

Oracle® Life Sciences Data Hub

System Administrator's Guide

Release 2.4.8

E95829-01

May 2018

Copyright © 2006, 2018, Oracle and/or its affiliates. All rights reserved.

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	ix
Finding More Information	ix
Documentation Accessibility	ix
1 Setting Up Services	
About Services	1-1
About the Distributed Processing Server	1-1
Setting Up the Distributed Processing Server	1-3
Defining Service Locations	1-3
Defining Services	1-4
Service Types	1-8
Service Prioritization	1-9
Service Instances.....	1-9
2 Setting Up Adapters to External Systems	
About Adapters.....	2-1
Adapter Structure	2-2
Creating Your Own Adapters	2-4
Setting Up Security for Adapters	2-4
Assigning User Groups to Adapter Domains and Adapter Areas	2-4
Creating Roles with the Required Operations for Adapters	2-5
Clearing Oracle Applications Cache	2-6
3 Setting Profile Values	
How to Log In to Oracle Applications Profile Forms.....	3-1
Setting Native Client Encoding.....	3-2
Using Character Semantics for Work Area Table Installation	3-2
Gathering Work Area Schema Statistics.....	3-3
Creating a New Tablespace for Work Area Schema Objects.....	3-4
Setting the Default Value for: Force Output Validation Status to Development	3-5
Setting the Maximum Number of Nested Domains	3-5
Adding the Direct SQL Option to SAS Libraries	3-6
Allowing User Preferences for SAS Clients.....	3-6
Set Default SAS Number Format.....	3-7
Set Processing Type for OC Data Extract Incremental Adapter.....	3-8

Turn debugging on or off	3-8
Setting Password Requirements	3-8
Setting Password Case Sensitivity (Required for Business Intelligence Publisher)	3-9
Setting Password Length Requirement	3-9
Setting "Hard to Guess" Requirement.....	3-10
Enabling "Forgot Your Password?" Functionality.....	3-10
Setting a Limit on Log-In Attempts.....	3-11
Setting a Time Limit on Password Reuse after Resetting Password	3-11
Oracle LSH: BIP Endpoint	3-11
Attachment File Upload Restriction Default	3-12

4 Adding and Modifying Lookup Values

About Lookups	4-1
Modifiable Lookups	4-1
File Types.....	4-2
Font Name.....	4-2
OTD Default Papersize.....	4-2
Report Type.....	4-2
SAS IDE Client.....	4-2
Service Type.....	4-2
Summary Output Validation Status.....	4-3
System Validation Steps.....	4-3
Tablespace Names.....	4-3
Technology Types	4-3
Logging In	4-3
Querying and Viewing Lookups	4-3
Adding, Modifying, or Discontinuing a Lookup Value	4-4

5 Registering Locations and Connections

About Remote Locations and Connections	5-1
Defining Remote Locations.....	5-2
Defining Connections	5-4
Searching for Remote Locations and Connections.....	5-5

6 Enabling Internal Metadata Table Auditing

About Internal Table Auditing.....	6-1
Audited Tables.....	6-2

7 Setting Up Printers

Setting Up Printers Globally.....	7-1
Setting Default Printers	7-1

8 Setting Up the Classification System

About Classification	8-1
Creating and Maintaining Hierarchies	8-2

Creating Hierarchies.....	8-3
Adding Levels to Hierarchies.....	8-3
Activating the Hierarchy and Making it Usable or Unusable.....	8-4
Adding Terms to Levels.....	8-5
Modifying Hierarchies and Terms.....	8-5
Assigning Levels to Object Subtypes.....	8-6

9 Setting Up the Security System

About Security in the Oracle Life Sciences Data Hub.....	9-1
About Security in Oracle Health Sciences Data Management Workbench.....	9-2
Setting Up User Accounts.....	9-3
About Users.....	9-3
Creating User Accounts.....	9-4
Maintaining User Accounts.....	9-4
Assigning Application Roles.....	9-5
Setting Up Specialized Administrators.....	9-7
Setting Up Object Security.....	9-9
Creating and Maintaining Object Subtypes.....	9-9
Creating and Maintaining Object Security Roles.....	9-11
Assigning Subtype Operations to Roles and Modifying Assignments.....	9-12
Creating and Maintaining User Groups.....	9-13
Granting Security Access to APIs.....	9-16

10 Creating Database Accounts

About Database Accounts.....	10-1
Database Accounts for Use in Definition.....	10-1
Database Accounts for Message-Based Submissions.....	10-2
Creating Database Accounts.....	10-2
Managing Database Account Privileges for Generic Visualization Business Areas.....	10-3
Setting Up TMS Security for Users.....	10-4
Resetting the Password.....	10-4
Removing a Database Account.....	10-4

11 Setting Up Oracle Business Intelligence Visualizations

Setting Up a Single-Computer Configuration.....	11-1
Setting Up a Multiple-Computer Configuration.....	11-3
Service Locations and Services Required in a Multiple Computer Installation.....	11-4
Single and Multiple Repository Files.....	11-6
Set Up Database Connectivity.....	11-6
Assign Repository Filename(s) to each Oracle BI Presentation Server Instance (10g Only)..	11-6
Add an Entry to the NQSCONFIG File for Each Repository File.....	11-7
Set up the Analytics Web DSN to Use the Repository File.....	11-7
Copy and Edit Revised Execution Command Scripts.....	11-8
Set the Default Authenticator to Optional (11g Only).....	11-9
Starting the WebLogic Server (11g Only).....	11-9
Set Up Oracle Business Intelligence Administration Tool.....	11-10

