Oracle® Product Data Quality

Oracle DataLens Server Administration Guide

Version 5.6

E20602-01

January 2011

ORACLE®

Oracle Product Data Quality Oracle DataLens Server Administration Guide, Version 5.6

E20602-01

Copyright © 2001, 2011 Oracle and/or its affiliates. All rights reserved.

Primary Author: Lorna Vallad

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface vii

Audience vii
Related Documents vii
Conventions viii

1 Overview

Operation 1-2
Web Application Server 1-2
Supported Environments 1-2

2 Oracle DataLens Server Administration Web Pages

Web Page Operation 2-1
Home Page 2-2
Menus and Information Displayed 2-2
Server Configuration Values 2-3
Documentation and Log Files 2-3
Viewing the On-Line Documentation 2-3
View the Oracle DataLens Log 2-3
View User Log 2-4
Operations 2-4
Debugging Options 2-4
Configuration Options 2-5
Platform Topology 2-5
Idling the Batch Service: 2-7
Starting the Batch Service: 2-8
Configuration Information 2-8
Configuration 2-10
DataLens String Constants Administration 2-10
DataLens Numeric Constants Administration 2-11
DataLens Boolean Constants Administration 2-13
Server Groups 2-13
Adding a New Server Group 2-14
Creating or Deleting Oracle DataLens Servers 2-15
Adding a Server 2-15
Delete a Server 2-15

Role Administration 2-16
User Administration 2-17
Database Connections 2-18
Web Services 2-19
FTP Connections 2-22
User Access 2-23
Data Lenses and DSAs 2-23
Dashboard 2-24
Server 2-25
Data Lenses 2-25
User Locks 2-26
Removing User locks 2-26
Adding User locks 2-26
Data Lens Repository History 2-27
Deploying a Data Lens 2-28
Data Service Applications 2-28
Data Service Application Details 2-29
Language Glossaries 2-30
Reports 2-30
Administrative Reports 2-31
Data Lens Reports 2-31
DSA Reports 2-31
Job Status 2-31
Data Service Application Jobs 2-32
Active Jobs, Pending and Completed Jobs 2-32
Job Details 2-34
Job History 2-34
Run a Job 2-34
Database Jobs 2-36
Schedule a Job 2-36
Running the Job Scheduler on Server Groups 2-38
Edit Scheduled Jobs 2-38
Troubleshooting 2-39
Non-English Characters Incorrectly Displayed 2-39
Windows Platforms 2-39
Memory Allocation Problems on Clients 2-40
Problems Starting a Non-Admin Oracle DataLens Server 2-41

A Configuration Files

Server.cfg A-1
SvrPaths.xml A-2
Web.xml A-3

B Logging

Oracle DataLens Server Log File B-1
Java Server Log Files B-1

C Server Configuration

Copying the Repository C-1
Copy the Directories C-1
Configure the New Server C-1
Job Continuation C-2
Expected Results C-2

D Mounting a Remote Repository

Windows-to-Windows Mounting D-1
Linux-to-Windows Mounting D-2
Windows to Linux Mounting D-3
Linux to Linux Mounting D-3
Configuring the SvrPaths.xml File D-4

E User Authentication with LDAP

Oracle DataLens Server LDAP Configuration Files E-1
LDAP User Restrictions E-2
Setting Up a Sample LDAP Directory E-3

F Tuning the Server(s)

Checking the Results F-1
Oracle DataLens Server Options F-1
Load-Balancing the Servers F-1
Round Robin Calls F-1
Ensure Tracing is Turned Off F-2
Data Service Application Optimization F-2
Simplify the Data Service Application Process Steps F-2
Running Ultra High-Priority Jobs F-2
Run Jobs at the Correct Priority F-2
File Writing Between Steps F-2
Data Lens Optimization F-2
Caching the Data Lenses F-2
Do Not Load Data Lenses That Are Not Being Used F-3
Tuning Multiple Parameterized Domains F-3
API Integration F-4
WSDL Versus Java API Calls F-4
Optimize the Available Hardware and Operating Systems F-4
Windows Memory and Application Servers F-4
Linux Memory, Windows Memory, and Java Servers F-4
Database Query Tuning F-4

G Installing East Asian Language Support Files for Windows XP

Preface

This purpose of this document is to describe the administration of an Oracle DataLens Administration Server.

To understand all of the features presented, you must use this guide in conjunction with the Oracle Product Data Quality documents listed "Related Documents" on page -vii.

Audience

A thorough understanding of the material in this guide is required for the followingcustomer personnel:

- IT Administrators responsible for configuring and tuning Oracle Product Data Quality.
- Oracle DataLens Server Administrators that manage the configuration of the system, such as setting up database connections, job control, and setting up users and assigning roles.

Related Documents

For more information, see the following documents in the documentation set:

- The *Oracle Product Data Quality Oracle DataLens Server Installation Guide* provides detailed Oracle Product Data Quality Oracle DataLens Server installation instructions.
- The *Oracle Product Data Quality COM Interface Guide* provides information about installing and using the Oracle DataLens Server COM APIs.
- The *Oracle Product Data Quality Java Interface Guide* provides information about installing and using the Oracle DataLens Server Java APIs.
- The *Oracle Product Data Quality Application Studio Reference Guide* provides information about creating and maintaining Data Service Applications (DSAs).
- The *Oracle Product Data Quality Knowledge Studio Reference Guide* provides information about creating and maintaining data lenses.
- The *Oracle Product Data Quality Governance Studio Reference Guide* provides information about creating and maintaining Data Service Applications (DSAs).
- The *Oracle Product Data Quality Glossary* provides definitions to commonly used Oracle Product Data Quality technology terms.

- The *Oracle Product Data Quality Services for Excel Reference Guide* provides information about creating a DSA based on data contained in a Microsoft Excel spreadsheet.
- The *Oracle Product Data Quality Task Manager Reference Guide* provides information about managing tasks created with the Task Manager or Governance Studio applications.

See the latest version of this and all documents listed at the Oracle Product Data Quality Documentation Web site:

http://download.oracle.com/docs/cd/E20593_01/index.htm

Conventions

The following text conventions are used in this document:

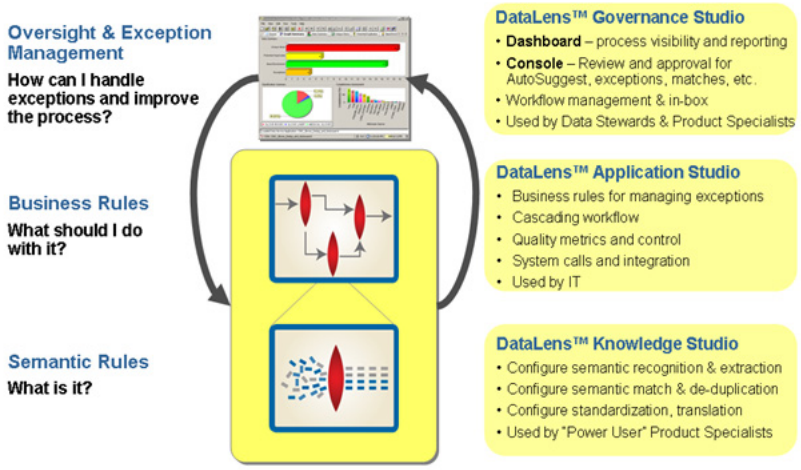
Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.
`monospace`	Boldface, monospace type indicates commands or text that you enter.

1

Overview

Oracle DataLens Server is built on industry-leading DataLens™ Technology to standardize, match, enrich, and correct product data from different sources and systems. The core DataLens Technology uses patented semantic technology designed from the ground up to tackle the extreme variability typical of product data.

Oracle Product Data Quality uses three core DataLens Technology modules: Governance Studio, Knowledge Studio, and Application Studio. The following figure illustrates the process flow of these modules.



Oracle Product Data Quality provides network-wide access to DataLens Servers (Data, Transaction, and Knowledge Engineering Services). The DataLens Administration Server is for archiving data lenses and Data Service Applications (DSAs) and running large amounts of data through these data lenses and DSAs.

Each Oracle DataLens Server consists of the following facilities:

- Servlet Engine
- Configuration Repository
- Oracle DataLens Server Administration Web Application Engine
- Oracle DataLens Web Application Engine
- Oracle DataLens Server Group Engine

Operation

The Oracle DataLens Server provides an HTTP service operating on a dedicated port (default port 2229), executing the Server Web Application. The Knowledge Studio and Administration Server Web pages both communicate with the Oracle Product Data Quality using HTTP SOAP requests.

Web Application Server

The Oracle Product Data Quality components run under a standard Java J2EE Servlet Engine in a web application server environment (for example, Oracle WebLogic). In this environment, Oracle Product Data Quality runs as a independent web application.

Supported Environments

The Oracle DataLens Administration Server supports both Linux and Windows servers, and WebLogic and Apache Tomcat application servers. For more information, see the interface and installation guides listed in "Related Documents" on page -vii.

2

Oracle DataLens Server Administration Web Pages

The Oracle DataLens Server includes a set of administrative web pages that allow both local or remote administration of the Oracle DataLens Server. All the administration for the Oracle DataLens Server topology is handled using the web pages accessed on the Administration Server.

The Oracle DataLens production servers can be accessed using a web page. The production server web page provides information on the status of the license, data directories, software versions, and log files. Each production server provides separate informational web page as previously described.

Web Page Operation

The Oracle Product Data Quality supports the Microsoft Internet Explorer browser version 7 or greater.

On the server, browse to:

`http://localhost:2229/datalens`

Or remotely browse to

`http://server:2229/datalens`

Where `server` is the name of your Oracle DataLens Server.

The Oracle Product Data Quality login screen is displayed

Enter a login name and password and click **Login**.

Home Page

Upon a successful login to the server, you are allowed to enter the Oracle DataLens Server home page. This page contains a welcome message, shows the type of server configuration and a menu to the left showing links to the Oracle DataLens Server Administrator Web pages.

This home page displays the name of the user currently logged into the Data Server Administrator, as well as, the type of server that was installed.

If logging onto an Oracle DataLens Transform Server, then this home page is the only page that is available.

If logging in as a user without Administrator privileges, then the Web pages are all read-only. There is no way to change the server configuration without logging in as a user with Administrator privileges.

Note: If you are logging in as the administrator, you may need to refresh the screen in order to see all of the administrative selections.

Menus and Information Displayed

These menus are available only to the Administration Server.

The versions of the software used by the server are displayed.

Any users connected to the Oracle DataLens Server from the Knowledge Studio or the Application Studio will be listed as part of the User Connections.

This home page displays information about the configuration and operation of the Oracle DataLens Server. The configuration and installation directories are displayed here. These directories are changed during the Oracle DataLens Server installation. If they need to be changed after the server has been installed, then the web.xml file will need to be manually manipulated. For more information, see Appendix A, "Configuration Files.".

Home Directory

The Oracle DataLens Server home directory is defined with the environmental variable, `OPDQ_HOME`, which is created and set during the installation process. For more information, see *Oracle Product Data Quality Oracle DataLens Server Installation Guide*.

This directory contains the configuration files, system information, and the Oracle DataLens log files used by the server. The installation WAR file, `datalens.war`, is

located in the home directory if a WebLogic Application Server was installed. When a Tomcat Application Server is installed, the WAR file is located as follows:

- **On Linux -**

  `OPDQ_HOME/tomcat/webapps`

- **On Windows -**

  `OPDQ_HOME\tomcat\webapps`

Connected Users

This will show what users are currently connected to the server from the Oracle Product Data Quality Client applications such as the Knowledge Studio and the Application Studio.

This is useful if an administrator needs to bring the server down, or make backups or make substantial changes to the server configuration and wants to know what users may be currently working and using the server. The administrator should also check the server jobs to check what users are running real-time or batch jobs against the server as well.

Server Configuration Values

Data Service Application Chunk Size

This size is used to:

- determine how many records are sent to a single step in a DSA one time
- determine how many records are delegated to a separate Oracle DataLens Server as a single group from an executing DSA
- determine how many records are read from a database at one time from a DSA

For information about setting this value, see "Configuration Options" on page 2-5.

Data Service Application Max Memory Lines

This size defines how many lines of data to keep in memory between the steps of a DSA. If the number of lines of data is greater than this parameter, then the data is written to disk between each step of the DSA, thus hinder performance.

For information about setting this value, see "Server.cfg" on page A-1.

Documentation and Log Files

The following sections explain how to find the OPDQ documentation and log files.

Viewing the On-Line Documentation

Click the **View On-Line Documentation** button on home page to display a link to the Oracle Product Data Quality on-line documentation.

View the Oracle DataLens Log

Click the **View DataLens Log** button on home page to display the contents of the Oracle DataLens log file. This file is automatically backed up when the file size reaches one megabyte in size. The older files are saved with a version number in the home directory. The file is listed in historical order, with the latest information at the bottom of the file.

This file lists any errors that have happened, as well as a complete list of all the administrative commands that have been issued on the server.

The following is an example of the time-stamped history in the log file:

```
DLS Log file
INFO 22 Dec 2009 11:26:49 [main] - Initialized Logging facility to check log.cfg
every 20 seconds
INFO 22 Dec 2009 11:26:50 [main] - The DataLens Administrator is starting.
INFO 22 Dec 2009 11:26:50 [main] - Version 5.6.02, build 9,488, on Dec 22, 2009 at
9:43 AM
INFO 22 Dec 2009 11:26:50 [main] - Data Service Application chunk size=5000, max
memory lines=5000
INFO 22 Dec 2009 11:26:54 [main] - Writing profile to database: LVALLAD-T60
INFO 22 Dec 2009 11:42:46 [main] - Initialized Logging facility to check log.cfg
every 20 seconds
```

View User Log

Click the **View User Log** button on home page to display the contents of the Oracle DataLens individual user log file. This log is a record of changes made by the individual users of the system.

The following is an example of the time-stamped history in the log file:

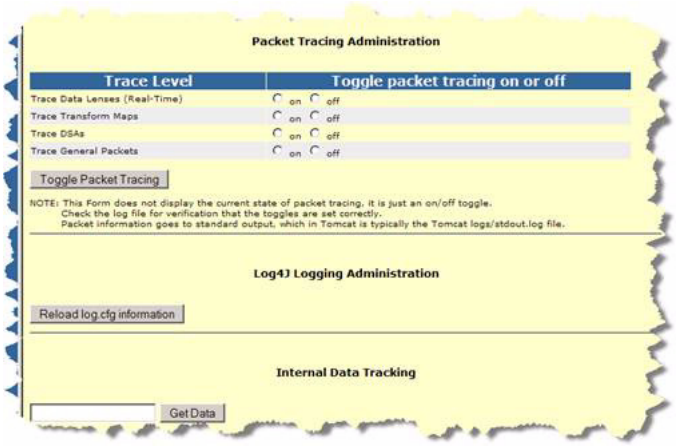
```
22 Dec 2009 12:04:27 - DB Connection MySQL added by: admin
22 Dec 2009 12:05:47 - User Added: lvallad (Lorna Vallad) by: admin
22 Dec 2009 12:06:23 - User Added: dleeper (David Leeper) by: admin
14 Jan 2010 09:04:12 - Data Service Application Retail_Capabilities_Showcase,
revision 1, Checked In by lvallad
14 Jan 2010 09:04:13 - Data Service Application Retail_Capabilities_Showcase,
revision 1, deployed to Development by lvallad
14 Jan 2010 09:32:24 - DB Connection MySQLData added by: admin
14 Jan 2010 09:37:16 - Data Lens Demo_Retail_Manufacturers, revision 1, Checked In
by lvallad
14 Jan 2010 09:37:16 - Data Lens Demo_Retail_Manufacturers, revision 1, deployed
to Development by lvallad
14 Jan 2010 09:38:40 - Data Service Application Retail_Capabilities_Showcase,
revision 2, Checked In by lvallad
14 Jan 2010 09:38:41 - Data Service Application Retail_Capabilities_Showcase,
revision 2, deployed to Development by lvallad
14 Jan 2010 09:38:44 - Data Lens Demo_Retail_Apparel, revision 1, Checked In by
lvallad
```

Operations

The **Operations** link is available only from the home page. The individual Oracle DataLens Servers only have the home page available when logging on using the Oracle DataLens Server Administration Web pages. This makes the operations available to any of the servers in the pod.

Debugging Options

This toggles informational output in SOAP format from external HTTP request to the server. This is useful for verifying or debugging information being sent from a client application to the Oracle DataLens Server.



Note: For additional information before using the Debugging Options, contact Oracle Consulting Services.

Configuration Options

These options configure the Oracle Product Data Quality server related elements including:

- Platform topology administration
- Configuration of server constants
- Server and server group administration
- Role administration
- User administration
- Database connection administration
- Web services administration
- FTP connection administration

Platform Topology

This is where the server configuration options are changed for the Administration Server, all other Oracle DataLens Servers, and the Server Groups. This also provides a view of the Oracle DataLens Platform Topology.

