating a JDBC Connection Pool](#)
- [Creating a JDBC Data Source](#)
- [Creating a Relational Database Data Source Description](#)
- [Summary of Configured Data Sources](#)
- [Modifying a Relational Database Source Description](#)
- [Removing a Relational Database Source Description](#)

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	<code>jdbc:pointbase://<hostname>:<portnum>/LDDB</code>	<code>com.pointbase.jdbc.jdbcUniversalDriver</code>
Oracle	<code>jdbc:oracle:thin:@<hostname>:<portnum>:SID</code>	<code>oracle.jdbc.driver.OracleDriver</code>
Microsoft SQL Server	<code>weblogic:mssqlserver4:CRM@<hostname>:<portnum></code>	<code>weblogic.jdbc.mssqlserver4.Driver</code>
Sybase	<code>jdbc:sybase:Tds:<hostname>:<portnum>/<dbname></code>	<code>com.sybase.jdbc.SybDriver</code>
DB2	<code>jdbc:db2://<hostname>/<dbname></code>	<code>COM.ibm.db2.jdbc.net.DB2Driver</code>

Creating a JDBC Connection Pool

For complete information on how to create Java Database Connectivity (JDBC) Connection Pools in WebLogic Server, see [Managing JDBC Connectivity](http://edocs.bea.com/wls/docs70/adminguide/index.html) in the [BEA WebLogic Server Administration Guide](http://edocs.bea.com/wls/docs70/adminguide/index.html) (<http://edocs.bea.com/wls/docs70/adminguide/index.html>).

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	<p>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.</p> <p>For example, we can create a connection pool name for a Wireless data source called MyWireless_POOL.</p>
URL	<p>URL for the database where your data source resides. The URL is passed to the driver to create the physical database connections.</p> <p>For our example Oracle database, we use the following URL:</p> <pre>jdbc:oracle:thin:@<hostname>:1521:SID</pre> <p>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.</p>

Table 7-2 JDBC Connection Pool Configuration

Field	Description
Driver Classname	<p>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.</p> <p>For our Oracle example, we use the following Oracle driver classname:</p> <pre>oracle.jdbc.driver.OracleDriver</pre> <p>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.</p>
Properties	<p>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.</p> <p>The following examples are based on our Broadband, CRM, and Wireless scenario:</p> <ul style="list-style-type: none"> ■ user=broadband; password=broadband ■ user=crm; password=crm ■ user=wireless; password=wireless <p>(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.)</p>
ACL Name	<p>Optional—use only if you are implementing security.</p> <p>Sets the ACL used to control access to this Connection Pool. Permissions available to this ACL are reserve and admin. reserve permission allows users to get logical connections from this Connection Pool. admin allows all other operations on this Connection Pool, including: reset, shrink, shutdown, disable, and enable. Lack of an ACL allows any user open access (provided that user passes other WLS security controls).</p>
Password	<p>Optional—use only if you are implementing security.</p> <p>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 <code>config.xml</code> file and when displayed on the administration console. Use this method to avoid storing passwords in clear text in <code>config.xml</code>.</p>

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 [Managing JDBC Connectivity](#) in the [BEA WebLogic Server Administration Guide](#).

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 (WLS), see [Managing JDBC Connectivity](#) in the [BEA WebLogic Server Administration Guide](#) (<http://edocs.bea.com/wls/docs70/adminguide/index.html>).

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.

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 the following table. (The fields described in the table are required—other optional fields are also displayed on this tab.)

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 <code>MyWirelessDS</code> .
JNDI Name	JNDI path to where this data source is bound. Applications that look up the JNDI path will get a <code>javax.sql.DataSource</code> instance that corresponds to this data source. This can be any name you choose to use for the data source, such as <code>MyWirelessDS</code> .
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 <code>MyWireless_POOL</code> .

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 [“Administration Console Security” on page 16-2](#).

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 7-1 Configuring a Liquid Data Source Description for a Relational Database

The screenshot shows a 'Configure' dialog box with the following fields and controls:

- Name:** Required field (marked with a red asterisk).
- Data Source Name:** Required field (marked with a red asterisk).
- User Name:** Text input field.
- Password:** Text input field.
- Schema:** Text input field.
- Catalog:** Text input field.
- Table Pattern:** Text input field.
- Shared Connection:** Check box.
- Isolation Level:** Dropdown menu with 'TRANSACTION_READ_COMMITTED' selected.
- Maximum Connections:** Text input field.
- Create:** Button at the bottom right.

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

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

Table 7-4 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 .
Username	Not supported.
Password	Not supported.

Table 7-4 Liquid Data Relational Database Data Source Description (Continued)

Field	Description
Schema	<p>Name of the schema you want to use for this Liquid Data data source.</p> <p>The requirements for setting the schema name vary depending on the relational database you are using:</p> <ul style="list-style-type: none"> ■ Oracle—Schema name corresponds to a database name. ■ PointBase—Schema name corresponds to a database name. ■ Microsoft SQL Server—Schema name corresponds to the <i>catalog</i> owner, such as <code>dbo</code>. 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 <code>db2admin</code>. (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.
Catalog	<p>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.</p> <p>The requirements for setting the catalog name vary depending on the RDBMS you are using:</p> <ul style="list-style-type: none"> ■ 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. <p>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.)</p>
Table Pattern	<p>Optional—A pattern used to filter the tables by name.</p> <p>Special characters are:</p> <ul style="list-style-type: none"> ■ <code>_</code> An underscore character is used to match any single character ■ <code>%</code> A percent sign is used to match of zero or more characters.

Table 7-4 Liquid Data Relational Database Data Source Description (Continued)

Field	Description
Shared Connection	<p>Toggle to set <i>shared connection</i> on (checked) or off (cleared).</p> <ul style="list-style-type: none"> ■ 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.
Isolation Level	<p>Sets the transaction isolation level.</p> <ul style="list-style-type: none"> ■ 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: <ul style="list-style-type: none"> 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. <p>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.</p>
Maximum Connections	<p>Optional—Specifies maximum number of connections 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.</p>

7. Click Create.

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

Note: If security is enabled on your Liquid Data server, 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.
- Configure security on a data source by selecting the data source and clicking the Configure ACL link. The Configure ACL link takes you to the Compatibility Security ACL configuration tabs. For more information about ACLs, 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 change 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 [“Administration Console Security” on page 16-2](#).

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 [“Administration Console Security” on page 16-2](#).

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.

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 15-8](#). Once configured according to the instructions in this topic, 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 topics 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 [“Administration Console Security” on page 16-2](#).

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 8-1 Configuring a Liquid Data Source Description for an XML File

The screenshot shows the BEA Administration Console interface. The breadcrumb navigation at the top reads: `Id_samples > Liquid Data > Configuration > Data Sources > XML Files > Configure a XML File Data Source Description`. Below the breadcrumb, there is a status bar with the text: `Connected to localhost:7001`, `Active Domain: Id_samples`, and `Fri Jun 21 18:11:45 PDT 2002`. The main content area is titled **Configure** and contains three required fields, each marked with a red asterisk:

- Name:** A text input field.
- Data File:** A text input field with a [Browse Repository](#) link to its right.
- Schema File:** A text input field with a [Browse Repository](#) link to its right.

A **Create** button is located at the bottom right of the configuration area.

6. Fill in the fields as detailed in the following table.

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

Field	Description
Name	Logical name of the XML file.

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

Field	Description
Data File	<p>XML data file. One of the following formats:</p> <ul style="list-style-type: none"> ■ 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 15, “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
Schema	Schema for the XML file in the server repository. Enter the file name, or click Browse Repository to select it.

7. Click Create.

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

Note: If security is enabled on your Liquid Data server, 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 15-13](#).

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.

- Configure security on a data source by selecting the data source and clicking the Configure ACL link. The Configure ACL link takes you to the Compatibility Security ACL configuration tabs. 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 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 [“Administration Console Security”](#) on page 16-2.

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 15-13](#).

Note: You must log in with `modify` access before you can remove a data source description. For more information, see [“Administration Console Security” on page 16-2](#).

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.

9 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 topic, 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 topics 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.

9 Configuring Access to Web Services

Note: You must log in with `modify` access before you can create a data source description. For more information, see “Administration Console Security” on page 16-2.

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 9-1 Configuring a Liquid Data Source Description for a Web Service

The screenshot shows the Administration Console interface. The breadcrumb navigation at the top reads: **Id_samples > Liquid Data > Configuration > Data Sources > Web Services > Configure a Web Service Data Source Description**. Below the breadcrumbs, a status bar indicates: **Connected to localhost:7001**, **Active Domain: Id_samples**, and **Fri Jun 21 18:12:56 PDT 2002**. The main content area is titled **Configure** and contains three required fields, each with a question mark icon and an asterisk: **Name:**, **Definition (WSDL):**, and **Operations:**. Each field has a corresponding text input box. To the right of the **Definition (WSDL):** field is a [Browse Repository](#) link. A **Create** button is located at the bottom right of the configuration area.

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

Table 9-1 Liquid Data Web Service Data Source Description Configuration

Field	Description
Name	Logical name of the Web service.

Table 9-1 Liquid Data Web Service Data Source Description Configuration

Field	Description
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: <code>getSalesPrices,getSalesDiscount</code>

- Click Create.

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

Note: If security is enabled on your Liquid Data server, 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.
- Configure security on a data source by selecting the data source and clicking the **Configure ACL** link. The Configure ACL link takes you to the Compatibility Security ACL configuration tabs. 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 [“Administration Console Security” on page 16-2.](#)

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 [“Administration Console Security” on page 16-2](#).

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.

10 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 topic, 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 *Deploying Liquid Data*.
- 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, refer to *Using Application Integration* (<http://edocs.bea.com/wli/docs70/interm/aihome.htm>) in the WebLogic Integration documentation. A good starting point is *Introduction to Application Integration*.