Security Configuration	11-10
Data Security	11-10
Creating an OBIEE Remote Location and Connection for RPD Password Security	11-11
Using Separate URLs to Control Access	11-12
12 Setting Up Security for Oracle Business Intelligence Publisher	
Overview	12-1
Creating and Assigning Oracle Applications Responsibilities	12-2
Creating User-Specific Responsibilities in E-Business Suite	12-2
Assigning Application Roles in Oracle LSH	12-3
Configuring Security in Oracle BI Publisher	12-4
Setting Up the Security Model and Creating the Local Superuser Account in Oracle BI Publisher 12-4	
Creating Data Sources and Assigning them User-specific Roles	12-5
Creating the SYSTEM Folder	12-5
Storing the BI Publisher Local Superuser Credentials in Oracle Life Sciences Data Hub	12-6
Creating the Oracle BI Publisher Remote Location in Oracle LSH.....	12-6
Creating the Oracle BI Publisher Remote Connection in Oracle LSH.....	12-6
The LSH: BIP Endpoint Profile.....	12-7
13 Setting Up Security for Informatica	
Setting Up User Accounts	13-1
Informatica Administrator Account	13-1
Accounts for Each Oracle LSH Definer	13-1
Creating an Informatica Remote Location and Connection	13-2
Creating the Informatica Remote Location in Oracle LSH	13-2
Creating the Informatica Remote Connection in Oracle LSH	13-3
14 Customizing Object Validation Requirements	
About Customizing Object Validation	14-1
Modifying the Custom Validation Package.....	14-1
15 Synchronizing an Apps Password Change	
16 Monitoring Jobs	
About Execution in the Oracle Life Sciences Data Hub	16-1
Querying for Jobs	16-1
Job Information	16-3
General Job Information.....	16-3
Job Execution Details	16-4
Job Statuses	16-7
Execution Statuses	16-7
17 Exporting and Importing Objects	
About Exporting and Importing Objects.....	17-1

Objects Included and Excluded	17-2
Objects Included.....	17-2
Objects Excluded.....	17-3
Source Independence	17-4
Exporting Objects	17-4
Full and Partial Export.....	17-5
Roles and Privileges Required for Exporting.....	17-6
Creating the Directory and Running the Export Script.....	17-6
Export Log File Content.....	17-7
Importing Objects	17-8
Upgrading an Existing Domain.....	17-8
Roles and Privileges Required for Importing Domains.....	17-9
Creating the Directory and Running the Import Script.....	17-9
Import Log File Content.....	17-10
Applying Security to an Imported Domain.....	17-11
Installing Objects	17-11
Running the Installation Script.....	17-12
Preserving Customizations	17-13

18 Stopping and Starting Services and Queues

Starting, Stopping, and Checking the Distributed Processing Service	18-1
Starting the Distributed Processing Server Service.....	18-2
Checking the Distributed Processing Server Service.....	18-3
Stopping the Distributed Processing Server Service.....	18-4
Stopping and Starting the Oracle Warehouse Builder Service	18-4
Stopping the OWB Service.....	18-4
Starting the OWB Service.....	18-4
Stopping, Starting, and Enabling the Job Queue	18-5
Starting the Job Queue.....	18-5
Stopping the Job Queue.....	18-5
Enabling the Job Queue.....	18-5
Starting and Stopping Message-Triggered Submission Queues	18-6
Starting a Message-Triggered Submission Queue.....	18-6
Stopping a Message-Triggered Submission Queue.....	18-6

19 Troubleshooting

Job Execution Problems	19-1
Job Remains at Submitted or Executing.....	19-1
Oracle Warehouse Builder Processes Do Not Start.....	19-2
Job Runs But Does Not Upload Outputs.....	19-2
Job Log File Does Not Contain Info after a Long or Failed Execution.....	19-3
Job Fails to Obtain Service.....	19-3
Report Set Jobs Do Not Produce Any Output when using SAS 9.2.....	19-3
Database Core Dumps during Object Creation	19-3
Query Exceeds 200 Rows	19-3
Table Auditing (Journaling) Not Working	19-3

Enabling a Trace on a Job	19-4
---------------------------------	------

A Object Ownership

Domains	A-1
Application Areas.....	A-2
Work Areas	A-4

Glossary

Index

Preface

This guide contains instructions for setting up the systems described in the *Oracle Life Sciences Data Hub Implementation Guide* and for performing system administration tasks.

Finding More Information

Oracle Help Center

The latest user documentation for Oracle Life Sciences Data Hub is available at <http://docs.oracle.com/en/industries/health-sciences/lsh-248>.

My Oracle Support

The latest release notes, patches and white papers are on My Oracle Support (MOS) at <https://support.oracle.com>. For help with using MOS, see https://docs.oracle.com/cd/E74665_01/MOSHPTOC/toc.htm.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Setting Up Services

This section includes the following topics:

- [About Services](#) on page 1-1
- [About the Distributed Processing Server](#) on page 1-1
- [Setting Up the Distributed Processing Server](#) on page 1-3
- [Defining Service Locations](#) on page 1-3
- [Defining Services](#) on page 1-4
- [Setting Up the Distributed Processing Server](#) on page 1-3

The LSH System Admin role is required to define services and service locations.

About Services

Oracle Life Sciences Data Hub (Oracle LSH) includes adapters that allow users to define and run objects in external systems for manipulating and reporting Oracle LSH data.

In order to integrate Oracle LSH with these external systems in your company, you must define a *service location* for each computer where an external system you are using is installed. Within the service location definition you must define a *service* for each service type required by each external system installed there.

Some external systems have more than one predefined service type; for example, one service type to integrate the system for use as an integrated development environment (IDE) and another service type to integrate the external system for use as a processing engine. See "[Service Types](#)" on page 1-8 for further information.

For each service, you must define one or more *service instances*. The number of service instances required, and their exact function, depends on the external technology and the service type; see "[Service Instances](#)" on page 1-9 for further information.

About the Distributed Processing Server

Oracle LSH uses its Distributed Processing Server (DP Server) to pass jobs to processing engines of different technologies to execute user-defined Programs, Load Sets, and Data Marts. The DP Server also passes jobs to XML Publisher to produce predefined reports and to handle Report Set post-processing.