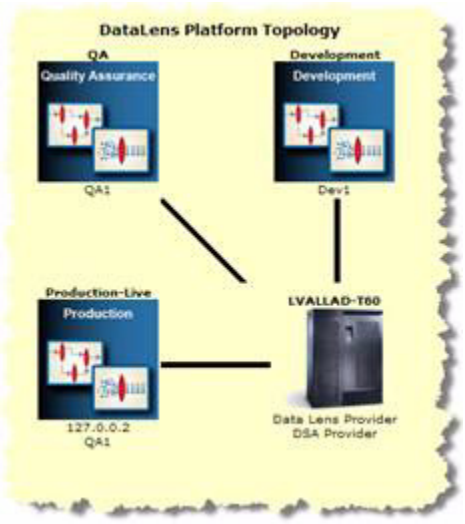
The administration of Oracle DataLens Servers and the Administration Server are controlled from the Administration Server.

In the following topology, there are the following servers:

- 1 Admin server
- 1 Development servers
- 1 QA servers
- 2 Production servers

It is more important for the QA server group to match the Production server group so realistic stress testing, load balancing, and performance monitoring can occur.

The initial page will look like the following, depending on the number of Oracle DataLens Server Groups that are part of the Oracle DataLens topology.

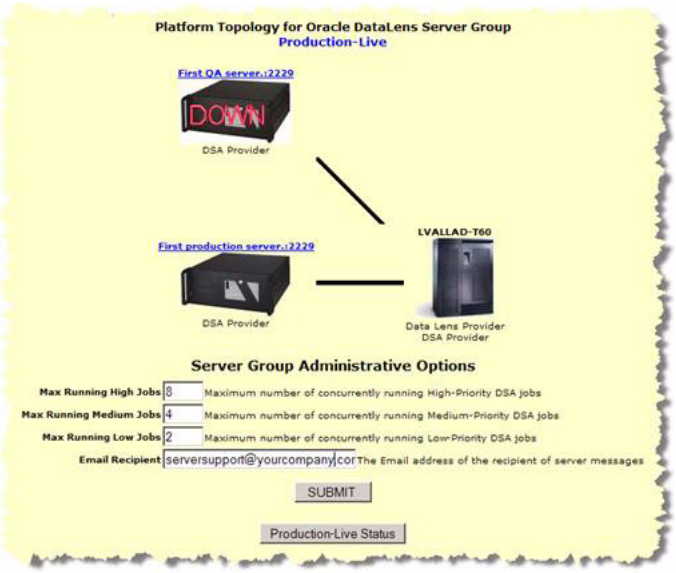


This initial screen shows the Oracle DataLens Administrator Server (LVALLAD-T60), with the Oracle DataLens Server Groups radiating out. This screen is generated dynamically based on the individual topology of your particular server configuration. The names of the Oracle DataLens Servers are listed in the Server Group in which they belong.

Note: In the preceding server topology, the server named LVALLAD-T60 is the administrator server. In some cases, the administrator server is also a development server thus is called a combo server. In this case, the topology will list the capabilities of the combo server below the icon as shown in the following.

The use of a combo server as a development server is not the recommended configuration. In this example, this combo server can be used for development testing of data lenses and DSAs.

Clicking on one of the server group icons advances you to the **Platform Topology** page for that particular Server Group as shown in the following when clicking on the Development Server Group.



The server in the center of the diagram is always the Oracle DataLens Administration Server (LVALLAD-T60 in this example). The other servers in the topology are Oracle DataLens Servers. Clicking on any server will take you to the administration page for that server.

The controls on the selected Server group page are used as follows:

Max Running High Jobs
This is the maximum number of high-priority jobs that will be run simultaneously on this server group.

Max Running Medium Jobs
This is the maximum number of medium-priority jobs that will be run simultaneously on this server group.

Max Running Low Jobs
This is the maximum number of low-priority jobs that will be run simultaneously on this server group.

Email Recipient
This is the email address of the administrator that is notified when warnings or errors occur at a server group level from the Oracle DataLens Server.

Server Group Status
Clicking this button allows you to idle or start the server group depending on its state.

You may occasionally need to stop and re-start the Batch Service. This is the best way to stop processing on the servers. The server group will wait until all current requests have finished processing before idling the servers in the group. Any requests that come in after the server group has been idled will be denied access.

Idling the Batch Service:

Click the **Idle this Server Group** button to set the server group to an idle state.

Upon successful completion of the stop request, the button changes to **Restart this Server Group**. If there are jobs running, and you need to stop the server, the currently running jobs will finish and the server will be idled.

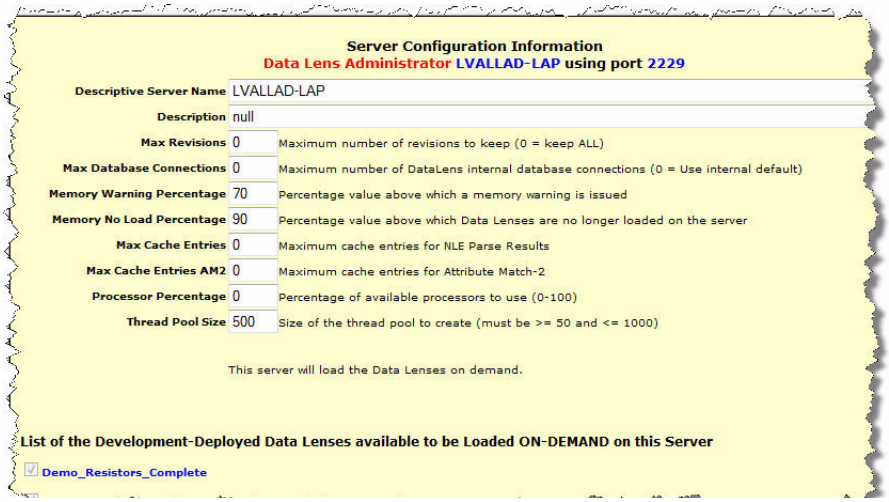
Starting the Batch Service:

Click the **Restart this Server Group** button.

Upon successful completion of the start request, the button changes to **Idle this Server Group**.

Configuration Information

Select an Oracle DataLens Server by clicking on the image of the Oracle DataLens Server from the **Platform Topology** Web page to get to the configuration information for the Oracle DataLens Server. If you select an Oracle DataLens Server image for a particular server, you are advanced to the configuration page for that particular Oracle DataLens Server. For this example, the configuration information for the LVALLAD-LAP Oracle DataLens Server is being changed.



You can modify the values on this page as follows:

Max Revisions

The number of versions of a data lens or DSA in the repository. The copies have revision history information with dates, comments and the user that changed them. Any revision that is stored in the repository can be checked out using the Knowledge Studio or Application Studio in case you want to revert to a previous version. Any version can be deployed as described in "Server" on page 2-25. This is only applicable to Oracle DataLens Administration Server.

Max Database Connections

This is the maximum number of connections that the Oracle DataLens Server will use to connect to the internal database.

Memory Warning Percentage

This is a value from 0 to 100. If the memory currently being used by Web Server and the Oracle DataLens Server exceeds this value, then a warning is issued to the main Admin and Server Group email recipients. For more information, see "Configuration" on page 2-10 and "Server Groups" on page 2-13.E

Memory No Load Percentage

This is a value from 0 to 100. If the memory currently is being used by the Application Server and the Oracle DataLens Server exceeds this value, then no more data lenses will be loaded into memory on this particular server. Email alerts are sent to the main

Admin and Server Group email recipients. For more information, see "Configuration" on page 2-10 and "Server Groups" on page 2-13.

Max Cache Entries

This is an integer value. It sets the maximum number of parsed lines of data that a data lens will cache. This cache is a Least Recently Used Cache, so when the maximum is reached, the least recently used entry is dropped and the newly parsed data is added as an entry. The default value, zero, indicates that caching will occur to the maximum amount of memory percentage set by the **Maximum Memory Percent for adding to Global Parse Cache** Numeric Constant. For more information about this constant, see "Configuration" on page 2-10.

Max Cache Entries AM2

This is an integer value. It sets the maximum number of entries in the Attribute Match 2 Cache. This cache is a Least Recently Used Cache, so when the maximum is reached, the least recently used entry is dropped and the new data retrieved from a database query is added as an entry. The default value, zero, indicates that *no* caching will occur; the higher the value the more memory is allotted for caching.

> **Note:** The behavior of this value is the opposite of the **Max Cache Entries** value.

Processor Percentage

This is a value from 0 to 100. It sets how DSA processing distribution is distributed across all CPUs in the server by percentage. If this value is set to 0, processing distribution is *off*; this is the default. If the value is > 0, processing is distributed to CPUs as calculated by the defined percentage value rounding to the nearest whole number though never less than 1. For example, if the server contains eight CPUs and you set the value to 87, processing is distributed to seven of the eight CPUs. To distribute processing to all eight CPUs, you would set this value to 100.

Thread Pool Size

This is an integer value from 50 to 1000. It sets the thread pool boundary on the number of threads in the pool that can be running on the Oracle DataLens Server. Because the server can run numerous jobs simultaneously and each job can start multiple threads to distribute the work across processors, a thread pool is created so that the server does not run out of threads. The default value is 500. Any value less than 50 is set to 50; any value greater than 1000 is set to 1000.

The following algorithm is intended to aid you in calculating the thread pool size value:

(Number of jobs running simultaneously * (Number of Lens Transform Maps in the DSA) + (Number of DB Update Transform Maps in the DSA) + (a few spare)

The following additional two values appear if the server is an Oracle DataLens Transform Server:

Server Alias

This is the alias for the selected Oracle DataLens Transform Server.

Server Group

The server group that will be used by the selected Oracle DataLens Transform Server.

Deployed Data Lenses

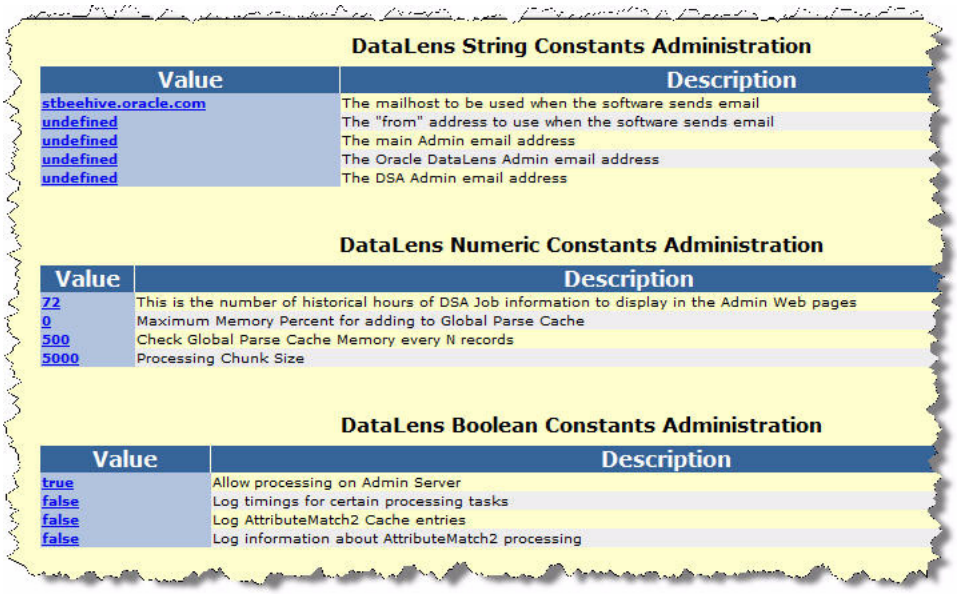
The set of check boxes for all the data lenses that are deployed to the Oracle DataLens Servers enables the system administrator further control over which data lenses are loaded on each particular Oracle DataLens Server. This is the most important consideration because the bulk of the processing usually takes place in the data lenses, and the data lenses take up the most memory on the server as well.

This is not applicable to the Oracle DataLens Administration Server because data lenses are loaded on demand. In other words, data lenses are loaded when needed for job processing.

Note: If all of the servers in a server group are set to deploy all of the data lenses, then the Oracle DataLens Administration Server will manage the load-balancing of datalens job processing across all the servers in the server group. Otherwise, you can use these settings for manual load-balancing.

Configuration

Click the **Configuration** link to modify the various Admin server options as follows:



DataLens String Constants Administration

Value	Description
stbeehive.oracle.com	The mailhost to be used when the software sends email
undefined	The "from" address to use when the software sends email
undefined	The main Admin email address
undefined	The Oracle DataLens Admin email address
undefined	The DSA Admin email address

DataLens Numeric Constants Administration

Value	Description
72	This is the number of historical hours of DSA Job information to display in the Admin Web pages
0	Maximum Memory Percent for adding to Global Parse Cache
500	Check Global Parse Cache Memory every N records
5000	Processing Chunk Size

DataLens Boolean Constants Administration

Value	Description
true	Allow processing on Admin Server
false	Log timings for certain processing tasks
false	Log AttributeMatch2 Cache entries
false	Log information about AttributeMatch2 processing

Each of these settings is a configuration parameter that is set globally. Select any of the values to change the current settings as described in the following sections.

DataLens String Constants Administration

Value	Description	What It Does
`stbeehive.oracle.com`	`The mailhost to be used when the software sends email`	The SMTP mail server to be used when OPDQ sends email to defined recipients. This value is validated so ensure that is accurate to avoid an error.

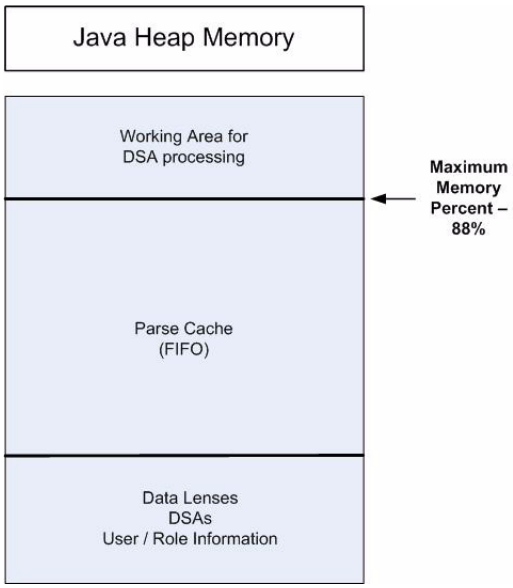
Value	Description	What It Does
undefined	The "from" address to use when the software sends email	The sending email address that will be used when OPDQ sends email to defined recipients and should be the Administrator of your Oracle DataLens Administration Server. This value is not validated so ensure that you have entered the text string correctly. For example, admin@yourcompany.com. Some email programs can misdirect this type of email to a junk or spam folder so you should ensure that this address is added to your safe email sender list.
undefined	The main Admin email address	The main Oracle DataLens Administration Server Administrator's email address and will likely be the same as the above constant value. All administrative change alerts are emailed to this recipient including user and role changes, DSA and data lens check-ins and check-outs, and server group changes. For example, admin@yourcompany.com. An email alias can be used to alert a group of administrators. This value is not validated so ensure that you have entered the text string correctly.
undefined	The Oracle DataLens Admin email address	The data lens Administrator's email address. This recipient receives alerts about actions to data lenses only. This value is not validated so ensure that you have entered the text string correctly. For example, dl_admin@yourcompany.com.
undefined	The DSA Admin email address	The DSA Administrator's email address. This recipient receives alerts about actions to DSAs only. This value is not validated so ensure that you have entered the text string correctly. For example, dsa_admin@yourcompany.com.

DataLens Numeric Constants Administration

Value	Description	What It Does
72	This is the number of historical hours of DSA Job information to display in the Admin Web pages	This is the number of historical hours of DSA Job information to display in the Administration Web pages though all history is logged for review. This value can be changed to dynamically view more or less hours of history. For example, setting to 720 will give you about 30 days of history. To view the transactional history, click the **Job Status** link.

Value	Description	What It Does
0	Maximum Memory Percent for adding to Global Parse Cache	Sets the portion of the total global Java Heap memory allocated for the Global Parse Cache. This total includes the memory already allocated at server startup for DSA, data lens, user, and role information needed for processing. The difference between the memory allocated at server startup and the threshold value is the Global Parsing Cache. The heap memory above this threshold is available for run time processing for each DSA step (all Transformation and Decision Maps).. Setting this value to zero turns it off, which turns off the **Check Global Parse Cache Memory** option. This global constant is most useful when your data contains the literally the same strings of repeating data. This global constant and the next are used in conjunction to tune your data so the values set will be unique to your environment. If memory allocation errors occur, see "Troubleshooting" on page 2-39.
500	Check Global Parse Cache Memory every N records	Sets the frequency, every *N* records, to check how close the Maximum Memory Percent for adding to Global Parse Cache constant is to its defined value. Setting this value too low can use valuable memory and processing while setting it to high can result in exceeding the allocated memory.
5000	Processing Chunk Size	Sets the number of records (lines of data) that are processed within each DSA step. For example, if 45,000 lines are sent into a DSA step for processing when this value is set to the default of 5,000 then the server will create chunks of 5000 records to process. The server processes each chuck one record at a time until all of the records are processed.

The two Global Parse Cache Numeric Constants, and the **Max Cache Entries** String Constant previously described, are controlled by the Java Heap as follows:



Following is an explanation of how these cache and memory settings operate together:

If **Max Cache Entries** = 0 and **Maximum Memory Percent** = 0, then NLE Parse Results caching is turned *off*.