The following topics 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 a New WebLogic Domain](#)” in *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 a Domain](#)” on page 2-3 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 refer to 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 10-3](#), 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 [“Administration Console Security” on page 16-2](#).

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.

- Click the Application View tab.
- 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 10-1 Configuring a Liquid Data Source Description for an Application View

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

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

Configure

? Name: *

? Application View Name: * [Browse Repository](#)

? Host: *

? Port: *

? User Name:

? Password:

? Operations:

Create

- Fill in the fields as detailed in the following table.

Table 10-1 Liquid Data Application View Data Source Description

Field	Description
Name	Logical name of the Liquid Data application view data source.
Application View Name	Name of the application view as defined in the WebLogic Integration Application View Console.
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 10-3.

10 *Configuring Access to Application Views*

Table 10-1 Liquid Data Application View Data Source Description (Continued)

Field	Description
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.

7. Click Create.

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

Note: If security is enabled on your Liquid Data server, 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.
- Configure security on a data source by selecting the data source and clicking the **Configure ACL** link. The Configure ACL link takes you to the Compatibility Security ACL configuration tabs. 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 10-3.](#)

Note: You must log in with `modify` access before you can modify a data source description. For more information, see [“Administration Console Security” on page 16-2.](#)

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 [“Administration Console Security” on page 16-2](#).

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.

11 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. For instructions on how to create data views, see [“Creating Data Views from Stored Queries” on page 15-16](#).

Once configured according to the instructions in this topic, 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 topics are included:

- [Creating a Data View Data Source Description](#)
- [Summary of Configured Data Sources](#)
- [Modifying a Data View Data Source Description](#)
- [Removing a Data View Data Source Description](#)

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 that tells Liquid Data how to find the data view.

Notes: You must log in with `modify` access before you can create a data source description. For more information, see [“Administration Console Security” on page 16-2](#).

There are two ways to create a data view data source description—using the Data View tab on the Liquid Data node, as described in this section, or using the Create Data View link in the server repository, as described in [“Creating Data Views from Stored Queries” on page 15-16](#).

To create a data source description for a data 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 Data Views tab.
5. Click the Configure a new Data View source description text link.

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

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

The screenshot shows a 'Configure' window with a light blue border. Inside, there are three rows of configuration fields. Each row starts with a question mark icon, followed by a label, a red asterisk indicating a required field, and a text input box. The first row is 'Name', the second is 'Query File', and the third is 'Schema File'. To the right of each input box is a blue hyperlink labeled 'Browse Repository'. At the bottom right of the window is a grey button labeled 'Create'.

6. Fill in the fields as detailed in the following table.

Table 11-1 Liquid Data Data View Configuration

Field	Description
Name	Logical name of the data view. You can use any meaningful name you want. The query uses this name to reference the data source. Required.
Query File	Name of a stored query file in the repository, created either in the Data View Builder or a hand-coded query. Enter the query file name or click Browse Repository to select it. Required.
Schema	Schema for the query. Enter the schema file name or click Browse Repository to select it. Required.

7. Click Create.

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

Note: If security is enabled on your Liquid Data server, 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 data view files in the repository, as described in [“Configuring Secure Access to Items in the Server Repository” on page 15-13](#).

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.
- Configure security on a data source by selecting the data source and clicking the **Configure ACL** link. The Configure ACL link takes you to the Compatibility Security ACL configuration tabs. 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 Data View Data Source Description

You can modify an existing data view data source description.

Note: You must log in with `modify` access before you can modify a data source description. For more information, see [“Administration Console Security” on page 16-2](#).

To modify an existing data source description for a data 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 Data Views 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 Data 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 data view to which it refers. To explicitly remove the data view file from the repository, see [“Deleting Folders and Files in the Server Repository” on page 15-13](#).

Note: You must log in with `modify` access before you can remove a data source description. For more information, see [“Administration Console Security” on page 16-2](#).

To remove a data source description for a data 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 Data Views tab.
A table of configured Liquid Data data views is shown.
5. Find the data view that you want to remove and click the trash can next to it.
6. When prompted, click Yes to confirm removal.

11 *Configuring Access to Data Views*

The Administration Console removes the selected data source description.

12 Deploying Liquid Data Components

This topic describes how to 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 topic describes how to deploy core Liquid Data software components only. There are additional tasks required to deploy other Liquid Data components—such as the distributing the Liquid Data Server repository, distributing Liquid Data configuration settings, and so on. For a summary of all deployment tasks, see [“Deployment Tasks”](#) in *Deploying Liquid Data*.

Liquid Data Components to Deploy

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

Table 12-1 Contents of the LDS.ear File

Component	Description
<code>ejb_query.jar</code>	EJB that contains all query-related classes, including the Query Processor and stateless session bean. For more information, see the Liquid Data Javadoc .
<code>ejb_qbc.jar</code>	EJB for Data View Builder to obtain configuration information from the Liquid Data Server.
<code>XMediator.war</code>	Initializes the Liquid Data Server.
<code>ldconsole.war</code>	Liquid Data configuration tabs that appear in the WebLogic Server Administration Console.
<code>cacheEjb.jar</code>	EJB that manages results caching for stored queries that are configured for results caching. For more information, see Appendix 18, “Configuring the Query Results Cache.”
<code>ldcacheListener.war</code>	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.

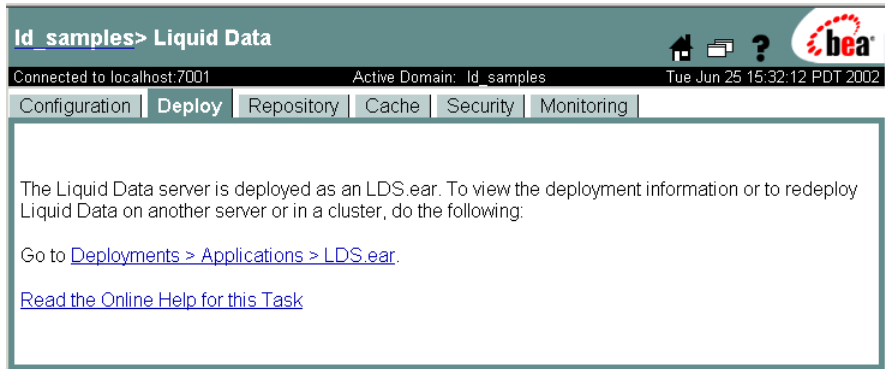
Navigating to the Deploy Tab

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.

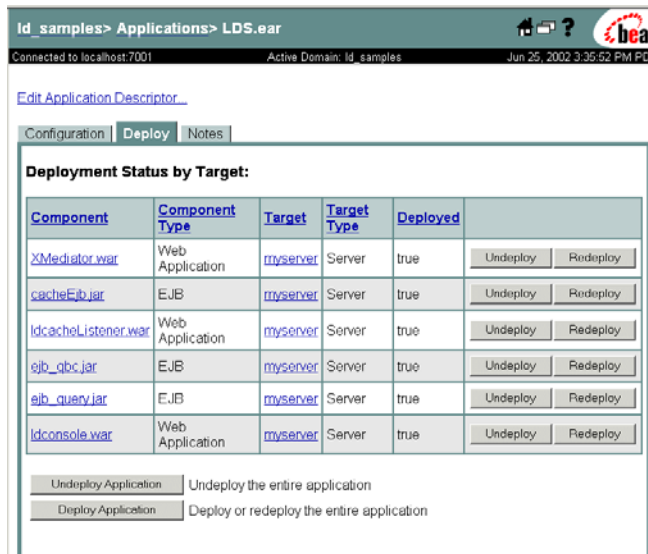
Figure 12-1 Deploy Tab on the Liquid Data Node in the Administration Console



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 12-2 WebLogic Deploy Tab



4. Select target servers and deploy or undeploy as needed. For instructions, see [“Deployment Tasks”](#) in *Deploying Liquid Data*.

13 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 topic describes how to configure access to custom functions. It includes the following topics:

- [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 topic includes the following sections:

- [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 topic, 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 15-8](#).
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 15-8](#).
4. For each custom function, create a custom function description, as described in [“Creating a Custom Function Description” on page 13-4](#).
5. If security is enabled, assign ACLs to the custom function description, as described in [“Configuring Secure Access to Custom Function Descriptions” on page 13-6](#), and also assigning ACLs 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 15-13](#).

Creating a Custom Function Description

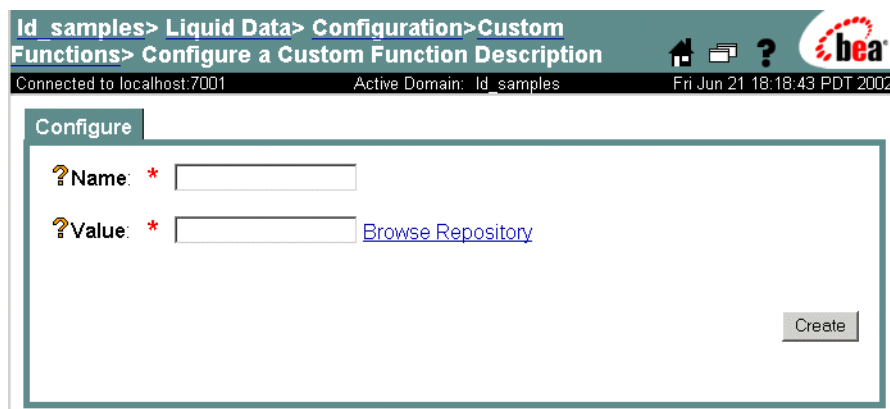
Note: You must log in with `modify` access before you can create a custom function description. For more information, see [“Administration Console Security” on page 16-2](#).