The DP Server is designed to make Oracle LSH's execution platform engine scalable. You can install one or more instances of the DP Server on one or more computers. A single DP Server instance can handle jobs for one or more processing engines. You can

install the DP Server on the same computer as the Oracle LSH database or application server or on another network node.

The DP Server requires access to XML Publisher for the purpose of producing coversheets for outputs and for merging outputs into single- or multi-volume PDF report sets. The DP Server also requires access to each processing engine that you install to use with Oracle LSH.

Conversely, each processing engine you install for use with Oracle LSH requires access to its DP Server instance.

You must also define one **service location** in the Oracle LSH user interface for each instance of the DP Server (usually one instance per computer) and define one or more **services** for each processing engine on that computer. For each service, you specify the number of **service instances** that can run simultaneously. A service instance handles one job at a time.

Oracle LSH tracks the availability of each service on each location and assigns new jobs according to their priority and the availability of appropriate service instances in different locations; see "[Job Allocation Algorithm](#)" on page 1-2.

The DP Server runs as a standalone Java program. It includes two dedicated threads:

- **Listener.** At each location, the listener listens on a predefined Advanced Queue message queue called the DP Server job queue for jobs that have been assigned to local services. When the listener detects an XML job message request for a local service on the job queue, it dequeues the message. The message contains all the necessary details of one or more jobs.
- **Job Processor.** The job processor thread uses the information in the XML message to connect back to the Oracle LSH database. Once connected, it downloads all the necessary data, including the executable program script, into a target directory identified in the XML message. The job processor then executes the job-specific script by spawning a new O/S-level process. This spawned process runs the processing engine, which in turn carries out the actual job execution using the downloaded job input data. After the spawned external process completes, the job processor connects back to Oracle LSH and uploads any output results into Oracle LSH.

Log Files The log directory in your DP Server Home directory holds DP Server log files. Each time you start the DP Server it creates one log file. The DP Server adds log information to that log file each time it runs a job.

Job Allocation Algorithm If more than one service of the same technology type and priority exists, including in different locations, Oracle LSH allocates each job using the following algorithm:

- Oracle LSH allocates the job to the service of the correct type and priority that has been least recently used, if the least-recently-used location has an available service instance.
- If there are no available service instances, Oracle LSH checks jobs currently using the appropriate service instances on the least-recently-used service and determines whether the jobs are active. If a job is not active, Oracle LSH sets its status to Failed and allocates the new job to the service instance. If all the jobs are active, Oracle LSH checks the next-least-recently-used service, and so on until all services of the correct technology type and priority have been checked. If all services are unavailable, Oracle LSH waits for a configurable interval (see "[Reuse Wait Time](#)" on page 1-5) and then checks job activity again.

Oracle LSH uses the following algorithm to determine which available service instance to use if a service instance of the priority requested for a particular job is not available:

- **If a high-priority service is requested** but not available, run a normal-priority service; if a normal-priority service is not available, run a low-priority service.
- **If a normal-priority service is requested** but not available, run a low-priority service; if a low-priority service is not available, run a high-priority service.
- **If a low-priority service is requested** but not available, run a normal-priority service; if a normal-priority service is not available, run a high-priority service.

Setting Up the Distributed Processing Server

You must install the Oracle LSH Distributed Processing (DP) Server on each computer where you have installed an external processing engine and also on the computer where you have installed XML Publisher. The following steps are required:

1. Installing the Distributed Processing Server. See "Install the Distributed Processing Server" in the *Oracle Life Sciences Data Hub Installation Guide*.
2. Copying files. See "Copy and Edit Files" in the *Oracle Life Sciences Data Hub Installation Guide*.
3. Defining Service Locations and Services; see "[Defining Service Locations](#)" on page 1-3 and "[Defining Services](#)" on page 1-4
4. Editing the files. See "Copy and Edit Files" in the *Oracle Life Sciences Data Hub Installation Guide*.
5. Starting the DP Server; see "[Starting the Distributed Processing Server Service](#)" on page 18-2.

When you start the DP Server code you must specify a location for DP server log files. You may need to read and analyze these log files if problems occur with jobs run on external servers. Instructions for creating appropriate directories for this purpose are included in the "Install the Distributed Processing Server" section of the installing Oracle LSH chapter of *Oracle Life Sciences Data Hub Installation Guide*.

Defining Service Locations

A service location is a computer where one or more external processing engines or IDEs used by Oracle LSH are installed. You must register each service location by defining it.

Note: Before defining a service location or services for use with Oracle Business Intelligence visualizations, read [Chapter 11, "Setting Up Oracle Business Intelligence Visualizations"](#).

Note: The Oracle LSH installation process automatically creates one service location with one normal-priority PL/SQL service with three instances allowed, for internal use. This enables batch processing and many functions in Oracle LSH Definition.

To define a Service Location:

1. In the Administration tab, click the **Service Locations** subtab.

2. Click **Create Service Location**. The Service Location screen opens.
3. Enter values in the following fields:
 - **Location Name**. Enter a name for your service location.

Note: For processing engine services including XML Publisher and SAS, this name must match (including case-sensitivity) the service location name defined in the DP Server Start script that you run on the computer that corresponds to this defined service location; see ["Starting the Distributed Processing Server Service"](#) on page 18-2. When you run the script, you pass the actual values required to identify the physical location.

Note: For OBIEE Service Locations where the BI Server is not installed, the name is very important. See [Chapter 11, "Setting Up Oracle Business Intelligence Visualizations"](#) for details on OBIEE services.

- **Description**. Enter additional information about the service location.
 - **Machine**. Enter the name of the computer that contains the location. This is for your own records; it has no effect.
4. Click **Apply**. The system creates the Service Location definition and returns you to the View Service Locations screen.

Note: In Oracle LSH Release 2.4.8 the **DP Server Status** field always displays the status "Running."