If **Maximum Memory Percent** > 0, then the following are true:

Max Cache Entries = 0 means *unlimited* cache entries

Max Cache Entries > 0 means limit cache entries to this number

The default **Maximum Memory Percent** is 88, which means that by default Global Parse Caching is *on* with *unlimited* entries. In conjunction, setting **Max Cache Entries** has the following two meanings:

Maximum Memory Percent = 0 means that caching is *off*

Maximum Memory Percent > 0 means that caching is limited only by memory, not by record count. For example, running with it set to 88 means that the cache will expand until 88% of Java Heap memory is used. It could be more, depending on how often you are checking the memory used with the **Memory Warning Percentage** String Constant previously described. At that point, the Java Heap stops increasing the cache size (it adds the latest and drops the least recently used) though a lot of memory is still being consumed by the cache.

DataLens Boolean Constants Administration

Value	Description	What It Does
`true`	`Allow processing on Admin Server`	Allow job processing on the Oracle DataLens Administration Server. To ensure good performance of this server, job processing should take place on production and development servers. When this is the case, this value should be set to `false`. If set to `true`, then the Oracle DataLens Administration Server will also function as a development server.
`false`	`Log timings for certain processing tasks`	Log timings for certain processing tasks. This value should only be changed at the direction of Oracle Consulting Services.
`false`	`Log AttributeMatch2 Cache entries`	Log AttributeMatch2 Cache entries. This value should only be changed at the direction of Oracle Consulting Services.
`false`	`Log information about AttributeMatch2 processing`	Log AttributeMatch2 Cache processing information. This value should only be changed at the direction of Oracle Consulting Services.

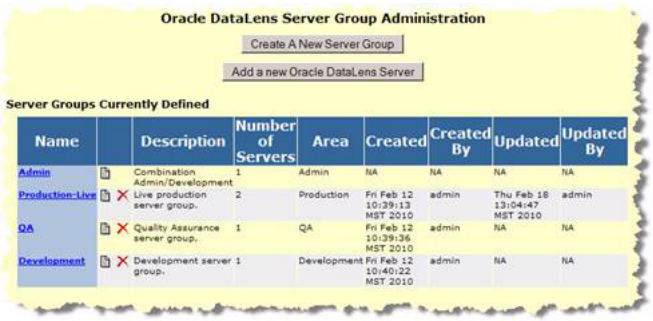
Server Groups

Server Groups allow Oracle DataLens Servers to be grouped together for use in the following scenarios.

- Development
- Production
- Quality Assurance Testing

Each Server in the group will participate in automatic server load balancing within the group. The servers can also be used for round robin checking from client applications to determine which server in the group is running and available to handle requests.

The Oracle DataLens Servers do not have a "type" until they are assigned to a server group. The type of server will depend on the type of group that the server is assigned to.



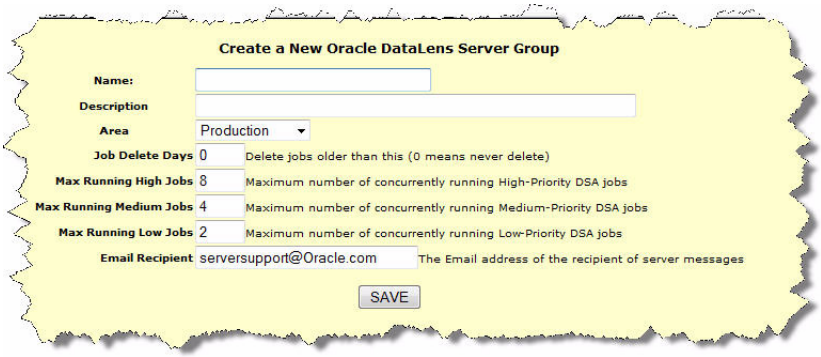
Server Groups can be created, edited, and deleted. The groups that are created can be seen graphically from the **Platform Topology** Web pages.

Note: The Oracle DataLens Administration Server resides in a hidden Admin Group that cannot be modified.

Additionally, a temporary "default" group is used if an Oracle DataLens Transform Server is created prior to creating any server groups. This default group is always a Development Group.

Adding a New Server Group

Click the **Create A New Server Group** button to get the following screen.



1. Enter the name of the new server group and give it a description.
2. Select a type. There are 3 choices in the drop-down menu
   - **Production** - This group will contain the servers used in a production environment to process data.
   - **Development** - This group will contain the servers used in a development environment to test new changes to data lenses, and DSAs.
   - **QA** - This group will contain the servers used in a Quality Assurance environment to test the work done by the developers, prior to their use in a production group.

3. Set the parameters for this server group. These parameters can be increased as new servers are added to the group.
4. Select **Save** to create the new Server Group

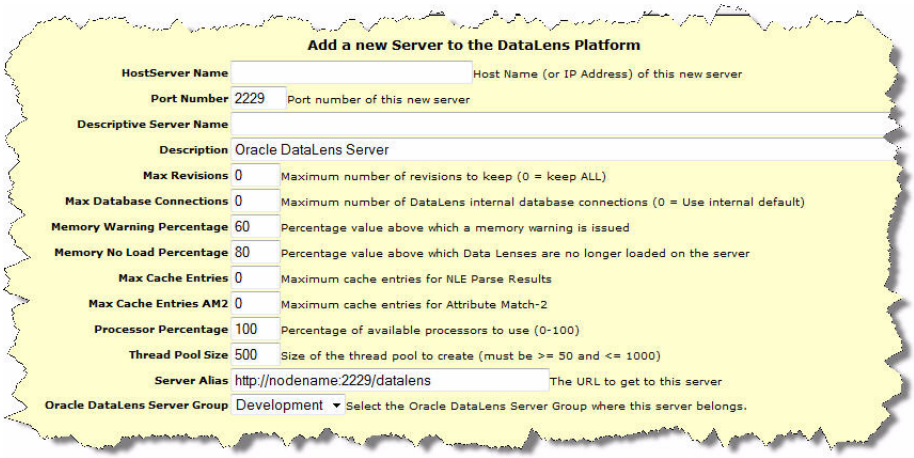
Creating or Deleting Oracle DataLens Servers

The following sections explain how to create or delete an Oracle DataLens Server.

> **Note:** You must create a Server Group before can create an Oracle DataLens Server. For more information, see "Adding a New Server Group" on page 2-14.

Adding a Server

In the server group administration page (as shown previously), click the **Add a new Oracle DataLens Server** button.



This will add a new Oracle DataLens Server to the Oracle DataLens topology. The server must be physically installed onto the hardware prior to adding the server to the topology. Once the server has been physically installed on a machine and started, then it is ready to be added as follows:

1. Enter the Network name or IP address of this new server as the server name.
2. Change the port number if not 2229.
3. Give the server a Description.
4. Select the configuration options (use the defaults if you are unsure of what to use). For more information about each of these options, see "Configuration Information" on page 2-8.
5. The server alias is a URL that will work within the network environment to take a user to the server Web page for this new server.
6. Select one of the Server Groups you have created.
7. Click **Submit** to create the new server in the Oracle DataLens Server topology.

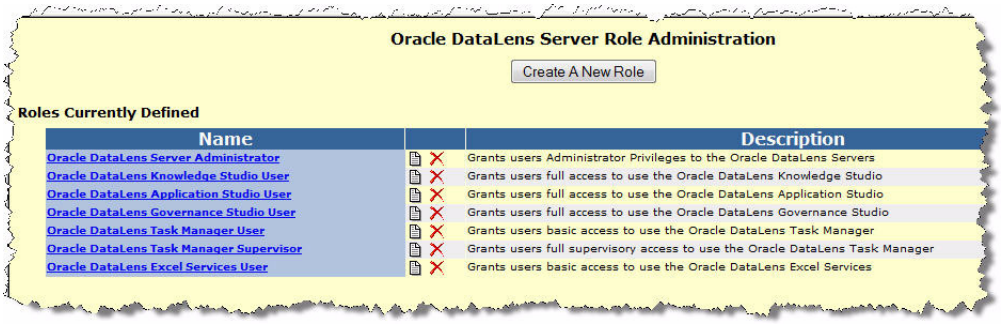
Delete a Server

From the **Configuration** page, select **Platform Topology**, and then select the server group and then select the particular server to be deleted from the topology.

Click the **Delete this datalens Server** to delete the server from the Oracle DataLens topology.

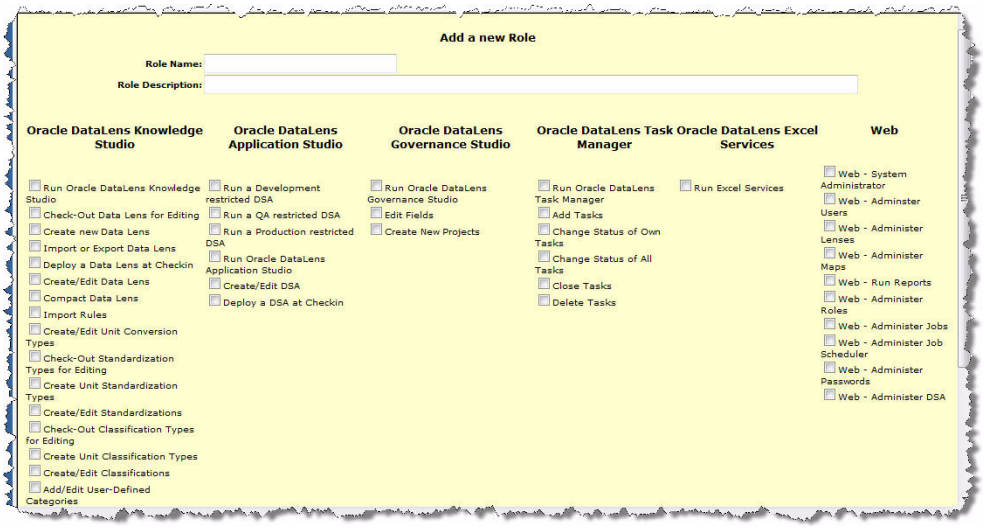
Role Administration

Selecting the **Role Administration** link takes you to the following page.



This allows you to add additional roles that can be assigned to users of the Oracle DataLens Server. These roles allow users to be conveniently grouped together for similar permissions. The roles can be created, edited, renamed, and deleted.

When creating a new role, the following page is displayed:



Select the permissions that you would like to be included in the new role and they will be part of the role when you select **Submit**.

When editing a role, the same page is also displayed. Add or delete permissions that you want included in the role and click the **Submit** button when done.

User Administration

Selecting the **User Administration** link takes you to the following page.

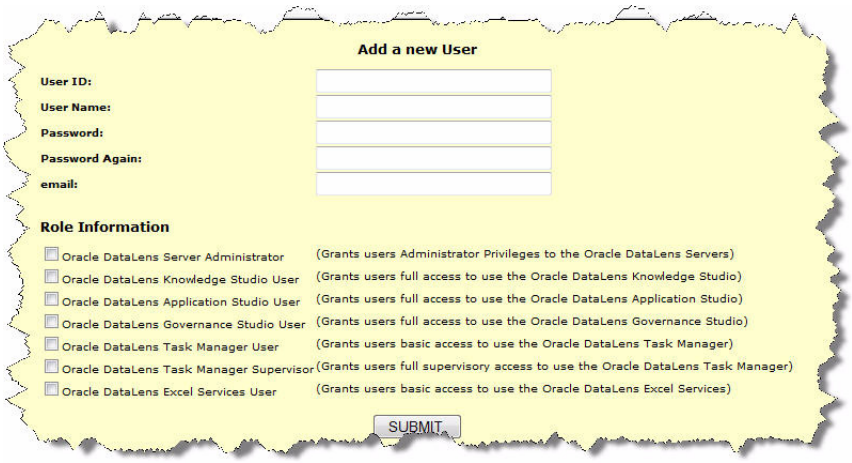
Additional users need to be added for use with the Oracle Product Data Quality client software and to enable logging into the Administration Web pages. These users need to be created before the Oracle Product Data Quality clients may be used.

This allows you to add additional users that can login to the Oracle DataLens Administration Web pages and change the passwords of existing users. These new users can be deleted as well.

> **Note:** User names and passwords are case-sensitive.

User admin is a special super-user that has access to parts of the Oracle DataLens Administration Web pages that other users do not have access. This user cannot be deleted, although you can change the default password.

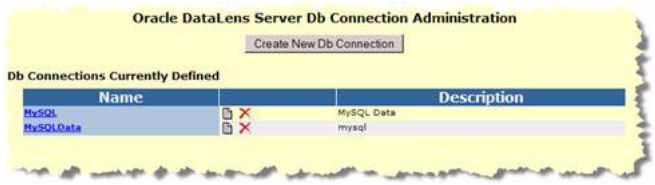
Users are created and assigned one or more roles, using the following screen.



Database Connections

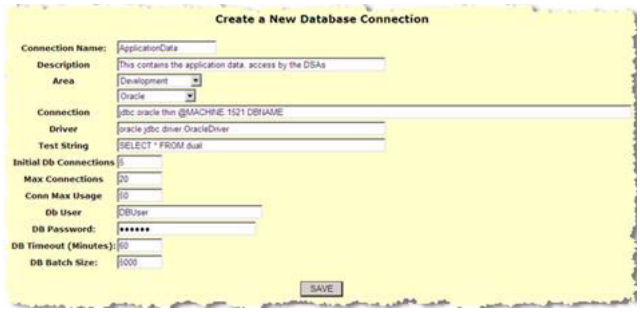
This page allows the administrator to create named database connections that can be used by DSAs across the Oracle DataLens Server topology.

The initial page will be empty; following is a page after several Db connections have been defined.



The names are those that can be used by any of the Jobs that need access to a database.

To create a new database connection the following page is used.



- Create a named connection to be used in all the DSAs.
- Select an area. Different user logins can be used for different areas. In fact, entire different databases can be used for different areas.
- Select a Database Type to pre-populate the Connection and Driver fields. There is default connection information for
  - User Defined - This allows the user to enter any connection information for alternate databases
  - Oracle
  - MySQL

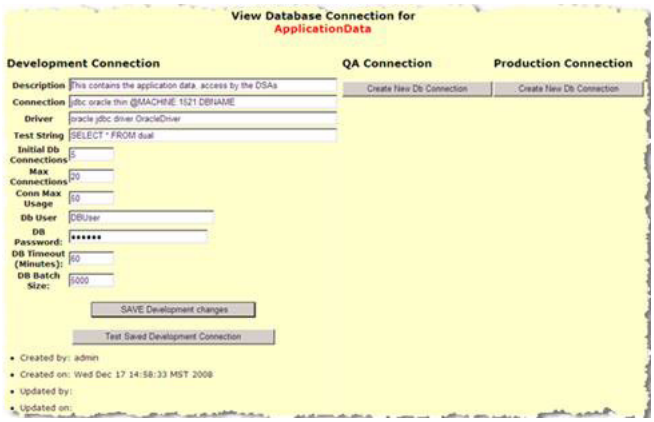
- SQL Server 2000
- SQL Server 2005
- PostgreSQL
- DB2

Use the default setting if unsure what values to use for the other options on this page.

Save the connection information and it will be added to the list of database connections.

Select the new connection that you have created to edit the connection and test that the connection definition is working properly.

> **Note:** This single page shows all the connection information for all the areas for which connection has been defined.



New connections for the other areas can also be created here for this same named database connection using the **Create** buttons.

There is also a **Test Saved Development Connection** button to check the saved database connection.

If your databases connection has been successfully defined, then a confirmation message is displayed.

If the connection definition has not been successfully defined, then an error message will be output, indicating the problem with the database connection.

> **Note:** You can edit the `DbConnectionTemplates.xml` file to add additional databases or modify the defaults for creating new database connections.

Web Services

Named WSDL connections can be created and configured for use in a DSA and its Transformation Maps (used by a DSA Step). These connections can be configured differently for QA, Development and Production servers, but all sharing a common name. This allows the Web Services to be different in different environments, but the DSAs will never need to change based on the deployment environment.

Note: The web service that is being called **must only return a single string**. All transforms in the Application Studio are expected to return a single string. This includes the web service calls.

Web services that return multiple fields (such as in the following example), return the multiple fields as character-separated data (such as a vertical bar) within the single string. In the Application Studio, you could then use the **Get Field** add-in function to extract the individual fields in the returned data.

In the following example, the WSDL creation wizard is used to create a named WSDL connection to the Oracle Product Data Quality Web Service to process a single line of data.

Click the **Web Services** link

Enter the name and description.

Enter a valid WSDL

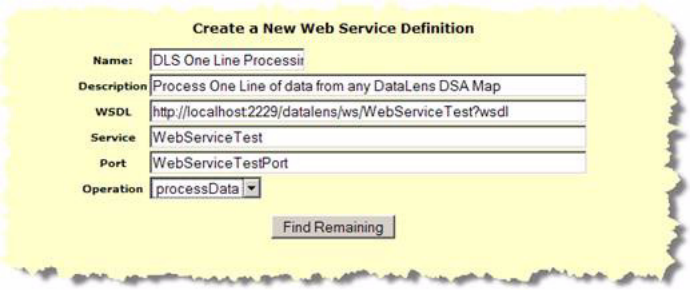
Click the **Find Service** button.

Select the service that you want from the Service drop-down selection box.