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 13-1 Configuring a Custom Function Description



The screenshot shows the Administration Console interface. The breadcrumb navigation at the top reads: `Id_samples > Liquid Data > Configuration > Custom Functions > Configure a Custom Function Description`. Below the breadcrumb, there is a status bar with three items: `Connected to localhost:7001`, `Active Domain: Id_samples`, and `Fri Jun 21 18:18:43 PDT 2002`. On the right side of the status bar are icons for home, a folder, a question mark, and the BEA logo. The main content area is titled 'Configure' and contains two required text input fields: `? Name: *` and `? Value: *`. To the right of the 'Value' field is a blue hyperlink labeled `Browse Repository`. At the bottom right of the form is a grey button labeled `Create`.

6. Enter the information described in the following table:

Table 13-1 Custom Function Description Information

Field	Description
Name	Logical name of the group of custom functions declared in the custom functions library definition (CFLD) file.
Value	<p>File name of the CFLD file that declares this custom function in an XML format. This file usually resides in the <code>custom_functions</code> folder in the Liquid Data Server repository, which is described in “Server Repository File System Hierarchy” on page 15-3.</p> <p>Either type the CFLD file name or click Browse Repository to browse the <code>custom_functions</code> folder and select it.</p>

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.
- Configure security on a custom function group by selecting the group name and clicking the **Configure ACL** link. The Configure ACL link takes you to the Compatibility Security ACL configuration tabs. For more information, see [“Configuring Secure Access to Custom Function Descriptions” on page 13-6](#).
- 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 ACLs to a custom function description. For more information, see [“Administration Console Security” on page 16-2](#).

If security is enabled on the Liquid Data server, you need to configure security for each custom function description using Access Control Lists (ACLs). You need to assign `execute` permissions to users who are authorized to execute queries that use particular custom functions. Before you assign ACLs, you must define groups, users, and access levels. For more information about Liquid Data security, see [Chapter 16, “Implementing Security.”](#)

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 ACLs assigned in the Custom Functions tab determine whether a user can execute a query in which a custom function is used. The ACLs 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 ACLs to a custom function description:

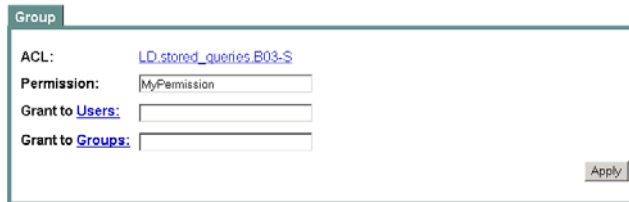
1. In the left pane, click the Liquid Data node.
2. In the right pane, click the Configuration tab.
3. Click the Custom Functions tab.
4. Select (check) the check box next to the custom function description to which you want to assign ACLs.
5. Click Configure ACL.
6. The Administration Console displays the WebLogic Server ACL configuration page.

Figure 13-2 WebLogic Server ACL Configuration Page



7. To add a new ACL, click on Add a new Permission.
8. The Administration Console displays the Group tab.

Figure 13-3 Group Tab for ACL Configuration



9. Assign `execute` permissions, users, and groups as needed according to the instructions in “[Assigning Permissions, Users, and Groups to ACLs](#)” on page 16-14.

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 “[Administration Console Security](#)” on page 16-2.

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 [“Administration Console Security” on page 16-2](#).

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 15-13](#).

14 Importing and Exporting Liquid Data Configurations

This topic 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 *Deploying Liquid Data*.

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 14-1 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 18, “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 13, “Configuring Access to Custom Functions.”

Note: For information about what is not included, see [“What Liquid Data Does Not Import or Export” on page 14-3.](#)

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 *Deploying Liquid Data*.

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 15-7](#).
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 15-8](#).

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 14-1 Import /Export Tab on the Liquid Data Node



Table 14-2 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 14-7 .
Export tab	Used to export a Liquid Data configuration to a file, as described in “Exporting a Liquid Data Configuration” on page 14-5 .

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.

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 14-4](#).
2. In the Import / Export tab, click Export.

The Administration Console displays the Export tab.

Figure 14-2 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: LDRootimport_export \

Folder icon

Name

Export File Name :

Export

The Export tab includes the following information:

Table 14-3 Export Configuration Information

Field	Description
Path	Shows the file system path to the <code>import_export</code> 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.

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 5-1](#).

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 15-8](#).

To import a Liquid Data configuration:

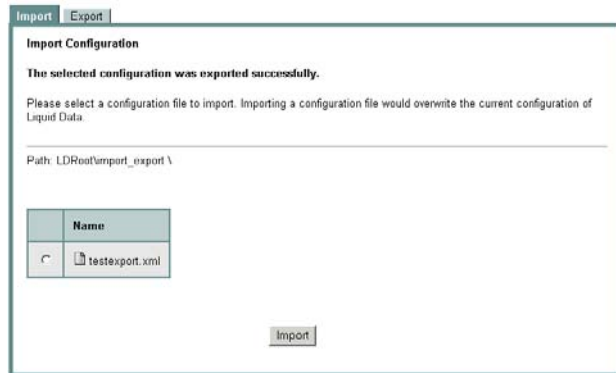
1. Complete the following tasks on the target Liquid Data server, depending on the type of data sources required:
 - For relational databases, you need to configure JDBC connectivity, as described in [“Creating a JDBC Connection Pool” on page 7-2](#). As an alternative to using the Administration Console, 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 *Deploying Liquid Data*.

14 Importing and Exporting Liquid Data Configurations

- 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 10-3.
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 15-8.
 3. On the target Liquid Data server, navigate to the Import / Export tab, as described in [“Navigating to the Import/Export Tab”](#) on page 14-4.
 4. In the Import / Export tab, click Import.

The Administration Console displays the Import tab.

Figure 14-3 Import Tab on the Liquid Data Node



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

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

15 Managing the Liquid Data Server Repository

This topic 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](#)
- [Creating Data Views from Stored Queries](#)
- [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 16, “Implementing Security.”](#)

About the Liquid Data Server Repository

This topic describes the server repository. It contains the following sections:

- [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

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:

```
BEA_HOME\user_projects\domainName\repositoryRootDir
```

where

- *domainName* is the domain name associated with the Liquid Data Server
- *repositoryRootDir* is the root directory of the server repository

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 *Deploying Liquid Data*.

Server Repository File System Hierarchy

The server repository root directory contains the following folders:

Table 15-1 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 <i>Invoking Queries Programmatically</i> and Chapter 13, “Configuring Access to Custom Functions.”

15 Managing the Liquid Data Server Repository

Table 15-1 File System Hierarchy of the Server Repository (Continued)

Folder	Contents
custom_lib	Java Archive (JAR) files containing the Java implementations of custom functions. For more information, see Using Custom Functions in <i>Invoking Queries Programmatically</i> and Chapter 13, “Configuring Access to Custom Functions.”
data_views	Stored data view (XV) files created using the Data View Builder. For more information, see Chapter 11, “Configuring Access to Data Views” and Designing Queries in <i>Building Queries and Data Views</i> .
dtlds	Not supported. Source document type definition (DTD) files. Source DTD files are associated with the XML data files stored in the <code>xml_files</code> folder. For more information, see Chapter 8, “Configuring Access to XML Files.”
import_export	Exported Liquid Data configuration files. For more information, see Chapter 14, “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 <code>xml_files</code> folder. For more information, see Chapter 8, “Configuring Access to XML Files.”
stored_queries	Stored query (XQ) files created using the Data View Builder. For more information, see Designing Queries in <i>Building Queries and Data Views</i> .
web_services	Web Services Description Language (WSDL) files for Web services used as data sources. For more information, see Chapter 9, “Configuring Access to Web Services.”
web_services_gen	Application archive (EAR) files of Web services that have been published through Liquid Data. For more information, see Chapter 19, “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 *Deploying Liquid Data*.

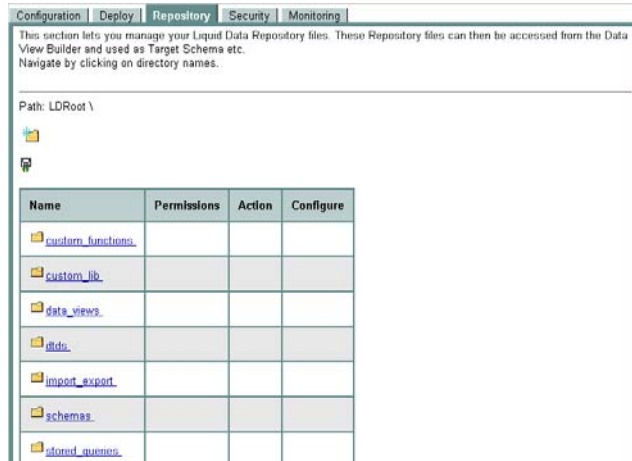
Navigating to the Repository Tab

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 15-3](#).

Figure 15-1 Repository Tab on the Liquid Data Node



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 [“Administration Console Security” on page 16-2](#).

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 ACL) that appears on the same row as the item you want to modify.

Working with Folders and Files in the Server Repository

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

- [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 topic, 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 15-8](#).

Note: You must log into the Administration Console with at least `read` access to download files from the server repository. For more information, see [“Administration Console Security” on page 16-2](#).

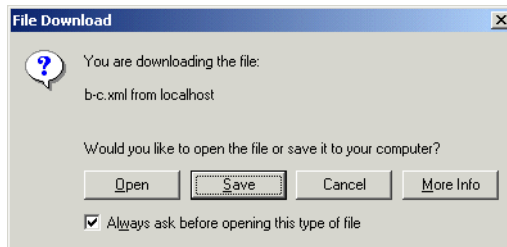
To download a file from the server repository:

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 15-2 Download Action Prompt



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 [“Administration Console Security” on page 16-2](#).

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.

Table 15-2 Uploading a File To the Repository

Field	Description
Local File	Name of the file to upload.

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.

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 would assign separate ACLs to each folder to ensure that only authorized users can access these queries, as described in [“Assigning ACLs to Liquid Data Resources” on page 16-13](#).

Note: You must log into the Administration Console with at least `modify` access to create a folder. For more information, see [“Administration Console Security” on page 16-2](#).