Defining Services

This section includes the following topics:

- [Service Types](#) on page 1-8
- [Service Prioritization](#) on page 1-9
- [Service Instances](#) on page 1-9

For each service location, define services as follows:

- One or more services for each XML Publisher, SAS, Informatica, Oracle Reports, Oracle Business Intelligence Presentation Services instance, and Oracle Business Intelligence Publisher (BIP) engine installed on that computer
- One and only one service for PL/SQL as a processing engine
- One and only one service for each IDE service type

For non-PL/SQL processing engine service types, you may want to define more than one service in order to allow jobs with different priorities to run on the same computer. For example, if you have both SAS and Oracle Reports installed on the same computer, you might define the following services for that service location:

- One high-priority SAS service with 2 instances allowed

- One normal-priority SAS service with 7 instances allowed
- One normal-priority Oracle Reports service with 5 instances allowed

To define a Service on a Service Location:

1. In the Administration tab, click the **Service Locations** subtab. The Service Locations screen opens.
2. Click **Service Location**. The Service Location screen opens.
3. Click the **Location Name** of the Service Location where you want to create a Service. The Maintain Service Locations screen opens.
4. Click **Create Service**. The Services screen opens.
5. Enter values in the following fields:
 - **Service Type**. Select a value from the drop-down list. For a description of each type, see "[Service Types](#)" on page 1-8.
 - **Description**. Enter a description of the service.
 - **Service Priority**. Select a value from the drop-down list. You must define each service as **High**, **Normal**, or **Low** priority. See "[Service Prioritization](#)" on page 1-9 for further information. IDE service types do not use this attribute.
 - **Number of Instances**. Enter the number of jobs you want to allow to use this service at the same time. See "[Service Instances](#)" on page 1-9 for further information.
 - **Launch Wait Time**. Enter the length of time (in seconds) for the program to start executing after the message to start the program is sent to the DP Server Listener. Set to a value long enough that even a busy service will have time to receive and process the message—but not too long or error recovery is adversely affected. 30 - 120 seconds is recommended.
 - **Reuse Wait Time**. Enter the length of time (in minutes) that you want the system to wait after the start of a program's execution until it starts checking whether a program is still executing when the system is looking for available service instances. Set to an interval large enough to allow most programs of the service's technology type to complete.
 - **Default Resource Group**. Enter the name of an Oracle database resource group to set the Oracle database priority for jobs using this service. For more information, see the *Oracle® Database Administrator's Guide* at http://download.oracle.com/docs/cd/E11882_01/server.112/e17120.pdf.
 - **Execution Command**. Enter one of the following:
 - For services used to launch an IDE, enter the URL of the development or visualization environment.
 - For services used to execute a job, enter the complete directory path and filename for the UNIX or Windows execution command script required for the service type. The service instance launches this script in order to run a job. The script must be located in the DP Server Home directory or a subdirectory of it. Instructions for copying and editing the sample execution command scripts, which are located in `$CDR_TOP/admin/template`, are included in the *Oracle Life Sciences Data Hub Installation Guide's* "Install the Distributed Processing Server" section.

You can, if you want to, customize the script to change the OS execution priority of the program to correspond to the service's priority; see "[Service](#)

[Prioritization](#)" on page 1-9.

The sample scripts include:

- **cdrizip-1.sh** for Text Data Marts
- **sasNormal.sh** for SAS Programs
- **oraexp.sh** for Oracle Export Data Marts
- **orareprunner.sh** for Oracle Reports Programs
- **sqlplus** for PL/SQL Programs
- **txtNormal** for Text Load Sets
- **xmlprunner.sh** for post-processing Report Sets
- **xmlpreprunner.sh** for the Oracle LSH system reports and for coversheets for outputs
- **obieinstall.cmd** for Oracle Business Intelligence Business Areas based on OBIEE 11g, or **obieinstall10g.cmd** for Oracle Business Intelligence Business Areas based on OBIEE 10g; see [Chapter 11, "Setting Up Oracle Business Intelligence Visualizations"](#).
- **obieedeploy.cmd** for Oracle Business Intelligence Business Areas based on OBIEE 11g, or **obieedeploy10g.cmd** for Oracle Business Intelligence Business Areas based on OBIEE 10g; see [Chapter 11, "Setting Up Oracle Business Intelligence Visualizations"](#).
- **obieepsrestart10g.cmd** or **obieepsrestart10g.sh** for Oracle Business Intelligence Business Areas based on OBIEE 10g that are using a multiple-computer configuration; see [Chapter 11, "Setting Up Oracle Business Intelligence Visualizations"](#).
- **infarunner.sh** for Informatica Programs
- **cdrunzip.sh** for creating Tables from a zipped file and Text Load sets
- **Root Directory.** Enter the directory to be used as the ROOT of the directory structure where jobs will execute. The DP Server creates a subdirectory in the root directory for each job, named with the job ID; for example:

```
root dir/3233210001
```

where the job ID is 3233210001. This directory will hold all the files and/or additional directories needed for the job to run.

- **Enabled.** If set to **Yes**, this service is available for use.
- **Grant Connect.** If set to **Yes**, the service can connect to the database to retrieve and manipulate data. If the service does not need to connect to the database, set to **No**.

Note: Set Grant Connect to **No** for the Workflow service.

- **Grant Resource.** If set to **Yes**, the execution accounts can create tables in the Service Instance account.

Note: Set Grant Resource to **No** for the Workflow service and to **Yes** for the Oracle Export service.

- **Set Synonym.** If set to **Yes**, synonyms are created when the user connects to the database to run the job or the development environment.

Note: Set the value of Set Synonym to **No** for the Workflow and Oracle Export services.