Click the **Find Ports** button.

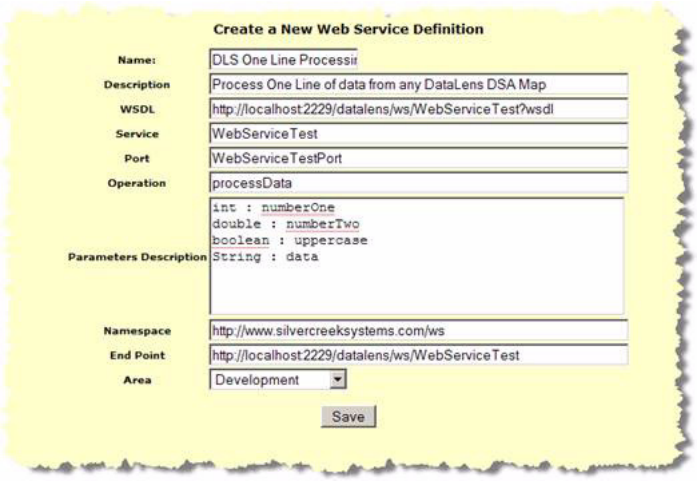
Select the Port that you want from the **Port** drop-down list box.

Click the **Find Operations** button.



Select the Operation that you want from the **Operation** drop-down list box.

Click the **Find Remaining** button.



In this example, the parameters are populated automatically from the Web Service and a sample line was added for testing.

The **Namespace** and **End Point** parameters are automatically populated.

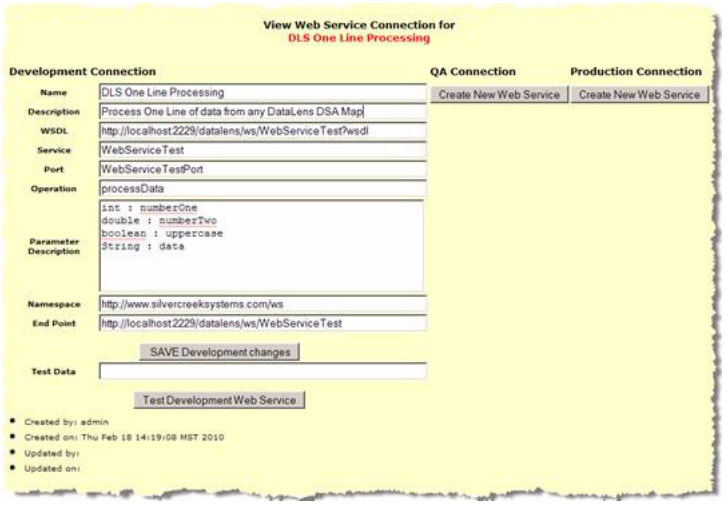
Select the **Area**.

> **Note:** It is useful to add a sample line for testing that you can cut and paste from for testing, as follows:

Click the **Save** button.

The Web Service Administration page is displayed with all Web Services.

Select a named Web Service definition to test the web service.



Enter the test data into the **Test Data** input field.

Click the **Test Development Web Service** button to test the connection by processing the data with the Oracle Product Data Quality Web Service.

The returned data is a single "|" separated string. This named Web Service is now ready for use in the DSAs.

FTP Connections

Named FTP connections can be created for use in DSAs and from DSA Transformation Maps (used by a DSA Step). The result data from a DSA job can be sent directly to one of these named FTP connections. Jobs run from the real-time programming interface can specify an FTP connection for the output.

Click the **FTP Connections** link.

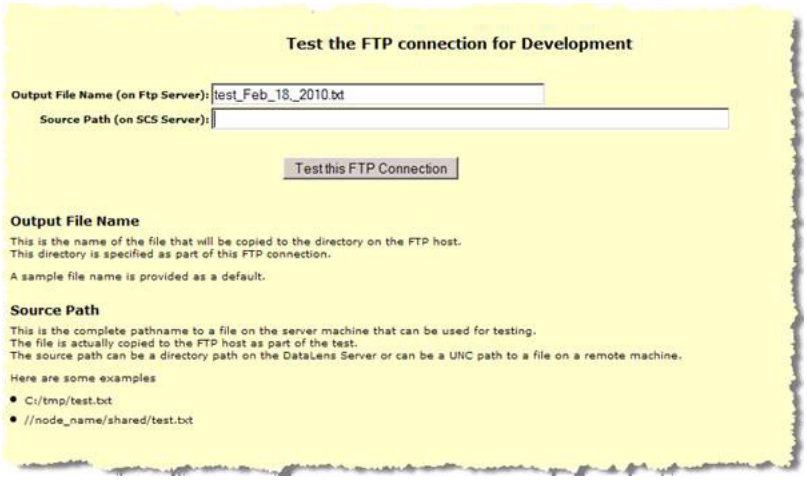
Select the **Create a New Ftp Connection** button.

1. Enter a name for the connection.

2. Specify a directory where the files will be placed in the remote FTP directory. This directory must be a sub-directory of the FTP connection destination directory, when connecting to the FTP host.
3. Enter the Port and Host.
4. Enter the user and password.
5. Save the new FTP connection definition.

The new connection is listed in the Ftp connections page.

The **Test** button can be used to test this new connection. During the test, a small file will be placed in the remote ftp directory. This is the test configuration page.



Select the **Test** button to test the connection. The source path needs to be a file on the server, or a UNC path of a file on the network (as shown previously).

If the FTP test is successful, then a confirmation message is displayed.

User Access

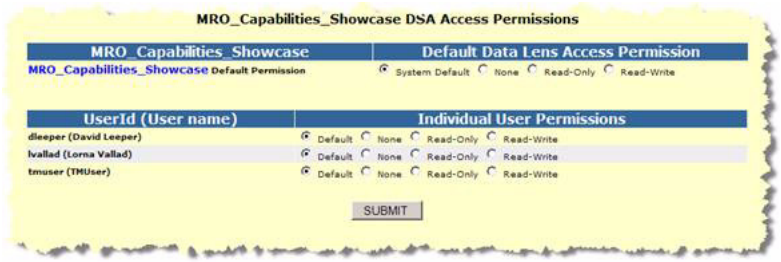
These options control the user access to the Oracle Product Data Quality applications such as the Knowledge Studio, Application Studio, and Governance Studio. This will also control the user access to the Web pages as well.

Data Lenses and DSAs

These pages allow the administrator to control exactly which users have access to which data lens and DSA, with specified levels of access for each user. This is in addition to the role-based permissions that these users are already assigned.

These three all work the same and DSAs are used in the examples in this section. A list of all the DSAs in the repository with a brief description is displayed.

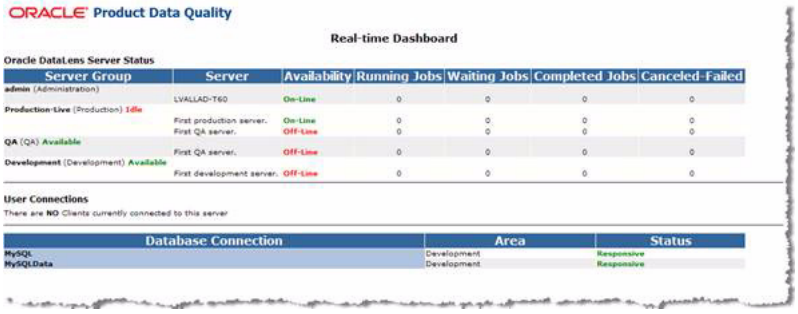
Select the hyperlinked name of the DSA to get to the user access page as shown follows:



The access each user is been granted is displayed. You can change permissions for any user and then click **Submit** to save the changes.

Dashboard

Click the **Dashboard** link.



This spawns a separate page that is used for Oracle DataLens Server monitoring. This page can be left up in a system command center for a company or enterprise to keep a constant monitor on the state of the Oracle DataLens Server. The page will refresh every minute.

Any problems with the Oracle DataLens Server are highlighted in RED.

- There are three users connected to the Administration Server from an Oracle Product Data Quality application, such as the Knowledge Studio.
- There is one database connection that is unresponsive. Any DSAs using these unresponsive database connections will prevent the applications from connecting to the database.

Server

The Server is where the data lenses and DSAs are stored for use by the Oracle DataLens Administration Server. This contains information on the different versions of the data lenses and DSAs as well as instances of the versions themselves. Only users of the Knowledge Studio, as well as users using the Administration Web pages for the Administration Server can directly manipulate these data lenses. There is *no* direct access to the repository from the Oracle DataLens Servers.

These data lenses are loaded to the server from the Knowledge Studio.

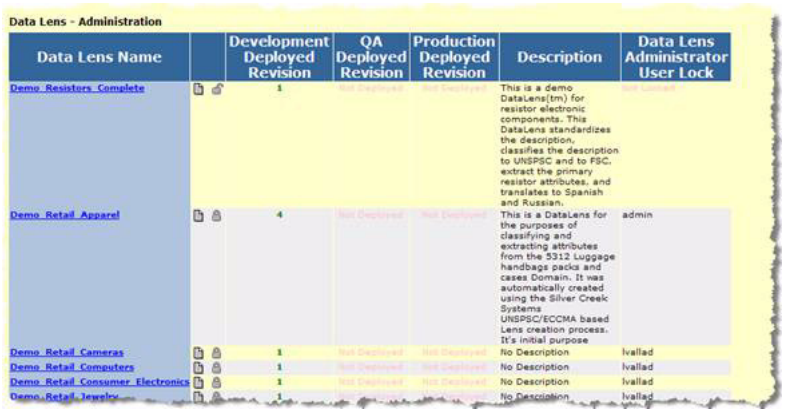
Server Data Lenses and Server DSAs are those that are being versioned by the Oracle DataLens Server for shared use by different team members. This prevents possible problems with more than one user updating a data lens at the same time and possibly losing work. This also provides a single location for backing up the valuable data lenses. Server data lenses can be checked out for edit or read by users of the Knowledge Studio.

- Data lenses are the basic unit of knowledge building.
- DSAs define the application of knowledge, data lenses, to record level data for the purposed of cleansing, attribute extraction, classification, and item level matching.

Once these are in the repository, any of the data lenses and DSAs can be edited and upgraded when needed.

Data Lenses

This page shows information about all current data lenses in the server in the Administration Server.


Data Lens - Administration

Data Lens Name		Development Deployed Revision	QA Deployed Revision	Production Deployed Revision	Description	Data Lens Administrator User Lock
Demo_Resistors_Complete		1	Not Deployed	Not Deployed	This is a demo DataLens(tm) for resistor electronic components. This DataLens standardizes the description, classifies the description to UNSPSC and to FSC, extract the primary resistor attributes, and translates to Spanish and Russian.	Not Locked
Demo_Retail_Apparel		4	Not Deployed	Not Deployed	This is a DataLens for the purposes of classifying and extracting attributes from the 5312 Luggage handbags packs and cases Domain. It was automatically created using the Silver Creek Systems UNSPSC/ECCMA based Lens creation process. It's initial purpose	admin
Demo_Retail_Cameras		1	Not Deployed	Not Deployed	No Description	lvallad
Demo_Retail_Computers		1	Not Deployed	Not Deployed	No Description	lvallad
Demo_Retail_Consumer_Electronics		1	Not Deployed	Not Deployed	No Description	lvallad
Demo_Retail_Jewelry		1	Not Deployed	Not Deployed	No Description	lvallad

Additional fields in this table include the following:

Data Lens Name

This is the name of the data lens, with a hyperlink to the complete history information for this data lens. The history area is where the specific versions of the repository Data lenses are deployed to the server (see the following.)

Actions

Edit		Fine-grained control over the version of the data lens to be deployed to the Administrator or Production servers.

Delete		This will permanently delete the data lens from the Server Repository.
Lock		This will lock the data lens with a named user.
Unlock		This will remove the user lock from the data lens.
Rename		Rename the data lens, keeping all the revision information.

Development Deployed Revision

The revision number of the deployed data lens to the Development (or the Administration/development) Server.

Black text means the Development-deployed revision is the last version that was checked into the repository.

Red text means the Development-deployed revision is *not* the latest version in the repository.

QA Deployed Revision

The revision number of the deployed data lens to the Quality Assurance data directory.

Production Deployed Revision

The revision number of the deployed data lens to the Production data directory.

Description

The initial description of the data lens when it was first checked into the repository. For more information, view the history of the data lens. This description can be updated from the Application Studio application.

Administration Server User Lock

This shows the Knowledge Studio user that has locked this data lens for exclusive access. "Not Locked" means that the data lens is available for checkout by any Knowledge Studio user.

User Locks

Typically, the user locks on the repository Data lenses are created by users of the Knowledge Studio. Occasionally, these locks will need to be overridden by the server administrator.

A user might be on vacation and left a data lens locked that is needed by another user.

A user might have inadvertently started editing a data lens in the Knowledge Studio and failed to get a user lock first. This user would not want to check out the data lens to get the lock because that would overwrite the work the user has done on the data lens so far. The administrator must be careful in this case because the data lens that the user started on (and did not check out) might not have been the latest version.

Removing User locks Just click the **Unlock** button in the second column to remove the lock that the user has on the data lens.

Adding User locks Click the **Lock** button in the second column to add a lock on this data lens

Data Lens Repository History

This hyperlink from the **Data Lens - Administration** page will display the complete history of the data lens listed from the most recent to the oldest.

Click the **View History** button in the second column to get to the following page.

Source

This is the language that the data lens uses as the "from" translation language.

Standardizations

This is a list of the alternate standardizations that the data lens uses. In this case there is only the default standardization being used, so this column is blank.

Classifications

This is a list of the Classifications that the data lens uses. In this case there are two separate classifications that are being used by the data lens.

Target Locales

This is a list of the languages that can be translated to.

Unit Conversions

This is a list of the numeric conversions that have been defined for this data lens.

The second table on the View History page is the version information and the deploy/undeploy icons as follows:

Revision

The particular revision, starting from one, listed in reverse order so the latest revision is always at the top of the list. In this example there is only 1 revision.

Development/QA/Production Deployment Icons

Select an icon to deploy a particular version of this data lens to the Administrator or Oracle DataLens Server. This means this data lens will be listed as a data lens available for loading on the Administrator or Oracle DataLens Server. In this example, there is not have a QA Server Group, so the QA deployment icon is not enabled.

Information

This shows the revisions of the data lenses including:

- Comments on what the Knowledge Studio user did to the data lens on this particular revision
- The name of the Knowledge Studio user that made the revision.
- The date/time that the revision was checked into the server

Administrator/Production Deployed Icons

These icons show which versions of the data lens are deployed to the Administrator and/or Oracle DataLens Server area.

Undeploy from Development

This will undeploy a data lens from the Development Server, which means it will not be listed as a data lens to be loaded as well. This button is only available for version 8 in the example, because that is the version that is deployed to the Administration Server.

Undeploy from Production

This will undeploy a data lens from the Oracle DataLens Server, which means it will not be listed as a data lens to be loaded as well. This button is *not* available in the preceding example, because this data lens is not currently deployed to the Oracle DataLens Server.

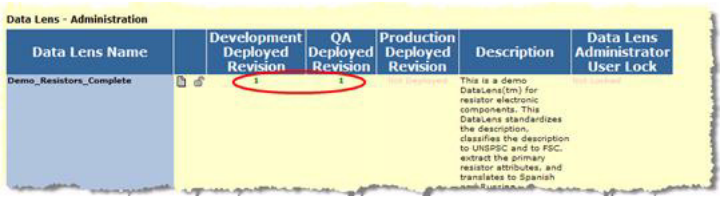
Undeploy from QA

If there is a QA server group defined, then there will be a button to undeploy or deploy to the QA area.

Deploying a Data Lens

From the history page of the data lens, select the **Deploy** ***data_lens_name*** **to Development** icon.

Now the data lens is deployed to the development area and is ready for use in the development group as in the following:



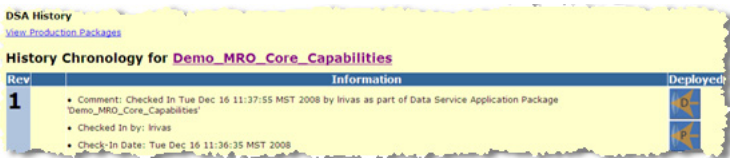
It is recommended that data lens only be deployed to the development area. Data lens deployment to the QA and Production areas can be done more easily by using the package deployment of DSAs, and the packages give a level of version tracking that cannot be done by manual data lens deployment.

Data Service Applications

This page shows information about all current DSAs in the repository on the server.

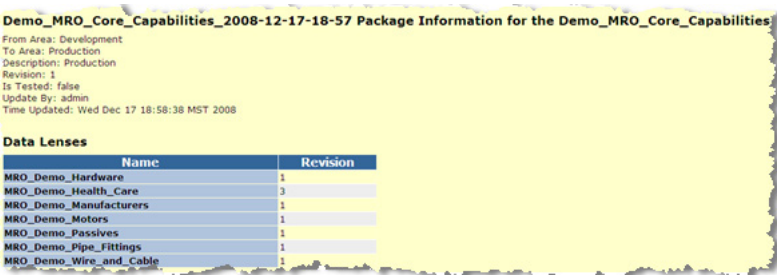
This page works the same as the Data Lenses page. See the preceding description for details on this page.