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.

Table 15-3 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.

3. Enter the name of the new folder.
4. Click Create.

The Administration Console creates the specified folder.

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 [“Administration Console Security” on page 16-2](#).

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.

Table 15-4 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.

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.

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 [“Administration Console Security” on page 16-2](#).

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 15-10](#).

2. Next to the folder or file to rename, click **Rename**.
The Administration Console prompts you to specify a different name.

Table 15-5 Renaming a Folder or File in the Repository

Field	Description
New Name	<p>New name for the selected folder or file.</p> <p>The name must comply with the naming standards of your file system.</p> <ul style="list-style-type: none">■ 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. <p>Note: Do not change the filename extension for files, such as stored queries or data views.</p>

3. Enter the new name of the item.
4. Click **Rename**.

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

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 [“Administration Console Security” on page 16-2](#).

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 15-10](#).

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 18, “Configuring the Query Results Cache.”](#)

Configuring Secure Access to Items in the Server Repository

If security is enabled on the Liquid Data server, you must explicitly configure security for all items in the server repository. To configure security, you assign permissions to folders and files using Access Control Lists (ACLs). At a minimum, you need to assign ACLs to the default repository folders. In addition, you might want to assign ACLs to individual files or for sub-folders in the default folders. Before you assign ACLs, you must define groups, users, and access levels in the Administration Console, as described in [Chapter 16, “Implementing Security.”](#)

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 15-6 Permissions Required for Server Repository Items in the Administration Console

Access Level	Task(s)
read	View, download, or copy a folder or file in the repository.
modify	Create, edit, rename, paste, delete, or upload a folder or file in the repository.
execute	Execute a stored query.

To assign ACLs to an item in the repository:

1. Browse the server repository to find the folder or file to which you want to assign ACLs.
2. Next to the folder or file you want, click ACL.

The Administration Console prompts you to specify the ACL Name.

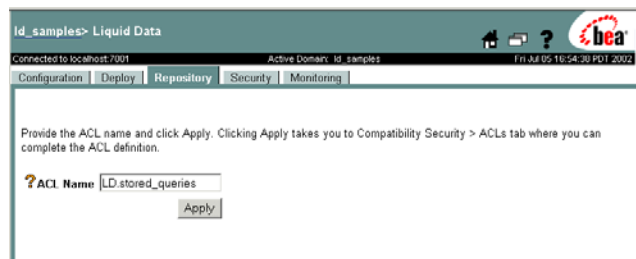


Table 15-7 Assigning an ACL to a Folder or File in the Repository

Field	Description
ACL Name	<p>The ACL name for the selected folder or file. By default, the Administration Console supplies a default ACL name using the following naming convention:</p> <p><code>LD.folderName.fileName</code></p> <p>where</p> <ul style="list-style-type: none"> ■ <i>folderName</i> is either the name of the folder to which to assign ACLs, or the name of the folder containing the item (folder or file) to which to assign ACLs. ■ <i>fileName</i> is the name of the filename, if assigning ACLs to a particular file.

3. Click Apply.

The Administration Console displays the WebLogic Server ACL page.



4. To add a new ACL, click on Add a new Permission.

The Administration Console displays the Group tab.

Figure 15-3 Group Tab for ACL Configuration

5. Assign permissions, users, and groups as needed according to the instructions in “Assigning Permissions, Users, and Groups to ACLs” on page 16-14.

Creating Data Views from Stored Queries

Before a Liquid Data 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. To create a data view, you need to configure the stored query using the Repository tab on the Liquid Data node in the Administration Console. For more information about data views, see [Chapter 11](#), “Configuring Access to Data Views” and “Using Views as Data Sources” in *Building Queries and Data Views*.

Note: There are two ways to create a data view data source description—using the Create Data View link in the server repository, as described in this section, or using the Data View tab on the Liquid Data node, as described in “[Creating a Data View Data Source Description](#)” on page 11-2.

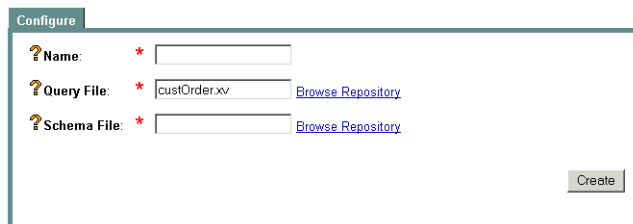
You can create a data view:

1. Browse the `stored_queries` folder in the server repository and select the stored query that you want to use for the data view.

Note: The stored query cannot already have an associated data view.

2. Click Data View Data Source.

The Administration Console copies the selected stored query to the `data_views` folder, assigns an `xv` extension to the file name, and displays the Data View configuration tab.



The screenshot shows a 'Configure' dialog box with a light blue border. At the top left is a tab labeled 'Configure'. Inside the dialog, there are three rows of configuration fields, each starting with a question mark icon and a red asterisk. The first row is 'Name' with an empty text box. The second row is 'Query File' with a text box containing 'custOrder.xv' and a blue 'Browse Repository' link to its right. The third row is 'Schema File' with an empty text box and a blue 'Browse Repository' link to its right. In the bottom right corner of the dialog is a grey 'Create' button.

3. Enter the following information:

Table 15-8 Liquid Data Data View Configuration

Field	Description
Name	Logical name of the data view. You can use any meaningful name you want. The query uses this name to reference the data source. Required.
Query File	Name of a stored query file in the repository, created either in the Data View Builder or a hand-coded query. Enter the query file name or click Browse Repository to select it. Required.
Schema	Schema for the query. Enter the schema file name or click Browse Repository to select it. Required.

4. Click Create.

The Administration Console displays a list of data views configured as data sources, including the data view you just created.

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 18, “Configuring the Query Results Cache.”](#)

16 Implementing Security

This section describes how to implement security in BEA Liquid Data for WebLogic™. It contains the following sections:

- [Overview of Liquid Data Security](#)
- [Liquid Data Server Security Implementation Process](#)
 - [Initial Security Setup](#)
 - [Defining a Compatibility Security Realm](#)
 - [Enabling Liquid Data Secure Mode](#)
 - [Configuring Liquid Data Users](#)
 - [Configuring Liquid Data Groups](#)
 - [Assigning ACLs to Liquid Data Resources](#)
- [Integrating Liquid Data Security With BEA Other Software](#)

Overview of Liquid Data Security

WebLogic Server provides the foundation for Liquid Data security. Liquid Data deployments can use the compatibility security features that WebLogic Server provides, including compatibility security realms, users and groups, Access Control Lists (ACLs) and permissions, the Secure Sockets Layer (SSL) protocol, authentication mechanisms, digital certificates, controlled access to resources, and so on.

The Administration Console provides configurable security mechanisms to prevent unauthorized access to core Liquid Data functionality. For complete information about using the Administration Console to manage WebLogic security, see [“Using Compatibility Security”](#) in *Managing WebLogic Security* in the WebLogic Server documentation.

By default, Liquid Data runs in non-secure mode. If you want to use WebLogic Server security features, you must explicitly enable Liquid Data security, as described in [“Enabling Liquid Data Secure Mode”](#) on page 16-8. Once enabled, system administrators use the Administration Console to manage Liquid Data security, which includes the following tasks:

- Setting up compatibility security realms, groups, and users
- Configuring access control lists (ACLs) for data sources and queries

Administration Console Security

The Administration Console requires a valid user login and selectively permits access to operations based on user roles. The following table describes the permissions required to perform certain Liquid Data security management tasks in the Administration Console:

Table 16-1 Permissions Required for Administration Console Tasks

Permission	Task
modify	<ul style="list-style-type: none"> ■ Creating, modifying, and deleting data source descriptions. ■ Creating, modifying, and deleting directories and files in the repository.
read	<ul style="list-style-type: none"> ■ Viewing the contents of an item in the repository. ■ Any user who can log into the Administration Console has <code>read</code> permission to view data source descriptions and the repository structure.

Data Access Security

If security is enabled for your Liquid Data Server installation, you *must* configure an ACL for the following resources: stored queries, data sources, repository directories, and file security. For more information, see [“Assigning ACLs to Liquid Data Resources” on page 16-13](#).

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 ACL associated with the data source.

Stored Queries

For stored queries, access is determined by the ACL associated with the file name of the stored query. The ACL is assigned to the stored query in the Administration Console, as described in [“Assigning ACLs to Liquid Data Resources” on page 16-13](#). 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 Query Engine for processing.

ACL names for stored queries use the following format:

```
LD.stored_queries.[subfolderName.]queryName
```

where

- `stored_queries` is the name of the folder in the server repository that contains stored queries.
- `subfolderName` is the name of the sub-folder that contains stored queries, if applicable.
- `queryName` is the file name of the stored query.

Ad Hoc Queries

For ad hoc queries, access is determined by the ACL associated with any data source(s) that the user attempts to use, as described in [“Data Source Security” on page 16-4](#). The ACL 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 ACL associated with the data source description. The ACL 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 in the Properties, ACL, and Password fields. For more information, see [“Creating a JDBC Connection Pool” on page 7-2](#).
- 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 10-3](#).
 - Configuring the application pool username and password in the Administration Console, as described in [“Configuring an Application View Data Source Description” on page 10-4](#).

- For data views used as data sources, access is determined by ACLs associated with the underlying query and data sources.

Repository Directory and File Security

Access to directories and files in the repository can be controlled by assigning ACLs to individual directories or files using the Administration Console. ACLs can be assigned to stored queries, data views, XML files, web service definitions, and custom functions. For more information, see [“Configuring Secure Access to Items in the Server Repository” on page 15-13](#).

ACL names for folders and files in the server repository use the following format:

`LD.folderName[.fileName]`

where

- *folderName* is either the name of the folder to which to assign ACLs, or the name of the folder containing the item (folder or file) to which to assign ACLs.
- *fileName* is the name of the filename, if assigning ACLs to a particular file.