- **Details.** This field takes different values for different adapters. Not all adapters use this field but it is mandatory for the adapters that do. The following adapters use it as follows:
 - **Informatica.** Enter the Informatica domain name, repository name, and integration service name, separated by a vertical bar; for example: Domain_infa444dom|PowerCenter_abcd|PowerCenter_abcd_Integration_Service.
 - **OBIEE Business Area Install Service.** If the service location is a UNIX computer, enter: os=unix
See [Chapter 11, "Setting Up Oracle Business Intelligence Visualizations"](#) for more information.
 - **OBIEE Business Area Deploy Service.** Enter the values described below, delimited by commas:

DSN. Enter the DSN name if you are using an ODBC system DSN, or enter the TNS name of the source Oracle LSH database server name if you are using OCI10g.

Call Interface. Enter ODBC if you are using an ODBC system DSN, or OCI10g if you are using OCI10g.

RPD. Enter the name of the OBI repository file you defined for this computer in the NQSConfig.ini file on the BI Server. The name must be an exact match of the value there, except the value here should not include the .rpd extension. Oracle recommends using the computer name to help ensure consistency.

Version. If you are using OBIEE 11g, add the string ver=11g at the end.

Examples

If you are using an ODBC DSN with OBIEE 11g:

```
dsn=name_of_DSN,callinterface=ODBC,rpd=name_of_the_OBI_Repository_file_to_be_deployed_on_this_Oracle_BI_Presentation_Server,ver=11g
```

If you are using OCI10g and OBIEE 10g:

```
dsn=tns_name_of_source_LSH_database_server,callinterface=OCI10G,rpd=name_of_the_OBI_Repository_file_to_be_deployed_on_this_Oracle_BI_Presentation_Server
```

- **SAS with SQL*Loader.** If you are running the SAS and the SQL*Loader services on the same DP server computer for the purpose of bulk loading

SAS data sets, enter the absolute path of the SQL*Loader executable in this field for the SAS service.

Service Types

Some service types use the DP Server and others do not. Their definition requirements are different:

- [Service Types that Do Not Use the DP Server](#) on page 1-8
- [Service Types that Use the DP Server](#) on page 1-8

Service Types that Do Not Use the DP Server

Development (IDE) service types do not use the DP Server. The PL/SQL processing engine also does not use the DP Server. For each of the technologies you plan to use, you must define a single service of the corresponding service type. You cannot control where these services are used. The system does not use the service location you define to determine where to run them. However, you must define them within a service location. You can define them on the service location where they are actually installed, or you may want to define a service location specifically for the purpose of defining these services, which include:

- **Oracle Reports for Development.** Oracle LSH uses these services to support the Oracle Reports integrated development environment (IDE).
- **PLSQL for Development.** Oracle LSH uses this service to support the SQL*Plus integrated development environment (IDE) for PL/SQL Programs.
- **SAS for Development.** Oracle LSH uses this service to support the SAS integrated development environment (IDE).
- **PLSQL.** Oracle LSH uses PLSQL services to run internal jobs and user-defined PL/SQL Programs and Text Data Marts.
- **OBIEE Business Area IDE Service.** This is the service that takes the OBIEE Presentation Server's URL as input for the Execution Command field. Oracle LSH uses additional OBIEE service types to support running visualizations through the OBIEE Presentation Server; see [Chapter 11, "Setting Up Oracle Business Intelligence Visualizations"](#). The other OBIEE Services are listed under "[Service Types that Use the DP Server](#)" on page 1-8.
- **BI Publisher Development.** Oracle LSH uses this service to support the BI Publisher integrated development environment (IDE).
- **Informatica for Development.** Oracle LSH uses this service to support the Informatica integrated development environment (IDE).
- **View for Development** Oracle LSH uses this service to support the SQL*Developer integrated development environment (IDE) for View Programs.

Service Types that Use the DP Server

Processing engines integrated with Oracle LSH use the DP Server. You can control the number of service locations and services for service types that use the DP Server. Service locations correspond to physical locations, and each one can have any number of services defined. Each service can have any number of service instances defined; see "[Service Instances](#)" on page 1-9.

For each technology you plan to use, define at least one service of the following service types:

- **Oracle Export.** Oracle LSH uses Oracle Export services for creating Oracle Export-type Data Marts.
- **Oracle Reports.** Oracle LSH uses these services to execute Oracle Reports Programs.
- **SAS.** Oracle LSH uses SAS services to execute SAS Load Sets, SAS Programs, and SAS Data Marts on a SAS server.
- **Text for SQLLoader.** Oracle LSH uses Oracle SQL Loader to execute Text Load Sets.
- **Workflow.** Oracle LSH uses Workflow services to execute user-defined and internal Workflows.
- **XML Publisher.** Oracle LSH uses XML Publisher services to generate PDF-format Report Sets.
- **XML Reports.** Oracle LSH uses these services to produce predefined reports and the coversheets for reports and report sets.
- **Zip Utility.** Oracle LSH uses Zip Utility services to produce zipped Text and SAS Data Marts. You can zip the text files if you want to.
- **Unzip Utility.** Oracle LSH uses Unzip Utility services to unzip uploaded zip files while creating Tables and Text Load Sets.
- **OBIEE Business Area Install Service.** This service is required to support running visualizations through the OBIEE Presentation Server, and using the OBIEE Admin Tool; see [Chapter 11, "Setting Up Oracle Business Intelligence Visualizations"](#)
- **OBIEE Business Area Deploy Service.** This service is required to support running visualizations through the OBIEE Presentation Server; see [Chapter 11, "Setting Up Oracle Business Intelligence Visualizations"](#).
- **BI Publisher.** Oracle LSH uses this service to execute BI Publisher Programs.
- **Informatica.** Oracle LSH uses this service to execute Informatica Programs.

Note: The following service types are inactive: Oracle Warehouse Builder and SQL*Plus. Do not define services of these types.