To view the packages that were used to deploy to the Production area, select the **View Production Packages** link from the main page for a particular DSA.



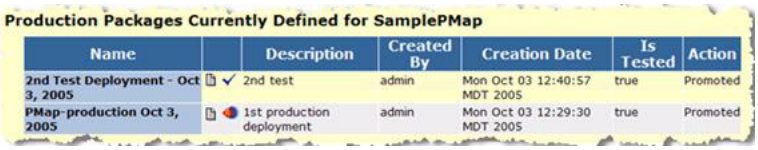
This will show the Production Packages as follows:

Drilling down into the details for this package using the view settings button gives the following details.



This reports that this package uses revision 1 of Demo_MRO_Core_Capabilities with a set of data lenses.

If there is more than one package for the DSA, then the package menu lists all the package versions as follows:



There are two new icons that are enabled when there is more than a single package. These are:

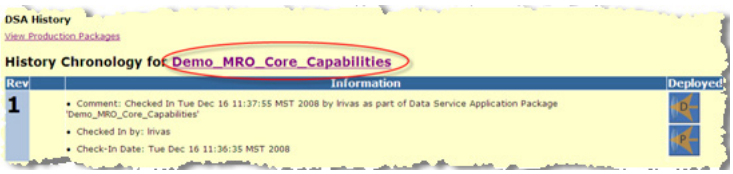
Current Package Tag

This marks the current package that is deployed to production.

Data Service Application Details

Details for a particular DSA can be obtained from the history page for a particular DSA.

> **Note:** Deploy and un-deploy icons are also available. Oracle recommends that only the Deploy to Development icon be used. For deployment to QA or Production, Oracle recommends that packages be used rather than these deploy icons, to be able to track versions in the QA or Production server groups.

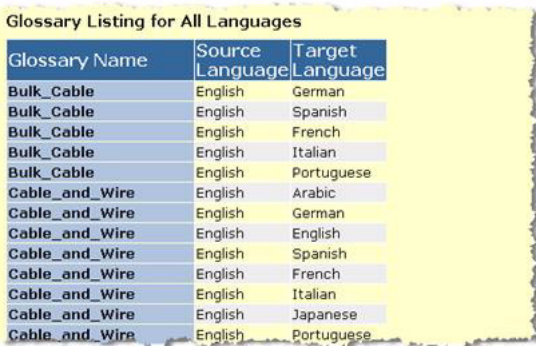


Click the DSA name link to view the DSA details as follows:

Language Glossaries

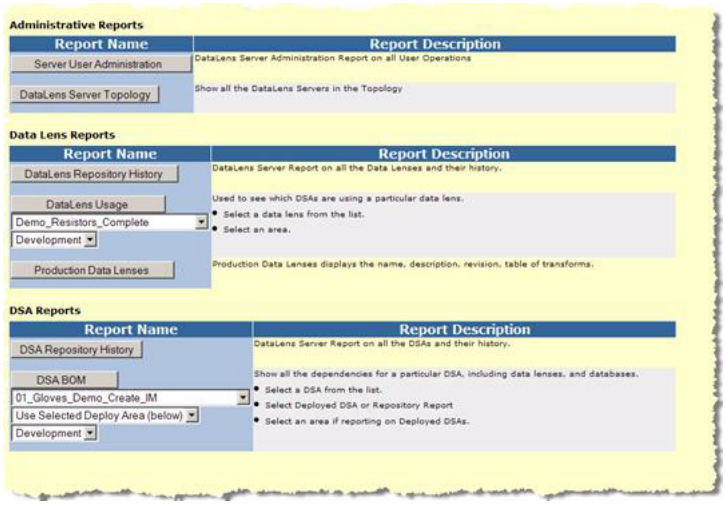
Click the **Language Glossaries** link to access information about the available translation Smart Glossaries.

This table is a listing of all the glossaries that are available to the Oracle DataLens Server topology. There is the glossary name with the source and target languages for the translation.



Reports

Click the **Reports** link to access all reporting functions.



Administrative Reports

These are generic reports on the state of the Oracle DataLens Server Platform Topology.

Server User Administration

This will show all the user activity on the Oracle DataLens Server.

DataLens Server Topology

This shows all the servers in the Oracle DataLens Platform Topology, with the capabilities of each server.

Data Lens Reports

Data Lens Repository History

This shows all the transformations that the data lens can perform along with a complete repository history.

Data Lens Usage

This is useful to see where a particular data lens is being used in the data lens topology for the selected deployment area.

Production Data Lenses

A detailed listing of all the data lenses that are deployed to Production.

DSA Reports

DSA Repository History

This shows a complete repository history of all the DSAs on the Administration Server.

DSA BOM Bill of Materials

This is useful to see what Data Lenses and database connections are used by a particular DSA. This can be listed from the repository or any of the selected deployment areas.

Job Status

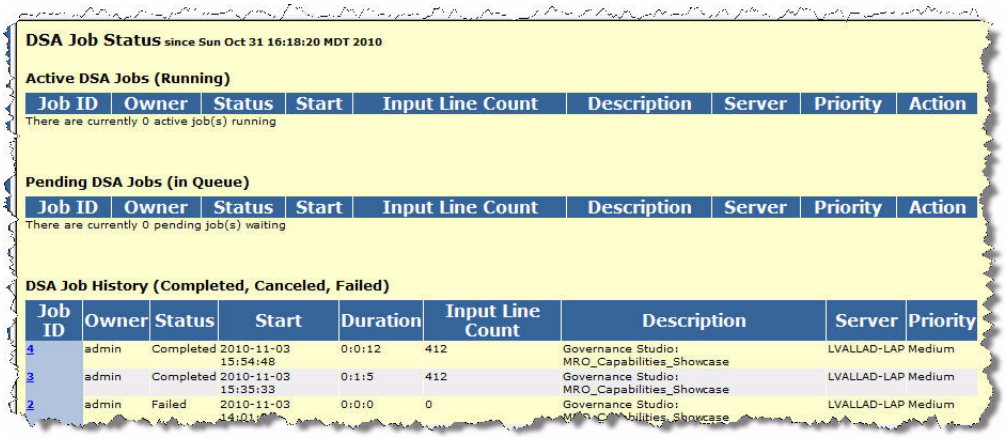
The administrator can view the status of DSA jobs and can change the state of the jobs on the Oracle DataLens Server.

Data Service Application Jobs

To view a list of the DSA jobs that have been run, click the **Job Status** link.

The page is separated into three types of DSA jobs.

- Those Active Jobs currently running
- Those Pending DSA jobs currently waiting in the queue
- The Job History with information on DSA jobs that have finished, have been cancelled, or have failed.



DSA Job Status since Sun Oct 31 16:18:20 MDT 2010

Active DSA Jobs (Running)

Job ID	Owner	Status	Start	Input Line Count	Description	Server	Priority	Action

There are currently 0 active job(s) running

Pending DSA Jobs (in Queue)

Job ID	Owner	Status	Start	Input Line Count	Description	Server	Priority	Action

There are currently 0 pending job(s) waiting

DSA Job History (Completed, Canceled, Failed)

Job ID	Owner	Status	Start	Duration	Input Line Count	Description	Server	Priority
4	admin	Completed	2010-11-03 15:54:48	0:0:12	412	Governance Studio: MRO_Capabilities_Showcase	LVALLAD-LAP	Medium
3	admin	Completed	2010-11-03 15:35:33	0:1:5	412	Governance Studio: MRO_Capabilities_Showcase	LVALLAD-LAP	Medium
2	admin	Failed	2010-11-03 14:01:[illegible]	0:0:0	0	Governance Studio: MRO_Capabilities_Showcase	LVALLAD-LAP	Medium

Active Jobs, Pending and Completed Jobs

The server will interleave multiple jobs so the jobs can run concurrently and a single very large batch job will not take control of the server.

The maximum number of concurrent jobs is controlled as part of the server configuration as described in "Configuration" on page 2-10.

An administrator can do two things with currently active batch jobs. The first is to look at the details of the job and the other is to cancel the job.

Job Id

The Job Id is assigned by the system during job submission and is comprised of the username followed by a job number.

Owner

This field shows the User Id of the person who submitted the job.

Status

This field shows the status of the job. Status definitions are as follows:

- **Running**

  The job is currently running.

- **Pending**

  The job has not started, but will start as soon as a slot becomes available. Only two jobs can run concurrently on the Oracle DataLens Server. Jobs submitted while two jobs are already running will get a status of "Pending" and will start in order of submission as the others jobs finish processing.

- **Finished**

  The job has successfully finished processing.

- **Cancelled**

  The Administrator canceled the job during processing or before processing started.

- **Failed**

  The job failed. This status means that something went wrong during the submission or processing of the data. Failed jobs will yield an entry in the Oracle DataLens Server Log.

Start
Shows the date and time the job started in YYYY-MM-DD HH:MM:SS format.

Duration
This shows the total time in hours/minutes/seconds for a completed job.

Input Line Count
This field shows the number of records processed so far for the DSA job. If a job is in "Running" status, this number will update when you click the browser's refresh button.

Description
The description the user entered when submitting the job. Additionally, jobs that are run from the Governance Studio or Services for Excel are identified as such.

Server
The server that the job was sent to for processing. In a server group with more than one server, there may be multiple servers handling the request.

Priority
The priority that the job was given.

- **Low priority**

  Jobs are large batch-type jobs processing tens of thousands and millions of lines of data.

- **Medium priority**

  Jobs are jobs where the results should be obtained while any low priority job is running.

- **High priority**

  Jobs are jobs with just a few lines to process, or jobs run from an interactive user environment, where the results need to be returned immediately.

Action
This is available for running or pending jobs. Two action buttons are available.

- **Cancel**

  This allows the administrator to cancel a running or pending job. The job may be on the incorrect data set or may just be too large for the user to run during the day when other users are also accessing the server.

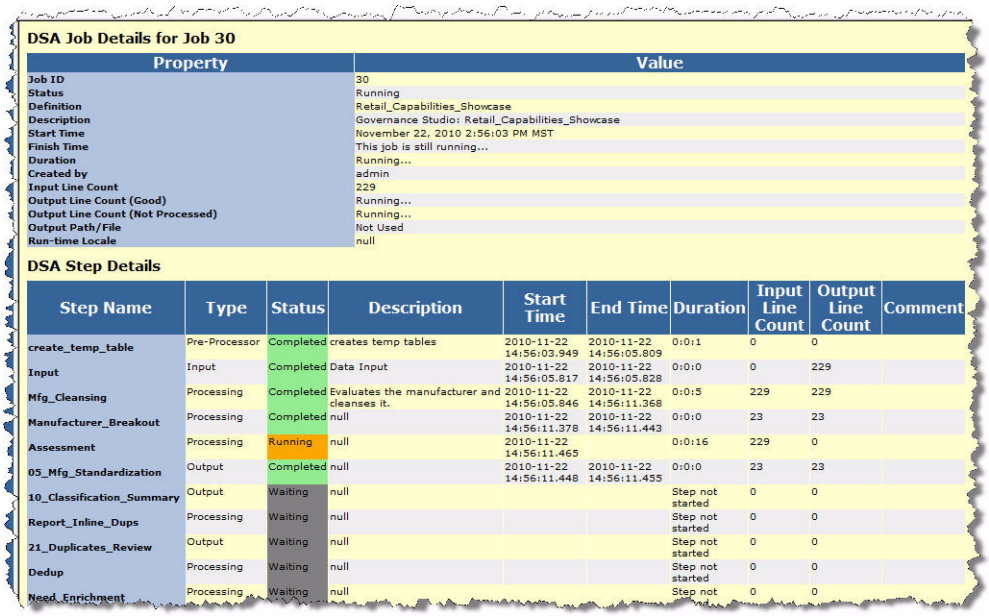
- **Priority**

  This allows the administrator to change the priority of a running or pending job.

Job Details

Select the hyperlink in the Job ID column for details on the currently running, pending or completed job.

By selecting the Job ID link in one of the Jobs tables, you can see the details of the selected job as shown in following the example:



DSA Job Details for Job 30

Property	Value
Job ID	30
Status	Running
Definition	Retail_Capabilities_Showcase
Description	Governance Studio: Retail_Capabilities_Showcase
Start Time	November 22, 2010 2:56:03 PM MST
Finish Time	This job is still running...
Duration	Running...
Created by	admin
Input Line Count	229
Output Line Count (Good)	Running...
Output Line Count (Not Processed)	Running...
Output Path/File	Not Used
Run-time Locale	null

DSA Step Details

Step Name	Type	Status	Description	Start Time	End Time	Duration	Input Line Count	Output Line Count	Comment
create_temp_table	Pre-Processor	Completed	creates temp tables	2010-11-22 14:56:03.949	2010-11-22 14:56:05.809	0:0:1	0	0	
Input	Input	Completed	Data Input	2010-11-22 14:56:05.817	2010-11-22 14:56:05.828	0:0:0	0	229	
Mfg_Cleansing	Processing	Completed	Evaluates the manufacturer and cleanses it.	2010-11-22 14:56:05.846	2010-11-22 14:56:11.368	0:0:5	229	229	
Manufacturer_Breakout	Processing	Completed	null	2010-11-22 14:56:11.378	2010-11-22 14:56:11.443	0:0:0	23	23	
Assessment	Processing	Running	null	2010-11-22 14:56:11.465		0:0:16	229	0	
05_Mfg_Standardization	Output	Completed	null	2010-11-22 14:56:11.448	2010-11-22 14:56:11.455	0:0:0	23	23	
10_Classification_Summary	Output	Waiting	null			Step not started	0	0	
Report_Inline_Dups	Processing	Waiting	null			Step not started	0	0	
21_Duplicates_Review	Output	Waiting	null			Step not started	0	0	
Dedup	Processing	Waiting	null			Step not started	0	0	
Need_Enrichment	Processing	Waiting	null			Step not	0	0	

This page contains the details of this currently running, pending, or historical (completed, canceled, or failed) DSA job. The data displayed is dependent on your selection on the **DSA Job Status** page, Active, Pending, or History.

This particular job is running and 229 lines of input data. You can easily see which steps have completed, are running, or waiting to run.

Job History

The administrator can look at the details of the job to get information and statistics on these finished jobs. In the configuration screen, the number of days of archived information can be controlled.

You can view the details of the job by clicking on the hyperlink with the job name to get the same details as shown in the preceding currently running jobs.

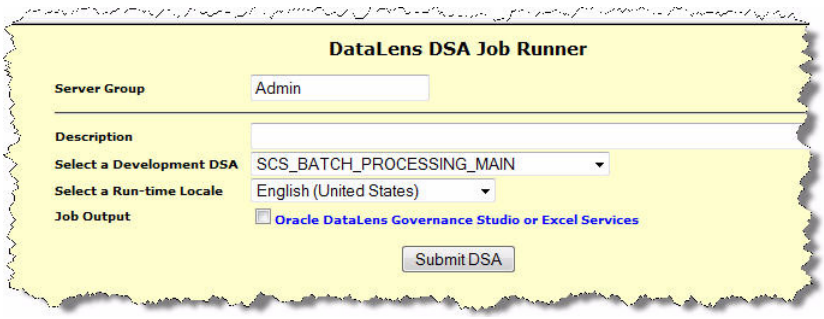
Run a Job

This allows an administrator to run a DSA job directly from the Administration Web pages. You must have the Web - Administer Job Scheduler permissions to access these menus.

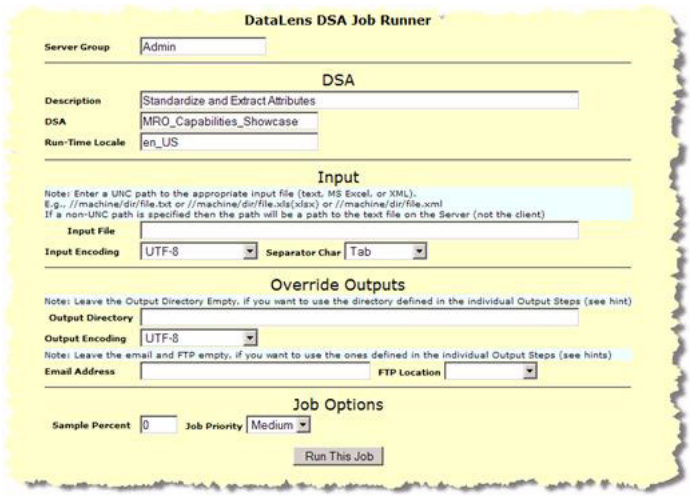
Following is an example of running a job using the **Run a Job** main menu item.

Select the Oracle DataLens Server Group that you want to process your DSA job.

Click the **Submit Server Group** button.



1. Enter a description for your job.
2. Select the **Data Service Application (DSA)** from the drop-down list.
3. Select the **Run-Time Locale** from the other drop-down list.
4. Select the **Job Output** check box if you want the results returned to the Governance Studio or Services for Excel.
5. Click the **Submit DSA** button.



1. Change the Separator character from the Tab character if needed
2. Change the Output encoding if needed.
3. Enter the Output Directory
4. Input File - Add the Full UNC pathname to a file on your network. Make sure that the directory with the file is shared and available on the network. A non-UNC pathname will write to a directory on the Admin Server.
5. Input File Encoding - Choose the input file encoding used.