Liquid Data Server Security Implementation Process

To implement Liquid Data security, you perform the following tasks in the Administration Console:

1. Complete the initial security setup, as described in [“Initial Security Setup” on page 16-6](#).
2. Set up a compatibility security realm, as described in [“Defining a Compatibility Security Realm” on page 16-7](#).
3. Activate secure mode for Liquid Data according to the instructions in [“Enabling Liquid Data Secure Mode” on page 16-8](#).
4. Add the `LDAdmin` group according to the instructions in [“Configuring Liquid Data Groups” on page 16-10](#).

5. Add users associated with each of these groups according to the instructions in [“Configuring Liquid Data Users” on page 16-9](#).
6. Assign ACLs to data sources, stored queries, and repository files and directories, according to the instructions in [“Assigning ACLs to Liquid Data Resources” on page 16-13](#).

Initial Security Setup

When deploying Liquid Data in a domain, you need to perform the following steps to set up Liquid Data security:

1. Start the Administration Console on the domain on which you are deploying Liquid Data, logging in with the username/password you had created when the domain was initially configured.
2. In the left pane, click on *domain_name*->Compatibility Security->Users and add a new user (*ldsystem*).
3. Click on *domain_name*->Compatibility Security->Groups and add a new group (*LDAdmin*).
4. Add the *ldsystem* user to the *LDAdmin* group.
Note: To add members to the *LDAdmin* group, you must add them as *users*, not as groups.
5. Add the *ldsystem* user to the *Administrators* group.
6. Click on *domain_name*->Compatibility Security->ACLs and add a new ACL (*LD*).
7. Add the following new permission attributes to the Liquid Data ACL: *execute*, *read*, and *modify*, and then add the *Administrators* group to the grantee list of permissions.

Defining a Compatibility Security Realm

A WebLogic Server *compatibility security realm* is a domain for a set of security features that provide access to ACLs, names of principals, and related security services. A realm provides a context in which the range of security operations and other security-related information governing Liquid Data users is defined. It determines how users are authenticated. The security features available for WebLogic Integration are built on the security functionality provided by WebLogic Server.

WebLogic Server provides the following types of compatibility security realms:

Table 16-2 Types of WebLogic Server Compatibility Security Realms

Realm Type	Description
File realm	Stores all user and group data for the File realm in the <code>fileRealm.properties</code> file.
Caching realm	Works with the File realm, alternate security realms, or custom security realms to fulfill client requests with the proper authentication and authorization.
LDAP security realm	Provides authentication through a Lightweight Directory Access Protocol (LDAP) server. This server allows you to manage all the users for your organization in one place: the LDAP directory.
Windows NT security realm	Uses account information defined for a Windows NT domain to authenticate users and groups. You can view users and groups in the Windows NT Security realm through the Administration Console, but you must manage users and groups through the facilities provided by Windows NT.
UNIX security realm	Executes a small native program, <code>wlauth</code> , to look up users and groups and to authenticate users on the basis of their UNIX login names and passwords. Runs only on the Solaris and Linux platforms.
RDBMS security realm	BEA-provided custom security realm that stores users, groups and ACLs in a relational database.

Table 16-2 Types of WebLogic Server Compatibility Security Realms

Realm Type	Description
Custom security realm	Custom built security realm that draws from an existing store of users such as directory server on the network. Requires an implementation of the <code>weblogic.security.acl.AbstractListableRealm</code> interface or the <code>weblogic.security.acl.AbstractManageableRealm</code> interface and uses the Administration Console to install this implementation.

The realm type you choose depends on the particular needs of your Liquid Data implementation. For example, the `ld_samples` domain uses a file realm defined in a `filerealm.properties` file.

To set up a compatibility security realm for Liquid Data, follow the detailed configuration instructions for your realm type in “Specifying a Security Realm” in [“Using Compatibility Security”](#) in *Managing WebLogic Security* in the WebLogic Server documentation.

Enabling Liquid Data Secure Mode

By default, Liquid Data runs in non-secure mode. To implement Liquid Data security, you must explicitly enable the security mode in the General tab on the Liquid Data node in the Administration Console.

To enable Liquid Data security:

1. In the left pane of the Administration Console, expand the Servers node until the Liquid Data node is displayed.
2. Click the Liquid Data node.
3. In the right pane, click the Configuration tab.
4. Click the General tab.

Figure 16-1 Enabling Liquid Data Security in the General Tab On the Liquid Data Node

The screenshot shows the 'Configuration' window with the 'General' tab selected. The 'Security Mode' checkbox is checked. The 'Repository Directory' is set to 'ldrepository', 'Custom Functions Classpath' is empty, 'Maximum Threads' is 20, 'Cache Results' is unchecked, and 'Swap Files Directory' is 'temp'. An 'Apply' button is in the bottom right.

Configuration	Deploy	Repository	Security	Monitoring
General	Data Sources	Custom Functions	Import/Export	

? Repository Directory:

? Custom Functions Classpath:

? Maximum Threads:

? Security Mode: ☒

? Cache Results: ☐

! ? Swap Files Directory:

Apply

5. On the General tab, select (check) Security Mode, as shown in the previous figure.
6. Click Apply.

Once security is enabled, you must explicitly define users, groups, and access control lists to Liquid Data resources. The Administration Console displays ACL hyperlinks that you can use to assign ACLs to associated resources. For more information, see:

- [“Configuring Liquid Data Users” on page 16-9](#)
- [“Configuring Liquid Data Groups” on page 16-10](#)
- [“Assigning ACLs to Liquid Data Resources” on page 16-13](#)

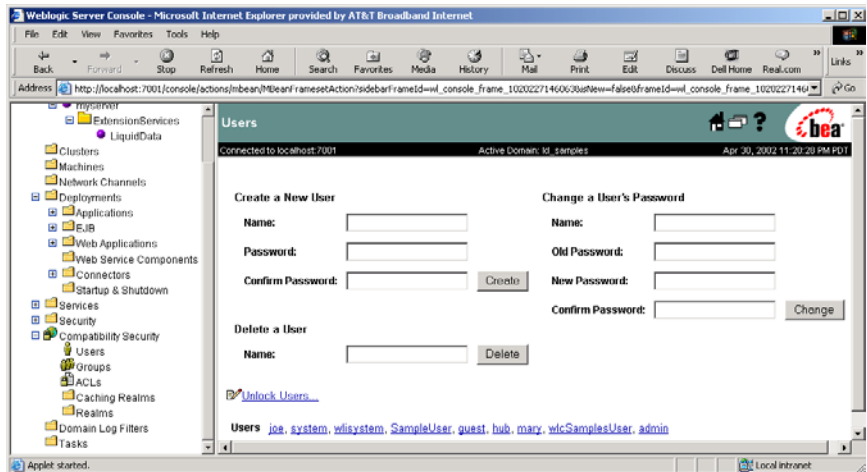
Configuring Liquid Data Users

You configure Liquid Data users using the Administration Console. At a minimum, you must define the `ldsystem` user according to the instructions in [“Initial Security Setup” on page 16-6](#). For detailed instructions about configuring users, see “Defining Users in the Compatibility Realm” in [“Using Compatibility Security”](#) in *Managing WebLogic Security* in the WebLogic Server documentation.

To configure Liquid Data users:

1. In the left pane of the Administration Console, click on the Compatibility Security node.
2. Click on the Users link.

Figure 16-2 Configuring Liquid Data Users



3. On the Users page, in the Create a New User section, enter the name and password of the user you want to add.
4. Click Create.

Configuring Liquid Data Groups

This topic describes how to configure groups for Liquid Data. It contains the following sections:

- [Group to Add for Liquid Data](#)
- [Configuring Groups](#)

You configure Liquid Data groups using the Administration Console. For more information, see “Defining Groups in the Compatibility Realm” in “[Using Compatibility Security](#)” in *Managing WebLogic Security* in the WebLogic Server documentation.

Group to Add for Liquid Data

For Liquid Data, you need to add the `LDAdmin` group, which allows Liquid Data administrators to perform the following tasks:

- Configure data sources
- Manage the server repository
- Assign ACLs
- Run queries

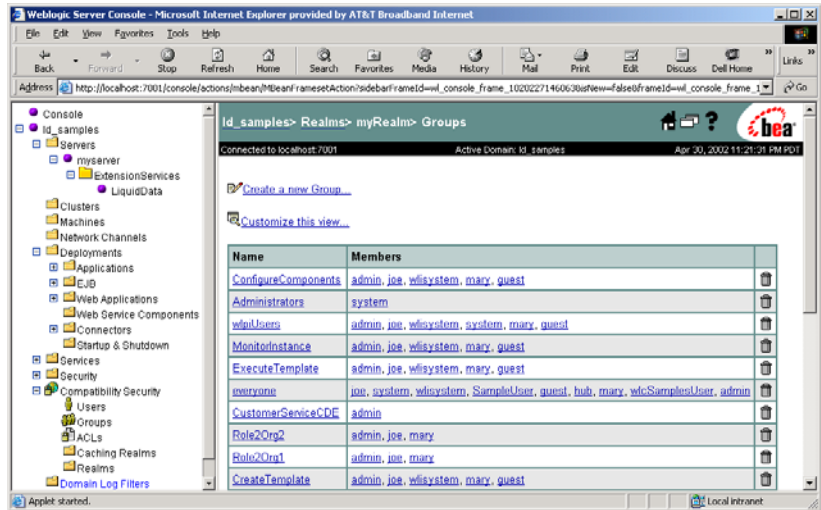
This group is added during [“Initial Security Setup” on page 16-6](#). This group needs to be mapped to the `Administrator` group (and must never be unmapped from that group).