Service Prioritization

You must specify a priority for each service: High, Normal, or Low. However, to ensure that services are actually given the appropriate priority, you should also do the following:

- Specify an **Oracle Database Resource Group** to set the Oracle database priority for jobs using this service. For more information, see the *Oracle® Database Administrator's Guide* at http://download.oracle.com/docs/cd/E11882_01/server.112/e17120.pdf.
- In the **Execution Command**, set the operating system priority for jobs using this service.

Service Instances

When you create a service, you must specify how many instances of the service are allowed to run at the same time. Since each instance can service one job (or Program or

Business Area definition) at a time, this effectively defines the number of jobs that can run on a particular service at a particular time.

For processing engine service types, service instances serve as the interface between the requested job and the Oracle LSH execution system. They track the association of a particular job to a particular service and manage many aspects of the job's execution.

IDE service instances control the association between the use of a particular IDE for a particular Program or Business Area (which are used for visualizations) and the security and data access needed by the particular IDE. When a user launches an IDE from the Program or Business Area user interface, the system assigns a service instance to the Program or Business Area. After a service instance is allocated, it remains assigned to that Program or Business Area until it is explicitly released by a Checkin or Undo Checkout.

You can modify the number of service instances for each service .

Note: For the OBIEE IDE Service, specify a number of Service Instances equal to the maximum number of users you expect to query the repository at the same time, plus a buffer.

If you do not create enough service instances, users may lose access to the database in the middle of a task. For example, if you define 10 service instances and 11 people log in to the same OBIEE URL, when the last person logs in, the system looks for a session that has been running longer than the defined Reuse Wait time for the service. If there is such a session, the system recycles it even if the user is active, and the original user loses his or her connection to the database.

It is better to solve this problem by defining a large number of service instances than a high reuse wait time because the same user cannot open a new session—for example, if he or she wants to select different launch settings—until his or her original session reaches the reuse wait time.

Setting Up Adapters to External Systems

This section contains the following topics:

- [About Adapters](#) on page 2-1
- [Adapter Structure](#) on page 2-2
- [Creating Your Own Adapters](#) on page 2-4
- [Setting Up Security for Adapters](#) on page 2-4
- [Clearing Oracle Applications Cache](#) on page 2-6

About Adapters

The Oracle Life Sciences Data Hub (Oracle LSH) includes a set of predefined adapters that handle the interface between Oracle LSH and external systems. Each adapter is customized for a particular system, containing the particular functions that are required internally by Oracle LSH to safely control the exchange of data and metadata with the external system.

When you upgrade to a new version of either Oracle LSH or one of these external systems, you may need to upgrade the adapter, but you do not need to upgrade the other system.

Oracle LSH also includes a set of adapters used for internal purposes.

Oracle LSH includes adapters to different types of external systems: [Programming Languages](#), [Source Data Systems](#), [Visualization Tools](#), and [Data Export Tools](#). You can create custom adapters to allow you to integrate additional systems such as third party visualization tools with Oracle LSH; see "[Creating Your Own Adapters](#)" on page 2-4.

Programming Languages Oracle LSH is integrated with adapters for programming languages to enable data transformation. A defined object called Program is also required. Oracle LSH Programs typically read data from source tables and write data into target tables. The source and target tables reside in Oracle LSH. Oracle LSH Programs can also work without interacting with tables, if the requirement is such.

When a Definer creates an Oracle LSH Program, the only Program types available to him or her are those that correspond to the particular adapters to which the Definer has security access.

Source Data Systems To load data into Oracle LSH from an external system, you must use an adapter. A defined object called a Load Set is also required.

When a Definer creates a Load Set, the only Load Set types available to him or her are those that correspond to the particular adapters to which the Definer has security

access. The system uses Parameters and Programs predefined as part of the adapter to create the Load Set. For all source data adapters except Text, Oracle LSH pulls table or data set metadata from the external system to create target Tables in Oracle LSH.

When a user runs the Load Set, the adapter handles the interaction between the two systems, including the actual loading of data.

Visualization Tools Oracle LSH also uses adapters to allow external systems access to Oracle LSH data for creating data visualizations and reports.

The defined object required for allowing data access to visualization tools is the Business Area, which specifies which data will be available to a visualization. When a Definer creates a Business Area, the only Business Area types available are those that correspond to the particular adapters to which the Definer has security access. It is possible for customers to create adapters to third-party visualization and reporting tools.

Data Export Tools To transport Oracle LSH data to a different system, you create a defined object called a Data Mart. There are three types of Data Marts: Text, Oracle Export, and SAS Data Mart.

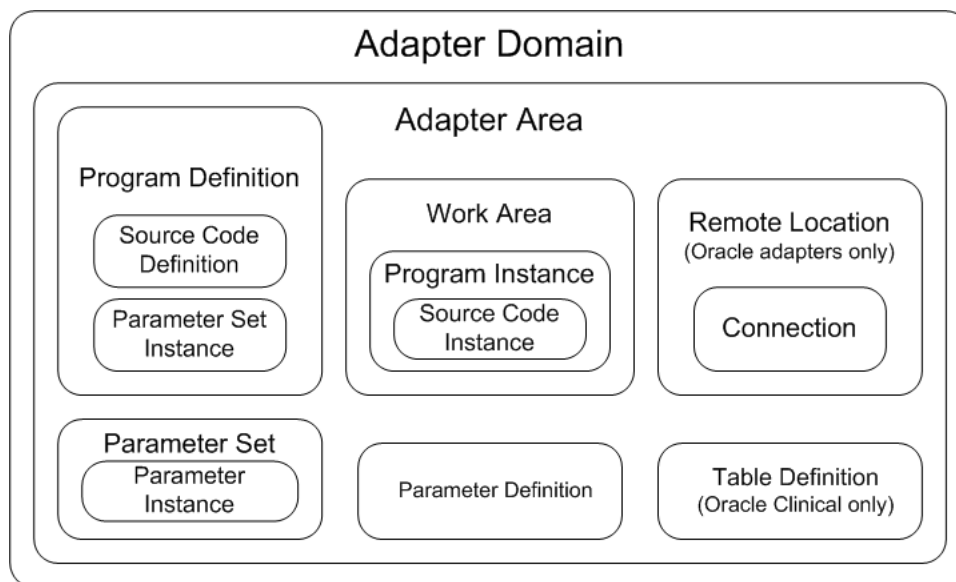
Adapter Structure

Oracle LSH adapters are defined objects like other defined objects in Oracle LSH.