> **Note:** UTF-8 also includes ASCII so most text files will work with this encoding

6. Separator Char - Select the type of field separator used in the input text file.

7. Output Directory - Add the full UNC pathname to a directory on your network to hold the output file. Make sure that this directory is shared and available on the network. A non-UNC pathname will write to a directory on the Admin Server.
8. Output File Encoding - Select the file encoding for the output file.
9. Email Address - Optionally add an email address to send the results to.
10. FTP Location - Optionally add a FTP location to send the results to.
11. Sample Percent - Enter a value for the percentage of the input file to sample, rather than using the entire file. The default of 0 will process the entire input file.

    If you have a huge file but only want to run a small test, change the sample percent to 1. This will randomly sample your input file, only sending 1% of the contents of the file to the server for processing.
12. Job Priority - Change the job priority if needed. Large jobs should be sent with a priority of Low and small jobs where you want the results immediately should be sent with a priority of high. If there are not multiple simultaneous jobs running on the server, then this will not matter.
13. Click the **Run This Job** button.

This will run the job and take you to the Data Service Application Job Status page that job statistics for the currently running job. For more information, see Job Status.

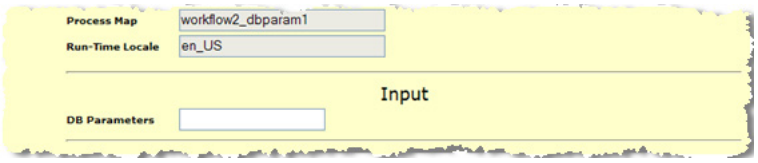
The finished job will output the results into the output directory and give the results a name as follows:

`JobId-ProcessMapName-StepName.txt`

For Example, a job was output to the `C:\tmp` directory with a file called `6-samplePMapIDef-output.txt`.

Database Jobs

If your job is a database job, then the input file screen will be replaced with a database parameter screen as shown below.



Enter a "|" separated list of database parameters.

Schedule a Job

This allows an administrator to run a DSA job directly from the Administration Web pages. You must have the Web - Administer Job Scheduler permissions to access these menus.

Click the **Schedule a Job** link to set a job to execute.

Select the Oracle DataLens Server Group that you want to process your DSA job.

Click the **Submit Server Group** button.

1. Enter a description for your job.
2. Select the **Data Service Application** from the drop-down list.
3. Select the **Run-Time Locale** from the other drop-down list.
4. Select the **Job Output** check box if you want the results returned to the Governance Studio.
5. Click the **Submit DSA** button.

1. Provide the requested information, which is the same as previously described in "Run a Job" on page 2-34.
2. Scheduled Time - Select the time of day for the job to be run.
3. Scheduled Recurrence - Select the time period to run this job.
   - One Time Only - The job is only run a single time.
   - Every Day - The job is run every day at the scheduled time
   - Every Weekday - The job is run every weekday (no weekends), starting with the initial run day.
   - Every Week - The job is run once per week on the day that the initial job is run.
   - Every Month (day) - The job is run on the particular day every month (2nd Wednesday for example).

- Every Month (date) - The job is run on a particular day every month (the 23rd for example).

4. Scheduled Duration (Start) - The specific day, month and year that the job is to start.
5. Scheduled Duration (End) - The specific day, month and year that the job is to end. Leave blank if the job scheduling has no termination date.
6. Data Options - These are the same as were documented for running a job immediately.
7. Click the **Schedule This Job** button to schedule the job.

This schedules the job and advances you to the **Scheduled Job Administration** page that lists the jobs that are currently scheduled.

Running the Job Scheduler on Server Groups

The job scheduler will run the jobs on a particular Oracle DataLens Server Group. The actual server that is selected in the server group is randomly selected by the Admin server as part of the round-robin server selection process.

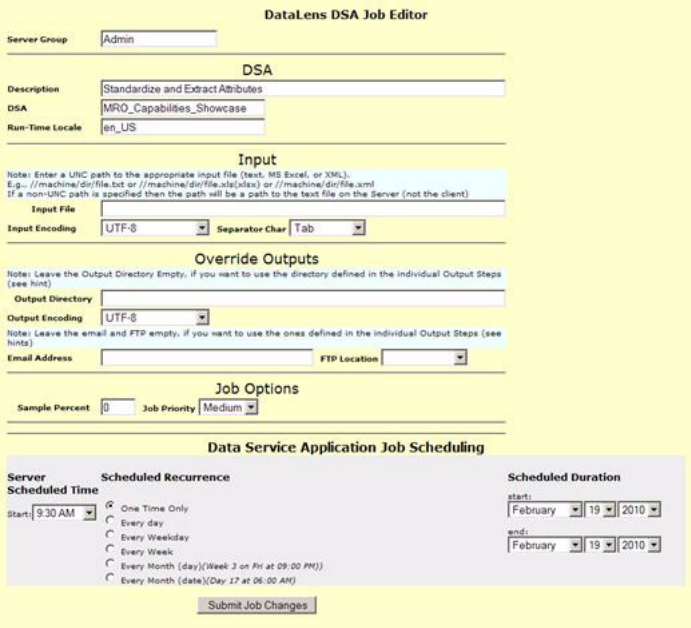
All the servers in the group are potential candidates to be the server that is selected to run the DSA This means that it is important that **all the servers in the group be setup to Load Data Service Applications**. This is done with the Administration Web pages in the **Platform Topology** pages.

Edit Scheduled Jobs

Click the **Edit Scheduled Jobs** link to modify a job that is scheduled to run.

Select the **Delete Scheduled Job** icon to delete a scheduled job. A confirmation query is displayed so that you can ensure that you want to delete the selected job.

Select the **View Settings and Edit Job** icon to change the parameters for a scheduled job.



All the parameters for your scheduled job can be changed, as previously described in this section, with the exception of the following:

- Server Group
- DSA
- Run-Time Locale

If you need to change the preceding parameters, you should create and schedule a new job.

Troubleshooting

Most of the troubleshooting information comes from log files. Look at the Log information from the Oracle DataLens Administration Web pages home page. Usually the information is needed, will be in this log. If there is a problem bringing up the server, so that this log cannot be accessed from the Web pages. For more information, see Appendix B, "Logging."

Non-English Characters Incorrectly Displayed

If you are running a Tomcat Application Server and non-English characters are not being displayed correctly, it is likely that the URI Encoding is incorrectly set in the `C:\Oracle\Middleware\opdq\tomcat\conf\`*`server`*`\server.xml` file. Edit this file and ensure that the `URIEncoding` flag is set to `UTF-8` as in the following example:

```
<Connector port="2229" protocol="HTTP/1.1"
   minSpareThreads="25" connectionTimeout="20000" URIEncoding="UTF-8"/>
```

Windows Platforms

The following sections provide solutions to issues found on Windows platforms.

Memory Allocation Problems on Clients

If memory allocation problems are experienced on client machines, an additional environment variable must be created for use by the OPDQ Java Web Start client.

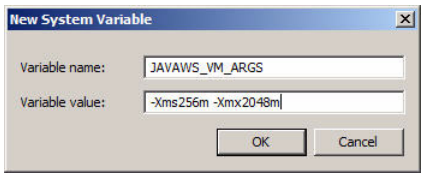
To add the necessary system environment variable and invoke it:

1. Right-click **Computer**, and then select **Properties**.
2. Click **Advanced system settings**.
3. Click **Environment Variables**.
4. From the **System variables** section, click **New**.
5. Enter **`JAVAWS_VM_ARGS`** as the **Variable name**.
6. Enter the **Variable value** to set the memory allocation values for the client using the following syntax:

   `-Xms`*`256`*`m -Xmx`*`2048`*`m`

   Where *`256`* is the minimum and *`2048`* is the maximum.

   Set these minimum and maximum memory values as appropriate for your client system.



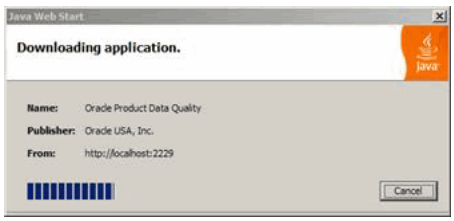
7. Click **OK** to close all dialogs.
8. Invoke the increased memory allocation for the OPDQ client applications, by initiating a connection and downloading the client software by browsing to:

   `http://`*`server`*`:2229/datalens/datalens.jnlp`

   Where *server* is the hostname of the Oracle DataLens Server.

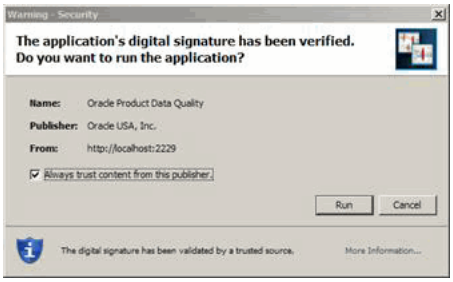
   Note: If you have setup a different port number for your application server other than 2229, you must use that port number in the following URL when browsing to the Oracle DataLens Server to download the client applications.

   The application download and verification begins.



Note: If you receive a **File Download** message indicating that the `.jnlp` file is not associated with a program, you do not have the supported JRE installed. Click **Cancel** and return to Step

After the verification completes, the installation begins. Oracle Product Data Quality files are digitally signed by a trusted source so the following security warning is displayed:



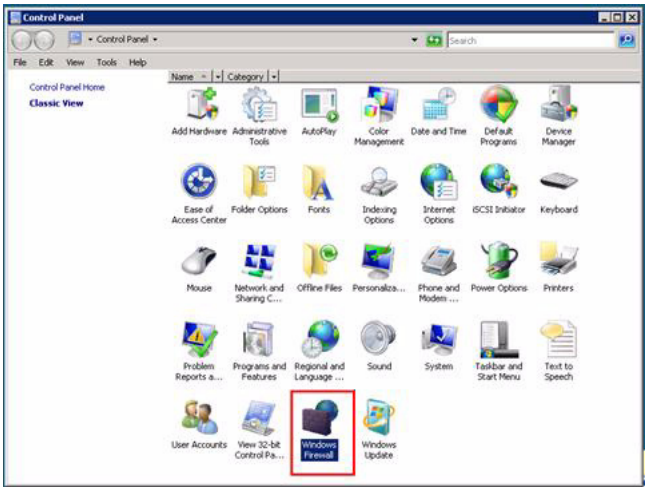
Problems Starting a Non-Admin Oracle DataLens Server

Two ports on the Admin server need to be accessible from the Oracle DataLens Servers. These are

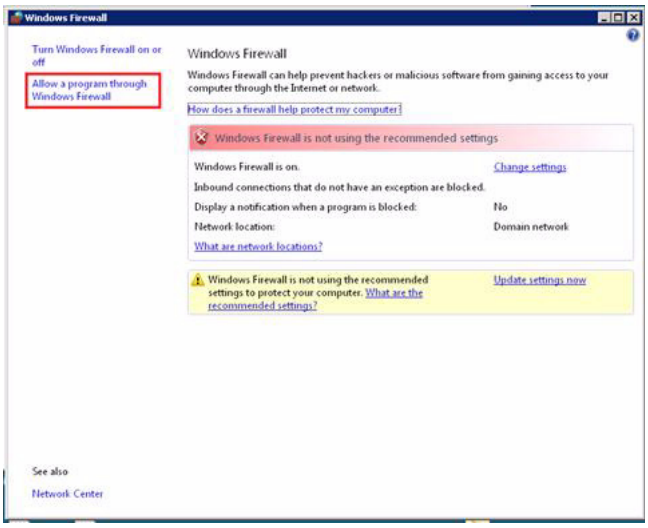
- 2229 - The IANA assigned port for the Oracle DataLens Server.
- 1527 - The port that is needed to connect to the data lens configuration information.

The following are the steps needed (on Windows) to open up these ports on the firewall if needed.

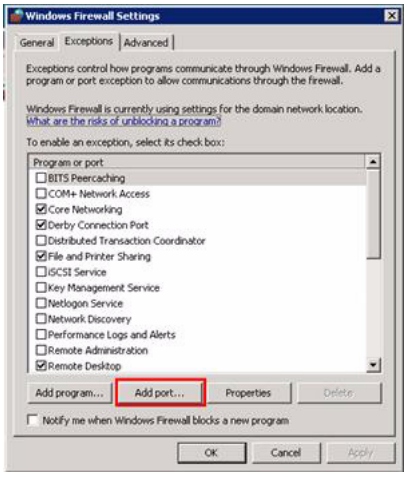
From the Windows control panel, start the **Windows Firewall**.



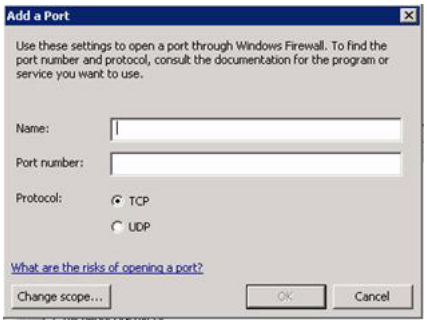
Select the **Allow a program through the Windows Firewall** link.



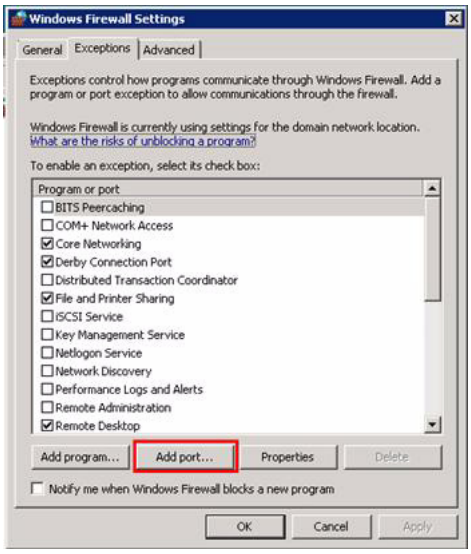
Select the **Add port…** button.



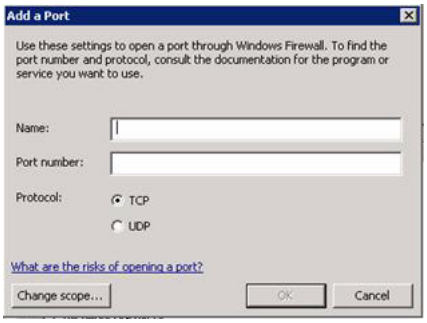
Enter PDQCMS Configuration Port in the **Name:** field and 1527 in the **Port number:** field.



Select the **Add port…** button again.



Enter PDQCMS in the **Name:** field and 2229 in the **Port number:** field.



Select **OK** and the Oracle DataLens Servers will be able to communicate with the Administration Server.

A

Configuration Files

This appendix describes updating the configuration file(s) changes the Oracle DataLens Server configuration and requires a server restart before the change will take effect.

Server Restart
Stop and re-start the server using the platform specific commands. Note that during the (short) Oracle DataLens Server downtime, active clients may fail to complete the current request.

File Types
You should not change any other properties in this file without first consulting with Oracle Consulting Services. Configuration files residing within the Oracle DataLens Server installation define the Oracle DataLens Server configuration. The Java Property file and XML document configuration file formats are used.

Server.cfg

This file is a Java property file. Unlike typical Java properties files, where properties are identified by a simple "key", data lens configuration files utilize a "path" based property specification. The path is a dot-separated list of names identifying the object. Lower case names in the path correspond to the application logical object structure, and the leaf element (starting with a capital letter) specifies the property name. The property value is separated from the property identifier by an equal sign.

For example, the following property file fragment may be used to configure the Administration Web Pages Statistics display:

```
# Any Production or Administration Server can send statistics if this flag is
# turned on. The Language server will not send statistics.
Server.send.statistics=true
```

The file `C:\Oracle\Middleware\opdq\`*`server`*`\config\server.cfg` defines the basic Servlet executive behavior as defined by data lens. Under normal circumstances, this file should not be changed. In fact, most of the parameters here can be modified from the Oracle DataLens Web Server Administration pages.

Some useful options for the `server.cfg` file are displayed in the following table.

`Server.send.statistics=true`	Any Oracle DataLens Server will send statistics if this flag is turned on.
`Server.qi.failure.count=100`	Shows the number of failures that will be accumulated by project. 0 or no entry means do not accumulate at all.

`Server.cookie.life.minutes=60`	Lifetime of login information stored in local cookies (default is 1 hour).
`statistics.chart.bgcolor=#FFFFCC`	Background color to use for the charts and chart keys.
`server.nle.instances=3`	Set the number of parameterized domain instances that will be loaded into memory. A single domain with two instances might want to set instances to 3. 1 for the first parameterized domain, another for the second, and a third for both in memory.
`wfg.maxmemorylines=5000`	Set the number of lines of data to keep in memory between the steps of a DSA. If the number of lines of data is greater than this parameter, then the data is written to disk between each step of the DSA, which hinders performance.