Configuring Groups

To configure Liquid Data groups:

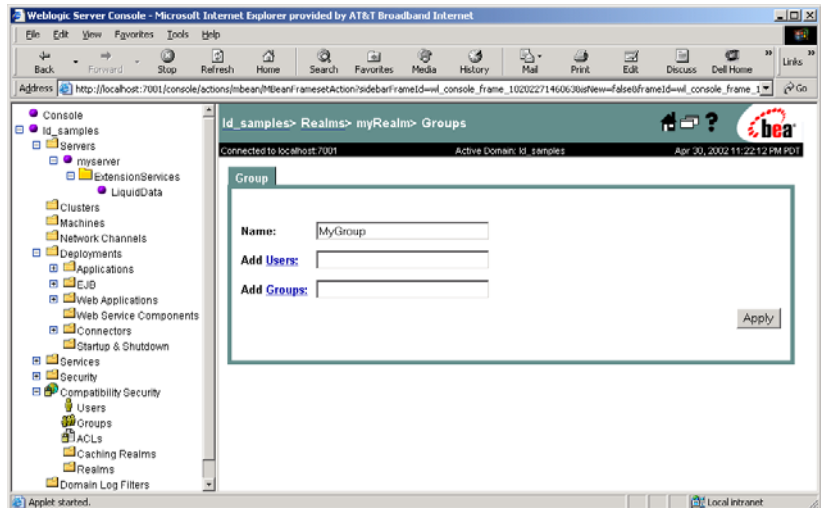
1. In the left pane of the Administration Console, click on the Compatibility Security node.
2. Click on the Groups link.

Figure 16-3 Configuring Liquid Data Groups



3. On the Groups page, click on the Create a New Group link.

Figure 16-4 Configuring a Liquid Data Group



4. On the Group page, enter the group name and add any users or other groups to this group.

5. Click Apply.

Assigning ACLs to Liquid Data Resources

This topic describes how to assign ACLs to Liquid Data resources. It contains the following sections:

- [Liquid Data Resources Requiring ACL Configuration](#)
- [Access Levels](#)
- [How ACLs Affect Access](#)

At a minimum, you must define an LD ACL, add permission attributes to it (`execute`, `read`, and `modify`), and add the `Administrators` group to the grantee list of permissions.

If Liquid Data is running in secure mode, then you must assign ACLs to *any* Liquid Data resource to which you want users to have access. In secure mode, WebLogic Server automatically denies user access to any of these resources if they do not have an associated ACL.

Liquid Data Resources Requiring ACL Configuration

Liquid Data uses ACLs to control access to the following resources:

Table 16-3 Liquid Data Resources Requiring ACL Configuration

Resource	Configuration Instructions
data source descriptions	“Configuring Secure Access to Data Source Descriptions” on page 6-3
directories and files in the server repository	“Configuring Secure Access to Items in the Server Repository” on page 15-13 Includes stored queries, data views, XML data files and schema.
custom function descriptions	“Configuring Secure Access to Custom Function Descriptions” on page 13-6

Access Levels

You can assign the following access levels in an ACL:

Table 16-4 Access Levels ACL Formats for Liquid Data Resources

Access Level	Task
execute	Run a stored query. For an ad hoc query, access is determined by whether the user has <code>execute</code> access to any data sources and custom functions associated with the ad hoc query.
modify	Create, modify, rename, or delete files or directories, or upload items to the directory. This level implies <code>read</code> access.
read	Browse or view the contents of an item, or download from the repository.

How ACLs Affect Access

ACLs ensure that only authorized users and groups can perform the following tasks:

- Access and execute a query. Liquid Data verifies user or group access, based on the ACL, 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.

You configure ACLs using the Administration Console. For more information, see “Defining ACLs in the Compatibility Realm” in [“Using Compatibility Security”](#) in *Managing WebLogic Security* in the WebLogic Server documentation.

Assigning Permissions, Users, and Groups to ACLs

In the Liquid Data configuration tabs in the Administration Console, if you click on the link to configure ACLs, the Administration Console displays the WebLogic Server ACL configuration page.

Figure 16-5 WebLogic Server ACL Configuration Page



To configure an ACL:

1. Do one of the following:

- If you are creating a new ACL, click on Add a new Permission.
- or
- If you are editing an existing ACL, click the name of the ACL that you want to edit.

The Administration Console displays the Group tab.

Figure 16-6 Group Tab for ACL Configuration

Group

ACL:

LD_stored_queries.B03-S

Permission:

MyPermission

Grant to Users:

Grant to Groups:

Apply

Table 16-5 ACL Configuration Settings

Field	Description
ACL	Unique name of this ACL.

Table 16-5 ACL Configuration Settings (Continued)

Field	Description
Permission	<p>Access level for this ACL. One of the following values:</p> <ul style="list-style-type: none"> ■ <code>read</code>—Grants <code>read</code> access to the resource, allowing authorized users to view this resource but not make any changes. ■ <code>modify</code>—Grants <code>modify</code> access to the resource, allowing authorized users to add, modify, delete, or otherwise change this resource. ■ <code>execute</code>—Grants <code>exec</code> access to the resource, allowing authorized users to execute associated queries on the Liquid Data Server. <p>For more information, see “Access Levels” on page 16-14.</p>
Users	List of users assigned to this ACL.
Groups	List of groups assigned to this ACL.

2. Specify a permission level.
3. Specify one or more users associated with this permission. Click on the Users link to display the Users tab for currently configured users.
4. Specify one or more groups associated with this permission. Click on the Groups link to display the Groups tab for currently configured groups.
5. Click Apply.

The Administration Console saves changes to this ACL.

Integrating Liquid Data Security With BEA Other Software

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

- [Web Services and Liquid Data Security](#)

- [Application Integration and Liquid Data Security](#)
- [Business Process Management and Liquid Data Security](#)
- [B2B Integration and Liquid Data Security](#)
- [WebLogic Portal 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 realms to protect access to workflows and other resources. User access is determined by the roles to which the user is assigned. The WebLogic Integration Studio is used to define users, organizations, and roles, and also to map roles to groups in WebLogic Server security realms. 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.

Business Process Management and Liquid Data Security

The Business Process Management component of WebLogic Integration has its own security mechanisms for controlling access to workflows and other resources. You use the WebLogic Integration Studio to perform such tasks as defining users, groups, roles, organizations, and permission levels. For more information, see “Step 3: Configure BPM Security” in [“Using WebLogic Integration Security”](#) in *Deploying Solutions* in the WebLogic Integration documentation.

B2B Integration and Liquid Data Security

For information about B2B Integration security, see [“B2B Security”](#) in the WebLogic Integration documentation.

WebLogic Portal and Liquid Data Security

If you want to use the WebLogic Portal security mechanisms as the entry point for user security, you can either:

- Run Liquid Data in non-secure mode, as described in [“Enabling Liquid Data Secure Mode”](#) on page 16-8.
- or
- Create a user in WebLogic Portal and, in Liquid Data, grant that user complete permissions (`read`, `modify`, and `execute`) to all Liquid Data resources, according to the instructions in [“Assigning ACLs to Liquid Data Resources”](#) on page 16-13.

For more information about WebLogic Portal security, see [“Adding Security to a Portal”](#) in the WebLogic Portal *Developer Guide* and [“Administering Users and Groups”](#) in the WebLogic Portal *Administration Guide*.

WebLogic Workshop and Liquid Data Security

For information about WebLogic Workshop security, see “[Workshop Security Overview](#)” in the WebLogic Workshop documentation.

17 Monitoring the Server

This topic 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

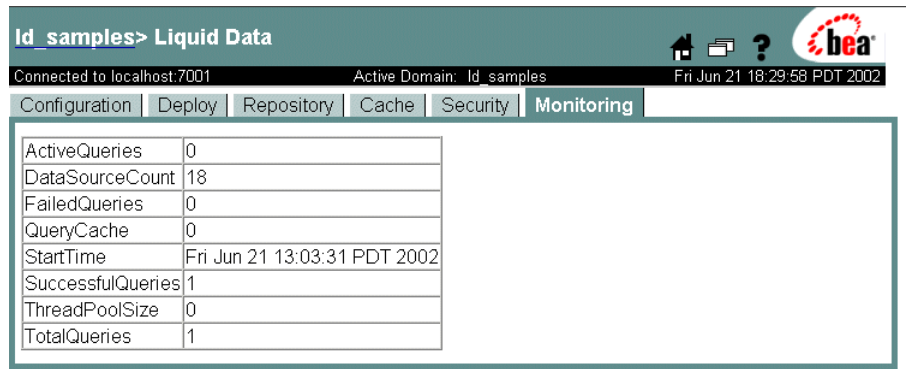
The Liquid Data node in the Administration Console includes a Monitoring tab for monitoring the current status of the Liquid Data Server.

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 17-1 Monitoring Tab on the Liquid Data Node



Id_samples > Liquid Data	
Connected to localhost:7001 Active Domain: Id_samples Fri Jun 21 18:29:58 PDT 2002	
Configuration Deploy Repository Cache Security Monitoring	
ActiveQueries	0
DataSourceCount	18
FailedQueries	0
QueryCache	0
StartTime	Fri Jun 21 13:03:31 PDT 2002
SuccessfulQueries	1
ThreadPoolSize	0
TotalQueries	1

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

Table 17-1 Monitoring Statistics for the Liquid Data Server

Field	Description
Active Queries	Number of active queries.
Data Source Count	Number of configured data sources.
Failed Queries	Number of failed queries.
Query Cache	Number of stored queries for which cached query plans exist.
Start Time	Date and time when the Liquid Data Server was started.
Successful Queries	Number of successful queries.
Thread Pool Size	Size of the thread pool.
Total Queries	Total number of queries 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 [“Using Log Messages to Manage WebLogic Servers”](#) in the *BEA WebLogic Server Administration Guide*.

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 *Creating and Configuring WebLogic Server Domains*.

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 *BEA WebLogic Server Performance and Tuning*.

18 Configuring the Query Results Cache

This topic 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 Results Caching for Stored Queries](#)

Notes: 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 18-4](#). 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 18-5](#).

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 18-2](#).

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 18-1 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 named `ldCacheDS` according to the instructions for [“Creating a JDBC Data Source” on page 7-5](#).