An adapter consists of an Adapter Domain with one or more Adapter Areas, each containing a Work Area. The Adapter Area also contains two Program definitions with source code for each custom function required by the external system, and the Parameter Set required for each Program. The Work Area contains an instance of each Program. [Figure 2-1](#) shows object ownership within an Adapter Domain.

Note: The Work Area, like other Oracle LSH Work Areas, must be installed in the database in order to function. Adapter Work Areas are installed as part of the post-installation job; see "Run the Post-Installation Programs" in the *Oracle Life Sciences Data Hub Installation Guide*.

Figure 2–1 Adapter Structure: Object Ownership



Oracle LSH Release 2.4.8 includes the following Adapter Domains (the names in parentheses appear in the Oracle LSH user interface):

- **Oracle Export (ORACLE_EXPORT)**. The Oracle Export adapter is required to create Oracle Export Data Marts.
- **Oracle Clinical (ORACLECLINICAL_ADAPTER)**. The Oracle Clinical Adapter Domain includes many different Adapter Areas, one for each type of Oracle Clinical Load Set. You can assign different user groups to each one if you want to.
- **Oracle Family (ORACLEFAMILY_ADAPTER)**. The Oracle adapter is required to create Load Sets of type Oracle Tables and Views. This adapter is also required to create Programs of type PLSQL and Oracle Reports.
- **SAS Family (SASFAMILY_ADAPTER)**. The SAS adapter is required to create SAS Load Sets.
The SAS adapter is also required to create Programs of type SAS, SAS Macro Catalog, and SAS Format Catalog.
- **SAS Export (SAS_EXPORT)**. The SAS Export adapter is required to create SAS Data Marts.
- **Text Export (TEXT_EXPORT)**. The Text Export adapter is required to create Text Data Marts.
- **Text Family (TEXTFAMILY_ADAPTER)**. The Text adapter is required to create a Text Load Set. It uses the defined Load Set Parameter values to create a control file for the SQL*Loader.
- **OBIEE Adapter (OBIEE)**. The Visualization adapter is required to create Oracle Business Intelligence (OBIEE) visualizations of Oracle LSH data.
- **BIP Adapter (BIP_ADAPTER_DOMAIN)**. The BIP adapter is required to create Oracle LSH Business Intelligence Publisher (BIP) Programs.
- **Informatica (INFORMATICA_ADAPTER_DOMAIN)**. The Informatica adapter is required to create Oracle LSH Informatica Programs.

- **Generic Visualization Adapter (GENERIC_VISUALIZATION)**. The generic visualization adapter is required to create visualizations of LSH Data in tools other than OBIEE.
- **InForm (INFORMFAMILY_ADAPTER)** This is required for use with Oracle Health Sciences Data Management Workbench to enable loading data and metadata from InForm into Oracle DMW.

Oracle-technology source data adapters (Oracle and Oracle Clinical) contain the Remote Location and Connection information you define for the external Oracle system; see [Chapter 5, "Registering Locations and Connections"](#).

Creating Your Own Adapters

You can create adapters to other external systems you want to use with Oracle LSH; for example, third party visualization and reporting tools.

You must create the same definitional object structure that the predefined adapters use (see ["Adapter Structure"](#) on page 2-2), and install the adapter Work Areas to the database. There is no user interface in Oracle LSH for creating adapter; you must use the Application Programming Interface (API). Instructions are available in a White Paper posted on My Oracle Support.

After you have created an adapter it appears in the Oracle LSH user interface and you can assign user groups to it the same way you do for the predefined adapters; see ["Setting Up Security for Adapters"](#) on page 2-4.

Setting Up Security for Adapters

This section contains the following topics:

- [Assigning User Groups to Adapter Domains and Adapter Areas](#) on page 2-4
- [Creating Roles with the Required Operations for Adapters](#) on page 2-5

You set up security for adapters the same way you do for other definitional objects: assign user groups to them, and assign users to roles within the groups; see ["Setting Up Object Security"](#) on page 9-9 for information on creating user groups and roles.

The LSH Adapter Security Admin role is required to see the Adapter Security subtab of the Security tab and to perform the tasks described in this section.

Assigning User Groups to Adapter Domains and Adapter Areas

This section contains the following topics:

- [Assigning User Groups](#) on page 2-5
- [Viewing and Modifying User Group Assignments](#) on page 2-5

As with other defined objects, objects contained in an Adapter Domain inherit the user group assignments of their containing object by default.

Some predefined adapters have only one Adapter Area in the Adapter Domain, and you need only assign a user group at the Domain level.

However, where an Adapter Domain contains multiple Adapter Areas, you can assign security to each Adapter Area separately; for example, if you want different users to be able to run different types of Oracle Clinical Load Sets, you can assign different user groups to the corresponding Adapter Areas.

SAS Family Adapter, Visualization Adapter, Oracle Family Adapter, and Oracle Clinical adapter Domains have more than one Adapter Areas.

Assigning User Groups

To assign a user group to an adapter, do the following:

1. Navigate to the Security tab and then the Adapter Security subtab.
2. Click the padlock icon in the Apply Security column for the adapter to which you want to assign a user group. The Manage Security screen opens.
3. Click **Assign Group**. The Search User Group screen opens.
4. In the **Group Name** field, enter the name of the user group you want to assign, or enter part of the name and use the % wildcard.
5. Click **Search**. The system displays the search results.
6. Select one or more groups and click **Apply** to assign those groups to the Adapter Domain. The system displays the assignments with a confirmation message.