SvrPaths.xml

The `SvrPaths.xml` file defines the following parameters used by OPDQ:

OPDQ_HOME
This environmental variable defines the location of the Oracle DataLens Server home directory. The following are the default directories:

- **On Linux -**

  `/opt/Oracle/Middleware/opdq`

- **On Windows -**

  `C:\Oracle\Middleware\opdq`

serverID
This is 1 for Oracle DataLens Administration Servers and 2+ for Oracle DataLens Transform Servers. The value must be 0 when first created for Transform Servers; it is automatically updated by OPDQ.

localDir
Uses the default, `$OPDQ_HOME/local`, if not defined.

reposDir
Defines the repository directory and is only needed on an Oracle DataLens Administration Server.

derbyDir
Uses the default, `reposDir/internal`, if not defined.

sharedDir
Defines the directory that must be shared between the Oracle DataLens Administration Server and all Transform Servers.

devDir, qaDir & prodDir
Uses the default, sharedDir, if not defined.

logWatchSecs
Defines the number of seconds to check for changes to the `log.cfg` file.

```
-->
```

```
<SvrPaths>
  <serverID>1</serverID>
  <localDir></localDir>
  <reposDir>C:\datalens\server\data\repository</reposDir>
  <derbyDir></derbyDir>
  <sharedDir>C:\datalens\server\data\shared</sharedDir>
  <devDir></devDir>
  <qaDir></qaDir>
  <prodDir></prodDir>
  <logWatchSecs>0</logWatchSecs>
</SvrPaths>
```

Web.xml

The `web.xml` file defines the specifics of the Oracle DataLens Server application and is used to route requests to Oracle DataLens Server Servlets. It is located in the WAR file.

This file should not be changed.

B

Logging

This appendix describes the Oracle DataLens Sever log files.

Oracle DataLens Server Log File

The main log file used by the Oracle DataLens Server is located in the server home directory

`C:\Oracle\Middleware\opdq\`*`server`*`\log\dataserver.log`

This file is most easily accessed using the Administrator Web Pages described earlier in this document, unless the server is failing to come up.

Java Server Log Files

See the installation guide for information on the log files for the particular Application Server that you are using.

C

Server Configuration

Copying the Repository

This scenario may occur if there is a complete test system that needs to be copied to a production environment.

> **Note:** In a topology with a central Oracle DataLens Administration Server and development and/or production server groups, then this is not needed because the package deployment will take *care of this*.

Prior to starting this procedure, stop the application server service for the target server.

Copy the Directories

Basically, the repository files need to be copied from the test system to the production system. The repository contains the following directories that will need to be copied.

Simply copy the data directory from the Oracle DataLens Administration Server data directory; defaults to `OPDQ_HOME/data` (for more information about `OPDQ_HOME`, see "SvrPaths.xml" on page A-2. For example, `C:\Oracle\Middleware\opdq\server\data`.

This data directory contains the following subdirectories:

- `repository`
- `shared`

For example, you have a test server with the data repository in the root `DataLensData` directory.

You want to copy the repository to the prod server. You need to copy the data directory listed from the `//test/datalens/server` directory to the `//prod/datalens/server` directory.

Configure the New Server

In the server with the newly copied directories, change the configuration to point to these new directories. Edit the file `SvrPaths.xml` file if you have deviated from the standard directory location used in the test system. See the preceding information for information on editing the web.xml file.

Edit the
`C:\Oracle\Middleware\opdq\`*`server`*`\data\shared\config\DbInfo.xml`.

Change the name of the server to the new server hosting the prod server.

Now, the target server can be restarted with the new data directories.

Browse to the Administration Web Page and re-create the following:

- The database connections that are needed in the target system.
- The FTP definitions that are needed.
- User accounts, privileges and roles that will be used.

Job Continuation

The Job continuation feature allows large batch jobs to continue to run and new API jobs to start and run even when Transform Servers lose connectivity with the internal database repository (Administration Server is down).

When this occurs, you can identify what step the DSA was processing by selecting the **Job ID** from the **Job Status** page. The step that was running when the Transform Server lost connectivity is identified by black in the **Status** column and a comment to this effect in the **Comment** column as in the following example:

DSA Step Details

Step Name	Type	Status	Description	Start Time	End Time	Duration	Input Line Count	Output Line Count	Comment
DB Input	DB Input	Results Retrieved	Input from Database	2010-11-11 13:05:16.89	2010-11-11 13:05:16.899	0:0:0	0	0	
get_tracker_fields	Processing	Results Retrieved	Get Tracker Fields	2010-11-11 13:05:17.041	2010-11-11 13:05:17.486	0:0:0	2350	2350	
Map_Tracker_Fields	Processing	Results Retrieved	Map the Tracker Fields to the corresponding BugDB fields for rptheader	2010-11-11 13:05:17.555	2010-11-11 13:05:24.69	0:0:7	2350	2350	
process_bug_status	Processing		Process bug status and Fixed Version field so Fixed Version is null for status values < 20	2010-11-11 13:05:24.826	2010-11-11 13:09:37.0	0:4:12	2350	0	Server lost connection to database while job was running
Keyword	Output	Waiting	Keyword data from the function field			Step not started	0	0	

Expected Results

The following table outlines the expected results for each job type and the DB connection status.

	Db down at startup	Db down - Interrupted	Reconnect
Governance Studio	fail	run	run
Scheduled Jobs	fail	run	run
Java API	run	run	run
COM API	run	run	run

DB down at startup:
Means that the Transform Servers do not have access to the database.

Note: If there is no DB connection when the job is started, all new jobs will get a JobID of a negative number. The JobID's negative number range is based on the server profile id * 100,000.

DB down - Interrupted:
Means that the Transform Server had access to the database when the job began, but lost access to the database at some point during the running of the job.

Note the following:

- Submitted from Services for Excel, Governance Studio, or from the Job Runner web page the text results are not retrievable. If the results are persisted in the file system or database, they are placed there by the job itself and those results are available.
- Once the Administration server is back up, the Job status page will be incorrect for interrupted jobs in two ways. The job will be considered running and will have to be canceled to clear the job from the running list and it will have incomplete step information because the transform server could not write back the results to the database.

Reconnect:
Means that the Transform Server access to the database has been fully re-established.

D

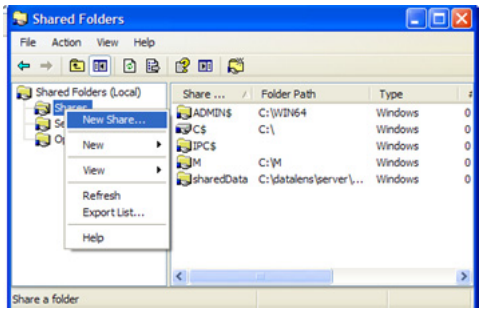
Mounting a Remote Repository

When mounting a remote repository, the only directory that needs to be mounted is the shared directory. This is typically located in the following directory.

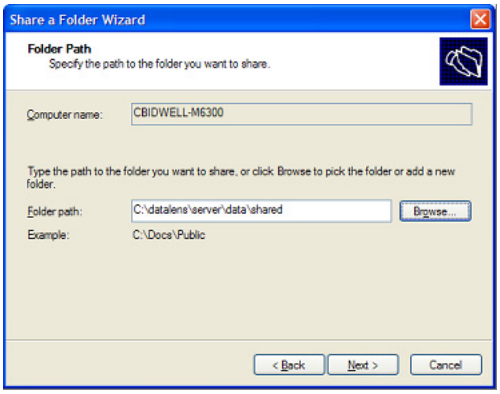
`C:\Oracle\Middleware\opdq\`*`server`*`\data\shared`

Windows-to-Windows Mounting

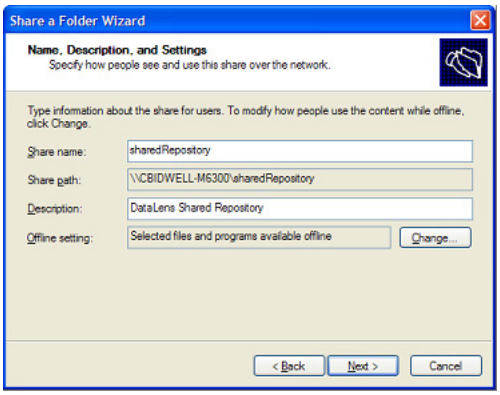
On the Admin server, click **Start->Run**, and type fsmgmt.msc in the **Run** dialog. This will open the shared folders administration tool. Click on shares, and right click to select **New File Share**.



Click next on the welcome page, then on the **Set Up a Shared Folder** page, enter `C:\Oracle\Middleware\opdq\`*`server`*`\data\shared` for the folder name.

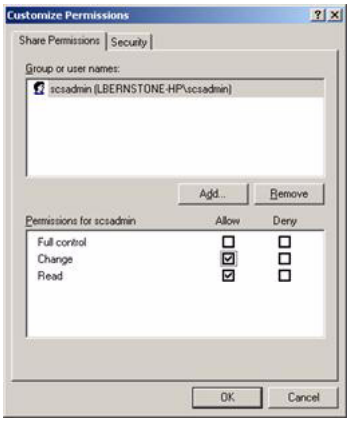


Enter **SharedRepository** as the share name and optionally a share description.



On the permissions page, choose to customize permissions, and then click the **Custom** button.

On the customize page, remove everyone from the list, and add your Application Server service user. Set the service user to have **Change** and **Read** permissions, and then click **OK**. Click **Next** on the permissions page, and then **Finish** to complete the share.



Once the share is created, the web.xml can be configured to use UNC paths, as long as all the Application Server service account users have the same name and password.

Linux-to-Windows Mounting

When connecting to a Linux Admin server from a windows production pod, samba can be used to provide SMB shares. On RHEL4, the samba-common and samba-server

packages will be needed. For more complex security environments, other packages and configuration may be needed.

Configure a samba share for your repository by appending the following to your `/etc/samba/smb.conf` file.

```
/etc/samba/smb.conf:
…
[repository]
comment = DataLens Data Repository
path = /opt/datalens/DataLensData
browseable = no
read only = no
valid users = dlsadmin
```

After modifying the file, restart samba with **`/etc/init.d/samba restart`**. The web.xml can now be configured using UNC pathing on windows pods as long as the user name and password (dlsadmin) is used as the service account for the Application Server on windows.

Windows to Linux Mounting

In an environment with a Windows Admin server and Linux nodes, `smbfs` can be used to mount a windows share. In RHEL4, this will require the samba-client and samba-common packages to be installed on the server. On the windows server, create a share called repository pointing to `c:\dlsdata`, as described in "Windows-to-Windows Mounting" on page D-1.

On the Linux servers, add the share to `/etc/fstab`.

```
/etc/fstab:
…
//adminsrv/repository /opt/datalens/DataLensData smbfs
credentials=/root/DataLensData,workgroup=adminsrv 0 0
The credentials file contains the following information, as configured in your
windows share:
/root/DataLensData:
username = DataLensData
password = HighlySecure
```

After modifying the files, run **mount -a** in order to activate the share. Proceed with configuration of web.xml in the next section.

Linux to Linux Mounting

In an environment with a Linux Admin server and Linux nodes, NFS provides a simple and robust method for mounting the `DataLensData`. On the Admin server, configure the exports file.

`/etc/exports`:

```
/opt/datalens/DataLensData        production1(rw) production2(rw)
```

After modifying the exports file, run **exportfs -a** as root on the server command line to have it refresh the nfs export cache. On the production servers, add the following to the fstab file.

`/etc/fstab`:

```
adminserver:/opt/datalens/DataLensData   /opt/datalens/DataLensData  nfs
```

```
defaults    0  0
```

After modifying the `fstab`, create the mount directory `/opt/datalens/DataLensData` and then run **`mount -a`** to mount the remote file system. Ensure that the dlsadmin user has the same UID and GID on all boxes with **getent passwd | grep dlsadmin**, as file access rights are determined by the numerical UID.

Configuring the SvrPaths.xml File

The `SvrPaths.xml` file will be automatically updated with the correct path to the shared repository files when installing non-administration Oracle DataLens Servers. If you need to make modifications to the repository after the initial installation, edit the `SvrPaths.xml` file in the `OPDQ_HOME/config` directory. Ensure that you do not to edit any of the xml tags.

E

User Authentication with LDAP

Lightweight Directory Access Protocol (LDAP) contains the functions to provide user authentication. It you use LDAP, then users do not need to be separately created in the Oracle DataLens Server Administration Web pages and assigned roles and passwords.

When LDAP is used, the Oracle DataLens Server contacts the LDAP server, and attempt to login with the provided credentials. If login is successful, it will find the group objects corresponding to its internal roles, and determine which of these groups the user is a member of in order to assign the appropriate rights.

Oracle DataLens Server LDAP Configuration Files

There are two files that need to be placed in the server `/config` directory (defined in `web.xml`):

- `ldap.keystore`
- `ldap.xml`

The Oracle DataLens Server will check for these two files in the `config` directory and will use LDAP user authentication if these files are found. The admin user within the DLS server will always be checked internally to prevent lockouts.

The ldap.keystore file needs to be created by your local IT Administrator with the Java keystore utility. Oracle Consulting Services can also create this file for you given the CA root certificate file generated by the LDAP directory server (for example, by Active Directory). The DLS supports LDAP and LDAPs connections, not SASL or negotiated TLS.

The `ldap.xml` file needs to be manually edited based on your local LDAP settings.

Following is an example `ldap.xml` file, configured for use with Microsoft Active Directory

```
<LdapInfo>
<!-- Connection Information -->
<host>AD-LDAP</host>
<port>636</port>
<!-- Security information -->
<keystoreFile>ldap.keystore</keystoreFile>
<!-Proxy Information à
<proxyUser>cn=proxytest,cn=users,dc=ad-ldap,dc=datalens,dc=com </proxyUser>
<proxyPass>password</proxyPass>
<displayNameAttribute>sAMAccountName</displayNameAttribute>
<!-- Users -->
<userPrefix>cn=</userPrefix>
<userSuffix>ou=dlsusers,dc=ad-ldap,dc=com</userSuffix>
```

```
<!-- Roles -->
<rolePrefix>cn=</rolePrefix>
<roleSuffix>ou=datalens,dc=ad-ldap,dc=datalens,dc=com</roleSuffix>
<memberAttribute>member</memberAttribute>
</LdapInfo>
```

The entries control the directory lookup behavior as follows:

`host`

This should be the LDAP server name or address. Host names must be able to be resolved on the Oracle DataLens Server.

`port`

Use port 389 for unencrypted, 636 for SSL connections.

`keystoreFile`

If a keystore file entry is included, SSL connections will be used to connect and make all queries against LDAP.

`proxyUser` (optional)

If a proxy user is included, an initial connection will be made as the proxy user to search recursively through the directory for a user whose entered name is found in the directory with a matching `displayNameAttribute`.

`proxyPassword` (optional)

Login password for the proxy user.

`displayNameAttribute` (optional)

If a proxy user is used, this will be the criteria used to search for the username entered in the login screens. For example, `sAMAccountName` is used for Active Directory, or UID in many other systems.

`userPrefix`

This is the attribute prefix on user objects, typically `cn=` or `uid=`

`userSuffix` (optional)

The base container in which to search for users. If a proxy user is not used, searches are conducted in this scope only, and are not recursive. If a proxy user is used, this tag is ignored.

`rolePrefix`

The attribute prefix for group objects, typically `cn=`.

`roleSuffix`

The base container in which to search for groups. This is a single scope search, and group names must exactly match the roles in the Oracle DataLens Server.

`memberAttribute`

The attribute of the group objects, which will contain the list of users within the group.

LDAP User Restrictions

The LDAP administrator must define roles (groups) that are used by the Oracle DataLens Server. These roles are mapped from LDAP to the Oracle DataLens Roles

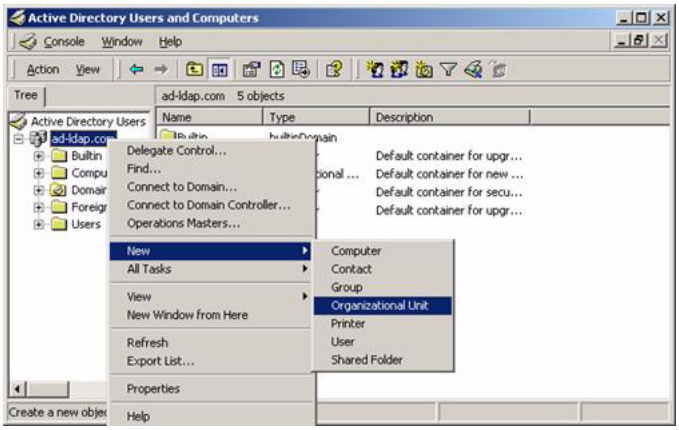
based on the name only. New roles can be created and used with LDAP, as long as corresponding groups are created in the LDAP directory.

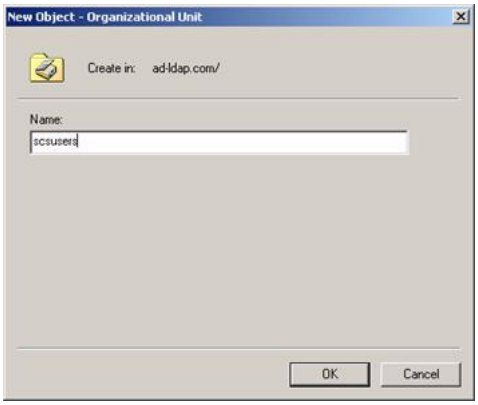
Standard role names for the Oracle DataLens Server include the following.