Note: You must use the exact data source name, `ldCacheDS`, for caching to work.

Once created, you can enable the result cache, as described in the following section, [“Enabling the Results Cache” on page 18-4](#)

Enabling the Results Cache

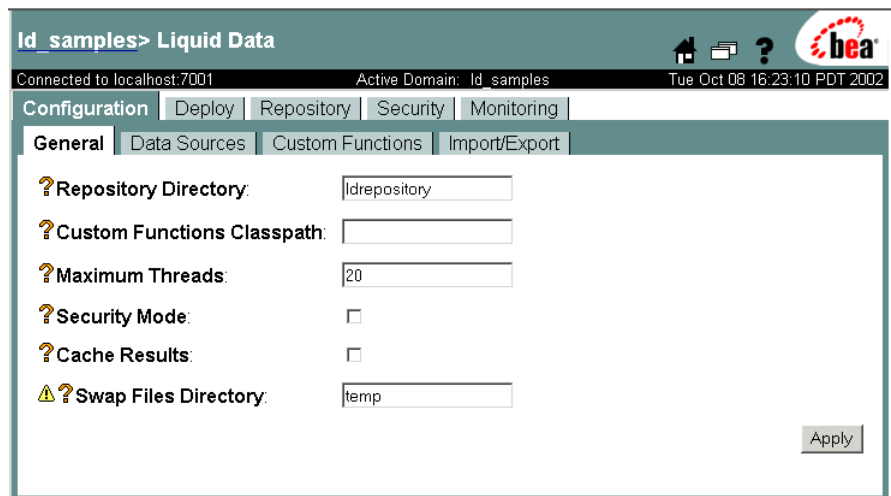
By default, results caching is disabled in Liquid Data. To use results caching, you must explicitly enable it in the General tab on the Liquid Data node in the Administration Console. Before you enable caching, make sure that you have set up the Liquid Data cache database as described in [“Setting up the Results Cache Database” on page 18-2](#).

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 18-1 Enabling Results Caching in the General Tab on the Liquid Data Node



4. On the General tab, select (check) Cache Results.
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 18-5](#).

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 topic describes how to configure the cache policy for stored queries. It includes the following sections:

- [Creating the Cache Policy](#)
- [Editing the Cache Policy](#)
- [Removing the Cache Policy](#)

Creating the Cache Policy

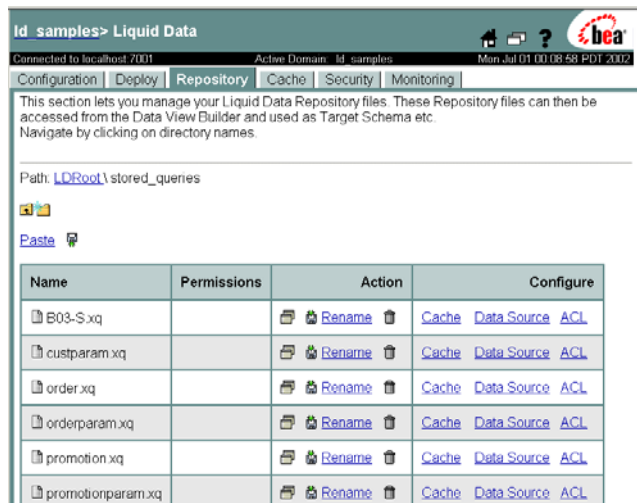
To configure the results cache for a stored query initially:

1. Navigate to the Repository tab on the Liquid Data node, as described in [“Navigating to the Repository Tab” on page 15-5](#).
2. On the Repository tab, click the `stored_queries` folder.

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

18 Configuring the Query Results Cache

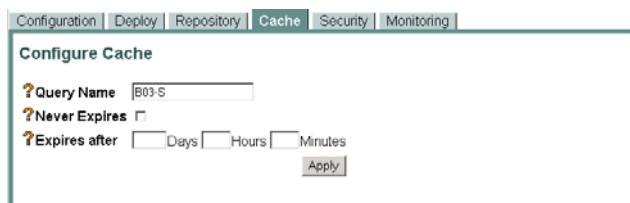
Figure 18-2 List of Stored Queries in the Repository



3. Select the stored query for which you want to configure caching.
4. Click Cache.

The Administration Console displays the Configure Cache tab.

Figure 18-3 Configure Cache Tab



5. Enter the information described in the following table.

Table 18-2 Configure Cache Settings

Permission	Task
Query Name	Name of the selected stored query.

Table 18-2 Configure Cache Settings (Continued)

Permission	Task
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	<p>Expiration time, which is calculated from when the query cache is first created.</p> <ul style="list-style-type: none">■ Days—Number of days, up to 999 days.■ Hours—Number of hours, up to 99 hours.■ Minutes—Number of minutes, up to 99 minutes. <p>If Days, Hours, and Minutes are all set to zero (0), then the results cache never expires.</p>

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.

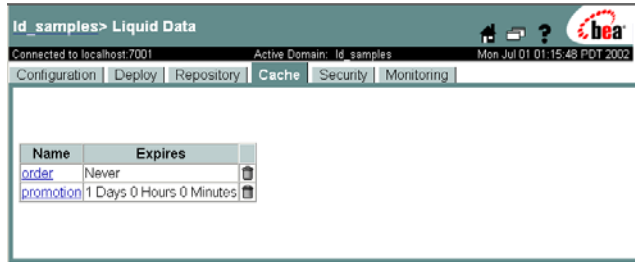
Editing the Cache Policy

To edit the cache policy for a stored query:

1. In the left pane, click Liquid Data.
2. Click the Cache tab.

The Administration Console displays the Cache tab.

18 Configuring the Query Results Cache



3. Click the name of the stored query whose cache you want to configure.
4. Change the information described in [Table 18-2, “Configure Cache Settings,” on page 18-6](#), as needed.
5. Click Apply.

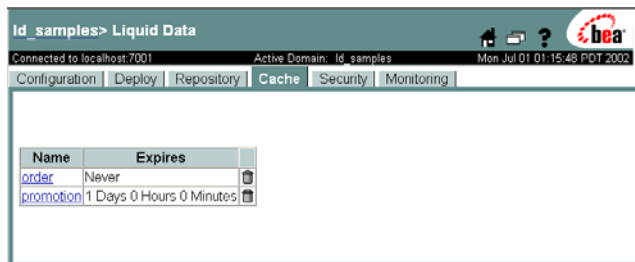
The Administration Console updates the selected cache policy and flushes any cached results for this stored query.

Removing 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.

The Administration Console displays the Cache tab.



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 15-13](#).

Flushing the Cache

Liquid Data flushes the cached query result for a given stored query whenever:

- 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 invocation* whenever:

- the query results have expired per the cache policy
- the cache policy for a stored query is updated or deleted

19 Generating and Publishing Web Services

This topic describes how to publish BEA Liquid Data for WebLogic™ stored queries as Web services. It contains the following sections:

- [Viewing a Demo](#)
- [About Web Services](#)
- [Creating a New Web Service from a Stored Query](#)
- [Modifying a Web Service](#)
- [Deleting a Web Service](#)
- [Testing a Generated Web Service](#)
- [Managing the Deployment of a Generated Web Service](#)
- [Invoking Published Web Services](#)

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 Repository tab.
3. Click the `stored_queries` directory.
4. If the stored query resides in a subdirectory of the `stored_queries` directory, navigate to the subdirectory.
5. Next to the stored query, click Generate Web Service.

Note: If the Generate Web Service link is not available, then a Web service has already been generated for this stored query.

The Administration Console prompts you to select a target schema for the web service.

Configure

Generate web service for order.xq

Select the target schema.

?Target Schema * [Browse Repository](#)

If the target schema does not exist in the repository, upload the schema file to the schema directory of the repository.

Generate

Table 19-1 Generating a Web Service

Field	Description
Target Schema	Target schema used by this query. The target schema you specify here must be available in the Liquid Data Repository. It should also be the same target schema you used to build and test the query, or at least match the structure of the target schema.

6. Enter the name of the target schema in the `schemas` directory, or click Browse Repository to select it by browsing.
7. Click Generate.

The Administration Console generates the Web service, storing the generated Web service as an EAR (enterprise archive) file in the `web_services_gen` directory, using the name of the stored query as the file name prefix. For example, the generated web service for a stored query named `order.xq` would be `order.ear`. The WSDL of the generated Web service is dynamically generated at runtime by the EAR file after it is deployed successfully.

The Administration Console automatically deploys the 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, see [“Managing the Deployment of a Generated Web Service” on page 19-7](#).

The Administration Console shows a confirmation message if it completes the process successfully, displaying the URL of the generated web service, as shown in the following example:

Configure

Generate web service for order.xq

The web service has been generated and deployed successfully.

The URI to access the web service is **`http://localhost:7001/liquiddata/order/webservice?WSDL`**

Note: Users should save the URL for future use.

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 Web Service” on page 19-5](#).
2. Regenerate the Web service according to the instructions in [“Creating a New Web Service from a Stored Query” on page 19-2](#).

Deleting a Web Service

You can delete a 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 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.
6. In the left pane of the Administration Console, click the Web Service Components node.
7. Scroll to find the Web service that you want to delete, and then click the delete icon next to it.

Testing a Generated Web Service

You can use BEA WebLogic Workshop™ to test a Web service that you have generated with the Administration Console.

Note: Before you begin, make sure that the WebLogic Workshop Example server and the Workshop tool use a different port number from the one that the Liquid Data server uses. For example, if the Liquid Data server uses port 7001, then the WebLogic Workshop Example server should be started on a different port (such as 7010) and the Workshop tool should be configured to access that different port.