- Server Admin
- Knowledge Studio
- Application Studio
- Governance Studio User

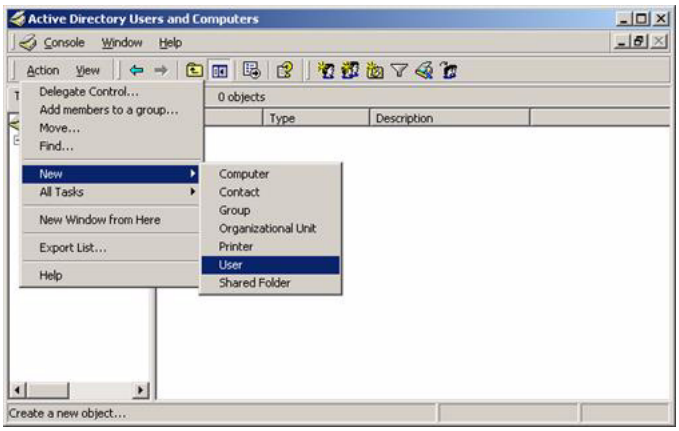
Setting Up a Sample LDAP Directory

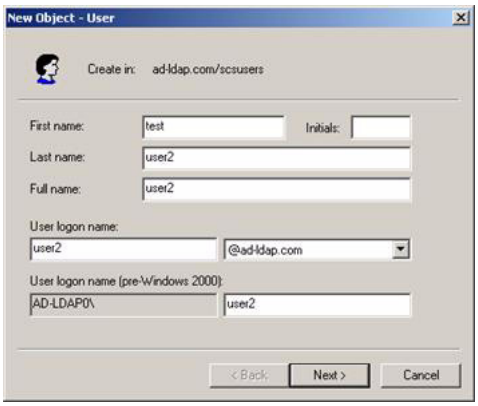
First, make a new OU (organization unit) to house your users:

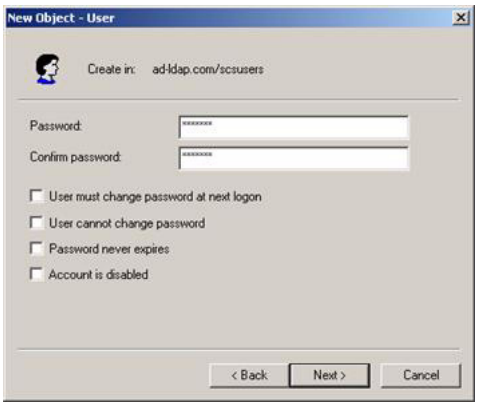




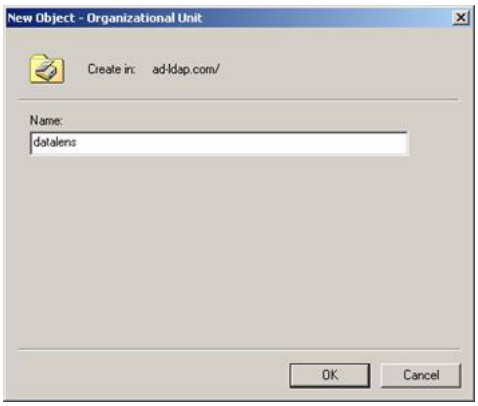
Now, make some users in that OU:



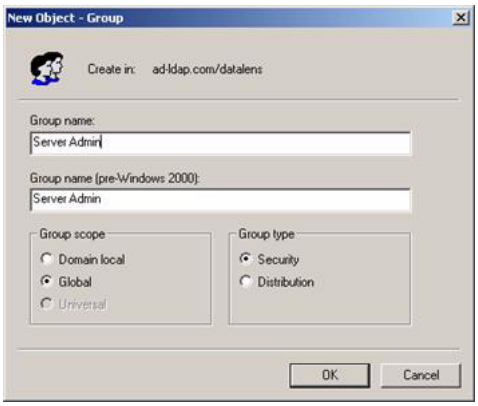


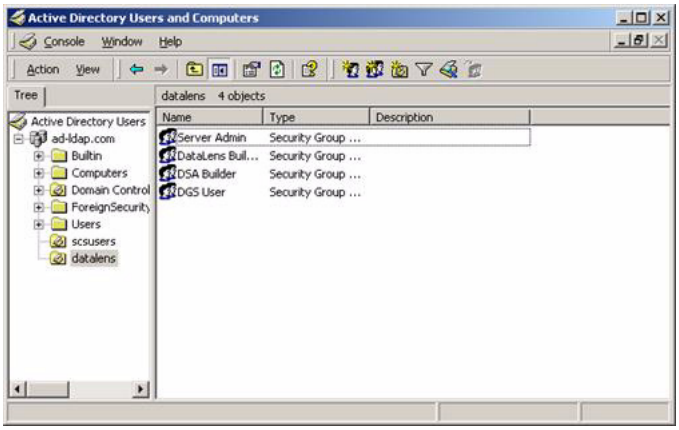


Then, make an OU for the Oracle DataLens roles:



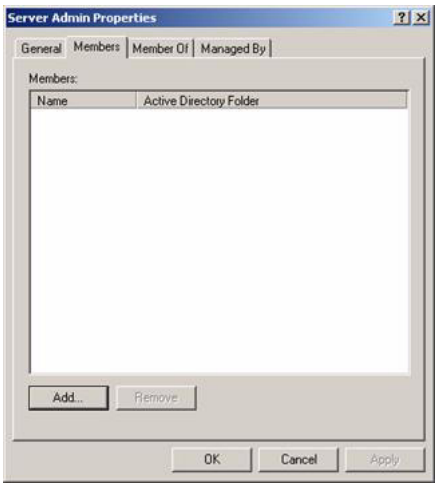
Create new groups for the Oracle DataLens Server roles:

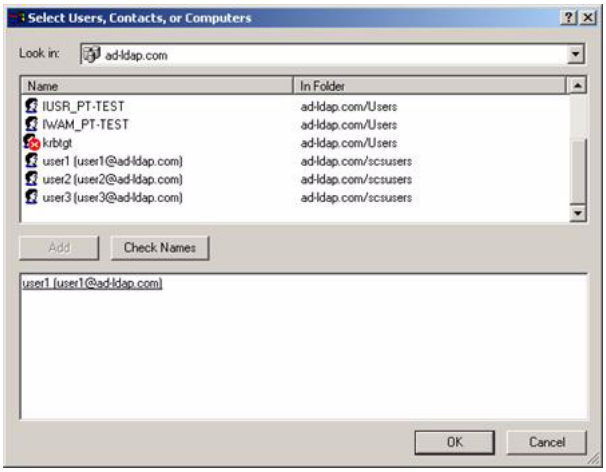


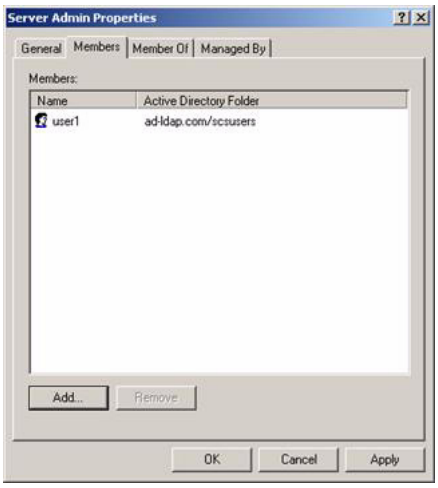


Then add users to the corresponding role groups.

> **Note:** Most likely, you will already have your network users set up. If this is the case, just add you existing users to the new Oracle DataLens Server group.







After configuring the Oracle DataLens Server with an appropriate ldap.xml and ldapstore.xml, the users will now be able to login with their Active Directory credentials.

ORACLE
PRODUCT DATA QUALITY
Powered by
DataLens™ Technology
Copyright © 2001, 2010 Oracle and/or its affiliates. All rights reserved.
Login Name:
Password:
Login

F

Tuning the Server(s)

This appendix describes steps that can be taken to improve the throughput of the servers. The emphasis is on running DSA jobs as fast as possible.

Checking the Results

The most accurate way to check the timing is to place a timer around the calls to run the DSA.

Another way is to look at the results of the job in the Administration Web pages and check the duration of the job as follows:

Process Map Job History (Completed, Canceled, Failed)

Job ID	Owner	Status	Start	Duration (seconds)	Input Line Count	Description	Server	Priority
135	Application_COM_Test	Completed	2006-10-25 09:53:02.0	4.0	105	QA Test Run (COM)	1	Medium
134	Application_COM_Test	Completed	2006-10-25 09:44:31.0	2.0	105	QA Test Run (COM)	1	Medium
133	Application_COM_Test	Completed	2006-10-25 09:43:20.0	0.0	105	QA Test Run (COM)	1	Medium
132	Application_COM_Test	Completed	2006-10-25 09:41:38.0	1.0	105	QA Test Run (COM)	1	Medium

Oracle DataLens Server Options

Load-Balancing the Servers

This cannot be taken advantage until there are two or more Oracle DataLens Servers in a single Server Group. The Oracle DataLens Server group will provide automatic load balancing and fail-over for all servers within a particular server group.

When running the application, be certain to call one of these production servers in the Server Group and *not* call the Admin server.

Manual load balancing can be performed for the servers in a single Server Group by selecting which data lenses are loaded by each server. Additionally, servers can be set to load DSA on a server-by-server basis. It is recommended that each server be setup with all the data lenses and DSAs and allow the Oracle DataLens Server to control the load balancing internally.

Round Robin Calls

When running DSA jobs from an application using the API, the Ping Servlet can be used to check for an active Oracle DataLens Server within a server group before making the call.

Ensure Tracing is Turned Off

This is turned off by default. Tracing is only turned on by Oracle Consulting Services to trace information flow in the system. This can be turned off in the Options menu of the Administration Web Pages. Additionally, there are a set of `scs.trace.network` flags that should be omitted or set to `false` in the `server.cfg` configuration file.

Data Service Application Optimization

Simplify the Data Service Application Process Steps

Each step in a DSA incurs additional overhead. This is because there is job information stored in the RDBMS repository for each the step of a DSA. Additionally there is overhead to package-up and ship the SOAP data contents from the DSA to each step during processing. What this means is that simplifying the DSA structure and placing as much of the process flow inside of Decision Maps will improve the speed of execution. We have observed timing improvements of up to .2 seconds for each DSA step that is replaced with a Decision Map.

Running Ultra High-Priority Jobs

Ultra-high priority jobs are supported. These DSA jobs do not store the step information in the RDBMS repository. The overhead of job execution is eliminated at the expense of job information and details of completed jobs. Especially for single-line jobs, ultra-high priority makes sense because the job execution will be as fast as possible and job details on thousands of single-line jobs will just clog up the DSA Job status Administration Web pages.

Run Jobs at the Correct Priority

The rule is that huge jobs should be run with a low priority, giving processing cycles to smaller medium and tiny high priority jobs. DSA jobs with a small number of input records and jobs where the user is waiting for a response need to be run at a high priority to get the fastest response time.

File Writing Between Steps

By default, when a DSA is being processed by the Oracle DataLens Server, all data will be held in memory, unless there are more than 5000 records being processed in a single DSA job. The speed of execution of these large jobs can be increased by setting the number of data records that are held in memory between these processing steps. This is controlled in the Oracle DataLens Server.cfg file with the following line:

```
wfg.maxlines=150000
```

Data Lens Optimization

Caching the Data Lenses

Individual data lenses can cache parsing rules in memory for re-use without re-loading the rule each time. This is mostly useful for data processing by data lenses

that reuse the same data repeatedly. Examples of this would be manufacturer names, redundant data, part numbers that are reused often. Data lenses that are not a good candidate are those that process things like descriptions that are different each time and would require a different parse tree for each line.

The cache should be large enough that the most often repeated lines are allowed to stay in memory (using a LRU Queue where the least often used rules will drop out of memory). For instance if there are 300 manufacturer names that are often reused among several thousand names, then the cache should be set to 1000 or perhaps 2000 depending on the frequency of use, to ensure that the 300 most often used names continue to reside in memory.

This is change is required for each data lens that need the caching.

- Check out the data lens to the client
- Go to the C:\Datalens\Applications\data\cbidwell\project\CablesF\config directory
- Edit the Project.xml file and modify the following line to the cache size

  ```
  <parseTreeCacheSize>0</parseTreeCacheSize>
  ```

- Save and check-in the project after making this change.

Do Not Load Data Lenses That Are Not Being Used

When running in a production environment, the number of data lenses is controlled by the lenses that are deployed to Production. Do not deploy data lenses to Production if they are not going to be used for actual production DSA jobs.

Fine-tuning of which data lens are used by a particular server can be controlled by setting the particular data lenses that are loaded by a particular Production Oracle DataLens Server.

Tuning Multiple Parameterized Domains

Set the number of parameterized domain instances that will be loaded into memory. A single domain with two instances should set instances to three to maximize performance when using these domains.

- One for the first parameterized domain

- Another for the second
- A third for both in memory

This is set in the `server.cfg` file as follows:

```
server.nle.instances=3
```

API Integration

WSDL Versus Java API Calls

The WSDL definition will create a dynamically generated Java API call that should have the same performance as the Oracle Product Data Quality Java API. Which method you use should be based on your current architecture, but not be based on any performance considerations.

Optimize the Available Hardware and Operating Systems

Windows Memory and Application Servers

See the section **Tune memory usage on the servers** for information on memory limitations of Windows servers.

Linux Memory, Windows Memory, and Java Servers

Linux running on 64-bit hardware does not have the 1.6 GB memory limitation for Java Web Server that we have observed on 32 bit Microsoft Windows servers. Windows 64 bit servers do not have this memory limitation either.

Important: In an Oracle Product Data Quality production environment, *only* run on a 64-bit server running a 64-bit installation of Java. *Never* try to run a production environment on any 32-bit servers.

Database Query Tuning

In database-intensive DSAs, major performance improvements can be made by tuning the database DDL statements. Simple things like indexing fields that are being searched on and reducing the number of tables in computationally intensive SQL joins can be very effective in improving the performance of the DSAs.

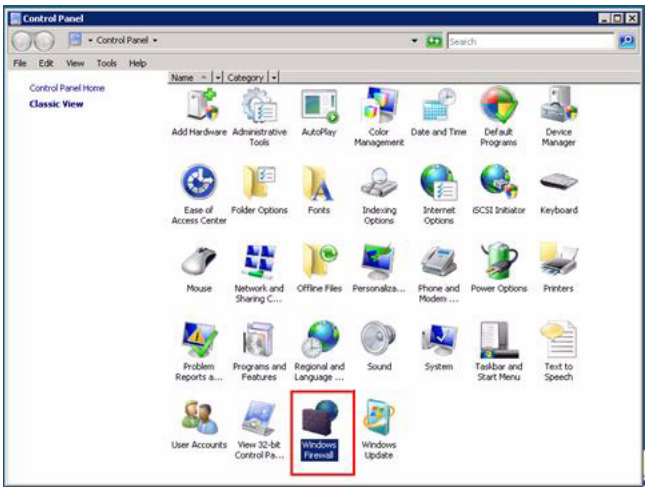
These tuning tasks are very dependent on the particular database schema and would need to be examined by a database professional or Oracle Consulting Services.

G

Installing East Asian Language Support Files for Windows XP

This appendix describes how to install East Asian Language Support Files for Windows XP.

To install support for East Asian Languages in Windows XP, open **Control Panel** and double-click **Regional and Language Options**.

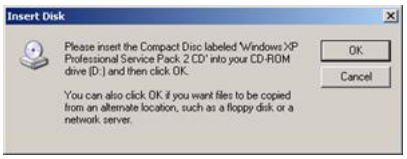


Click on the **Languages** tab and check the box next to **Install files for East Asian Languages**. Click **OK**.

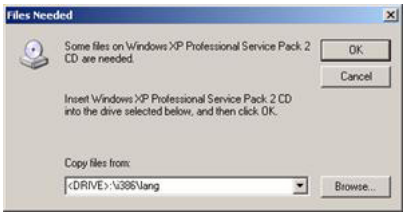
Read the dialog box that appears. Assuming you have 230 MB free to accommodate the installation, click **OK**.

Back on the **Regional and Language Options** dialog box click **OK** or **Apply** to begin the installation.

If prompted for installation media, click **OK**.

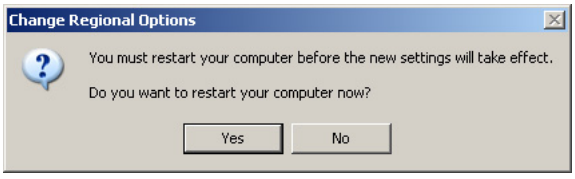


Browse to *DRIVE:\i386\lang* where *DRIVE* represents the location of the installation media. Click **OK**.



Depending on your system, you may be prompted a second time to insert your installation CD. If so, browse to *DRIVE:\i386* where *DRIVE* represents the location of the installation media. Click **OK**.

When asked to restart your computer, click **OK**. The installation is now complete.



The process to remove East Asian Languages support files from your system is simply the reverse of the installation process