To test the Web service:

1. Start WebLogic Workshop.
2. In the Design View, click Add Control, and then select Add Service Control.
3. In the Add a Service Control dialog box, specify a variable name for the service control (such as `testws`).
4. Click Create a Service Control from a WSDL, and then enter the URL of the generated Web service.
5. Click Add Operation, select Add Method, and specify a method name (such as `test`).
6. Click the new method to edit it.
7. In the Source View, scroll to the `import` section and copy the name of the `AnonType_*` associated with the Web service.
8. In the method declaration, replace `void` with the `AnonType_*` type, as in the following example:

```
public AnonType_CustomerOrderReport test()
```

9. In the method body, add a `return` statement and specify the name of the Service Control you specified earlier (such as `testws`), as in the following example:

```
public AnonType_CustomerOrderReport test()  
{
```

```
        return testws  
    }  
}
```

10. Select the `testws` variable. WebLogic Workshop shows the operation name of the generated Web service in bold.

11. Click the operation so that the source reads:

```
public AnonType_CustomerOrderReport test()  
{  
    return testws.order();  
}
```

12. Click the Start button to get to the Test Form.
13. In the Test Form, click the method name (such as `test`).
14. WebLogic Workshop runs the test, displaying the request (External Service Request) and the query result (External Service Response) in the Test Form.

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”](#) in the WebLogic Server *Administration Guide*.

For detailed information about WebLogic Web services, see [Programming WebLogic Web Services](#) in the WebLogic Server documentation.

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 *Invoking Queries Programmatically*.

Index

A

access control lists. See ACLs.

ACLs

- access levels 16-14

- assigning permissions, users, and groups 16-14

- configuring 16-13

- custom function descriptions 13-6

- data source descriptions 6-3

- folders and files in the server repository 15-13

- how ACLs affect access 16-14

ad hoc queries, security for 16-4

Administration Console

- home page 4-5

- Liquid Data node 4-5

- overview 4-3

- security 16-2

- starting 4-2

- using 4-2

administration tasks, overview 1-1

Application Integration security, integrating with Liquid Data 16-17

Application View console, defining application views 10-3

application views

- data source descriptions

 - creating 10-4

 - modifying 10-7

 - removing 10-8

- defining in the Application View

- Console 10-3

- maximum threads, configuring 5-3
- security for 16-4

- summary of configured data sources 10-6

B

B2B Integration security, integrating with Liquid Data 16-18

Business Process Management security, integrating with Liquid Data 16-18

C

cache, flushing 18-9

cacheEjb.jar file 12-2

caching realms 16-7

compatibility security realms 16-7

custom functions

- administration tasks 13-3

- classpath, configuring 5-3

- components of 13-2

- custom function descriptions

 - creating 13-4

 - defined 13-2

 - modifying 13-7

 - removing 13-8

 - security, configuring 13-6

- custom functions library definition (CFLD) file 13-5

- defined 13-2

- summary of configured custom function groups 13-5
- use cases for 13-2
- custom security realms 16-8
- custom_functions folder 13-4, 15-3
- custom_lib folder 13-4, 15-4

D

- data access security 16-3
- data source descriptions
 - application view data sources 10-4
 - data view data sources 11-2
 - defined 1-3
 - distributing to other servers 6-6
 - filter 6-3
 - RDBMS data sources 7-7
 - removing 6-5
 - secure access, configuring 6-3
 - Web service data sources 9-1
 - XML data sources 8-1
- data sources
 - security 16-4, 16-13
 - supported data source types 1-3
 - viewing all configured data sources 6-2
- data views
 - creating from stored queries 15-16
 - data source descriptions
 - creating 11-2
 - modifying 11-4
 - removing 11-5
 - summary of configured data sources 11-4
- data_views folder 15-4
- deploying
 - components to deploy 12-2
 - Deploy tab, navigating to 12-2
- domains
 - adding Liquid Data to 2-3
 - creating 2-3
 - defined 2-2

- Liquid Data and 2-2
- monitoring 17-3
- WebLogic Integration domains, adding Liquid Data to 10-2
- WebLogic Platform domains, adding Liquid Data to 10-2
- downloading files from the server repository 15-7
- dtads folder 15-4

E

- ejb_qbc.jar file 12-2
- ejb_query.jar file 12-2
- exporting Liquid Data configurations 14-5

F

- file realms 16-7
- file swapping, configuring 5-4
- files
 - cacheEjb.jar 12-2
 - ejb_qbc.jar 12-2
 - ejb_query.jar 12-2
 - ldcacheListener.war 12-2
 - ldconsole.war 12-2
 - XMediator.war 12-2
- flushing the cache 18-9
- folders
 - custom_functions 13-4, 15-3
 - custom_lib 13-4, 15-4
 - data_views 15-4
 - dtads 15-4
 - import_export 15-4
 - schemas 15-4
 - stored_queries 15-4
 - web_services 15-4
 - web_services_gen 15-4
 - xml_files 15-4

G

groups

- configuring 16-11
- LDAdmin group 16-11

I

- Import/Export tab, navigating to 14-4
- import_export folder 15-4
- importing Liquid Data configurations 14-7

J

- JDBC connection pools
 - creating 7-2
 - driver names for data sources 7-2
 - modifying 7-12
- JDBC data sources
 - creating 7-5
 - modifying 7-12
 - results cache database 18-3

L

- LDAdmin group 16-6, 16-11
- LDAP security realms 16-7
- ldcacheListener.war file 12-2
- ldconsole.war file 12-2
- ldsystem user 16-6, 16-9
- Liquid Data configurations
 - defined 14-1
 - exporting 14-5
 - importing 14-7
- Liquid Data node in the Administration Console 4-5
- Liquid Data server
 - monitoring 17-1
 - server settings, configuring 5-1
 - starting 3-1, 10-3
 - stopping 3-3
- logging 17-3

M

- maximum threads, configuring 5-3
- monitoring
 - domains 17-3
 - Liquid Data server 17-1
 - monitoring tools 17-3
 - server log 17-3

N

- non-secure mode 16-8

Q

- queries, security for 16-3, 16-13

R

- RDBMS data sources
 - connection pools
 - configuring 7-2
 - URLs 7-2
 - data source descriptions
 - creating 7-7
 - modifying 7-12
 - removing 7-13
 - JDBC data sources, creating 7-5
 - JDBC data sources, driver names 7-2
 - security for 16-4
 - summary of configured data sources 7-11
- RDBMS security realms 16-7
- realms, compatibility security realms 16-7
- Repository tab, navigating to 15-5
- repository. *See* server repository. 15-2
- results caching
 - cache policy
 - creating 18-5
 - editing 18-7
 - removing 18-8
 - configuring 18-5

- configuring for stored queries 15-17
- database set up 18-2
- defined 18-1
- disabling 5-3
- enabling 5-3, 18-4
- flushing 18-9

S

- schemas folder 15-4

- secure mode 16-8

- security

 - ACLs. *See* ACLs.

 - Administration Console security 16-2

 - Application Integration security,
 integrating with 16-17

 - application views 16-4

 - B2B Integration security 16-18

 - Business Process Management security
 16-18
 - compatibility security realms, defining
 16-7

 - data access security 16-3

 - data sources 16-4

 - disabling 5-3

 - enabling 5-3

 - groups, configuring 16-11

 - implementing, overview of 16-5

 - initial security setup 16-6

 - integrating with other software 16-16

 - LDAdmin group 16-6

 - ldsystem user 16-6

 - mode, configuring 5-3

 - non-secure mode 16-8

 - overview 16-2

 - query security 16-3

 - relational databases 16-4

 - secure mode 16-8

 - server repository files and folders 16-5

 - users, configuring 16-9

 - Web services security 16-17

 - WebLogic Integration security 16-17

 - WebLogic Portal security 16-18

 - WebLogic Workshop security 16-19

- server repository

 - browsing 15-6

 - considerations for evolving 15-4

 - contents of 15-2

 - copying files 15-11

 - data sources, configuring 15-16

 - deleting folders and files 15-13

 - downloading files from 15-7

 - file system hierarchy 15-3

 - folders, creating 15-10

 - location of 15-3

 - organization of 15-2

 - pasting files 15-11

 - renaming folders and files 15-12

 - Repository tab, navigating to 15-5

 - results cache, configuring 15-17

 - root directory, configuring 5-2

 - security for 16-5

 - security, configuring 15-13

 - uploading files to 15-8

- server settings

 - configuring 5-1

 - modifying 5-4

- starting the Liquid Data server 3-1, 10-3

- stopping the Liquid Data server 3-3

- stored queries

 - creating data views from 15-16

 - generating Web services from 19-2

 - results cache, configuring 15-17

 - results caching, configuring 18-5

 - security for 16-3

- stored_queries folder 15-4

- sub-folders, creating in the server repository
 15-10

- swap files directory, configuring 5-4

U

- UNIX security realms 16-7
- uploading files to the server repository 15-8
- users
 - configuring security for 16-9
 - ldsystem user 16-9

W

- Web services
 - data source descriptions
 - creating 9-1
 - modifying 9-4
 - removing 9-5
 - defined 19-2
 - deleting 19-5
 - demo 19-2
 - deploying 19-7
 - generating from stored queries 19-2
 - invoking queries 19-8
 - maximum threads, configuring 5-3
 - modifying 19-5
 - security, integrating with Liquid Data 16-17
 - summary of configured data sources 9-3
 - testing 19-6
- web_services folder 15-4
- web_services_gen folder 15-4
- WebLogic Integration
 - adding Liquid Data to a domain 10-2
 - security, integrating with 16-17
- WebLogic Platform, adding Liquid Data to a domain 10-2
- WebLogic Portal security, integrating with Liquid Data 16-18
- WebLogic Server
 - log file 17-3
 - starting 3-1
 - stopping 3-3
- WebLogic Workshop
 - security, integrating with Liquid Data

16-19

- testing generated Web services 19-6
- Windows NT security realms 16-7

X

- xMediator.war file 12-2
- XML data sources
 - data source descriptions
 - creating 8-1
 - modifying 8-4
 - removing 8-5
 - summary of configured data sources 8-3
- xml_files folder 15-4