Viewing and Modifying User Group Assignments

To see the user groups currently assigned to an adapter, do the following:

1. Navigate to the Security tab and then the Adapter Security subtab.
2. Click the padlock icon in the Apply Security column for the adapter to which you want to assign a user group. The Manage Security screen opens, displaying the current user group assignments.

You can change user group assignments as follows:

- To assign another user group to the adapter, click **Assign Group** and follow Steps 4 through 6 above.
- To unassign a user group that was explicitly assigned to the Adapter Domain or Adapter Area, click the icon in its **Unassign** column.
- To unassign a user group that was assigned to an Adapter Area by inheritance, click the icon in the **Revoke** column.
- To reassign a user group that was assigned to an Adapter Area by inheritance and then revoked, click the icon in the **Un-Revoke** column.

Creating Roles with the Required Operations for Adapters

To define an Oracle LSH object of a particular type, a user must belong to a user group assigned to the corresponding Adapter Area (either explicitly or by inheritance) and must be assigned a role within the user group that has View privileges on the Adapter Area and on all the objects it contains.

Create roles and add operations to roles as usual in Oracle LSH; see "[Creating Object Security Roles](#)" on page 9-11.

The View operation on the following objects (of subtype Default) is required:

- Adapter Domain
- Adapter Area
- Work Area
- Program

- Program Instance
- Parameter

Note: Do not include Modify privileges on Parameters. If you include Modify privileges, users with that role will be able to modify adapter Parameter definitions through Parameter instances contained in Load Sets, Data Marts, and Business Areas, because they reference the Parameter definition in the Adapter Area.

Modifying Parameter definitions in the Adapter Area will increment the version number of the adapter and all Load Sets, Data Marts, and Business Areas of that adapter type will no longer work.

- Parameter Instance
- Parameter Set
- Parameter Set Instance
- Source Code
- Source Code Instance
- Table (required for Oracle Clinical adapters only)

Note: Do not include Modify privileges on Tables. If you include Modify privileges, users with that role will be able to modify adapter Table definitions through Table Descriptors of Oracle Clinical Load Sets, or through Table instances created from those Table Descriptors, because both reference the Table definition in the Adapter Area.

Modifying Table definitions in the Adapter Area will increment the version number of the adapter and all Load Sets of that type will no longer work.

- Remote Location (required for any Oracle-technology adapters)
- Connection (required for any Oracle-technology adapter)

In addition, for Oracle-technology adapters include all operations on Connections.

Clearing Oracle Applications Cache

Every time you assign a user to a user group, clear the Oracle Applications cache so that users logging into Oracle LSH do not see unassigned privileges. Note that by not clearing the cache, users can only see unassigned privileges but not use them. However, to avoid any confusion, it is recommended to clear the cache.

Do the following:

1. Log in as an Oracle LSH user with the Functional Administrator responsibility. See ["Assigning Application Roles"](#) on page 9-5 for instructions on granting this responsibility.

The Security tab's Grants screen appears.

Note: You can also log in as the `sysadmin` user if you do not want to assign the Functional Administrator responsibility only for the purpose of clearing cache.

2. Click the **Core Services** tab from the Grants screen. The Lookup Types screen appears.
3. Click **Caching Framework** from the list of secondary tabs. The Overview screen opens.
4. Click **Global Configuration** from the left panel. The Global Cache Configuration screen opens.
5. Click **Clear All Cache**. A warning appears.
Click **Yes**.

Setting Profile Values

This section contains the following topics:

- [How to Log In to Oracle Applications Profile Forms](#) on page 3-1
- [Setting Native Client Encoding](#) on page 3-2
- [Using Character Semantics for Work Area Table Installation](#) on page 3-2
- [Gathering Work Area Schema Statistics](#) on page 3-3
- [Creating a New Tablespace for Work Area Schema Objects](#) on page 3-4
- [Setting the Default Value for: Force Output Validation Status to Development](#) on page 3-5
- [Setting the Maximum Number of Nested Domains](#) on page 3-5
- [Adding the Direct SQL Option to SAS Libraries](#) on page 3-6
- [Allowing User Preferences for SAS Clients](#) on page 3-6
-
- [Set Processing Type for OC Data Extract Incremental Adapter](#) on page 3-8
- [Turn debugging on or off](#) on page 3-8
- [Setting Password Requirements](#) on page 3-8
- [Oracle LSH: BIP Endpoint](#) on page 3-11
- [Attachment File Upload Restriction Default](#) on page 3-12

Oracle Applications includes a set of profiles whose settings affect the way your application looks and behaves. You must have the System Administrator responsibility to set profile values at the site level.

For further information, see the *Oracle E-Business Suite System Administrator's Guide - Maintenance* at http://download.oracle.com/docs/cd/B53825_08/current/acrobat/121samg.pdf.

How to Log In to Oracle Applications Profile Forms

1. Open your Oracle LSH URL.
2. Log on with the system administrator account. An E-Business Suite screen opens.
3. In the Main Menu pane, expand the **System Administrator** (not System Administration) node, then the **Profile** node, and then click **System**.

Note: This user interface requires Java 6 Update 7 (1.6.0.0.7). If you do not have it installed, you can download it from <http://www.oracle.com/technetwork/java/javase/archive-139210.html>. To use this feature, you must accept all warnings.

A new browser screen opens with several windows open and the Find System Profile Values window on top.

Tip: If you lose the Find System Profiles window at any point, click the Search icon in the toolbar or click System Profiles in the Top Ten list.

Setting Native Client Encoding

The value for this profile determines the encoding used when uploading and downloading files from the client. It must be set to UTF-8. The system administrator can set it at the site level. Individual users (or the system administrator) can set it at the user level in the Oracle LSH User Preferences user interface. The user setting overrides the site setting.

The shipped default value is WE8MSWIN1252.

To change the default value, do the following:

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter `FND: Native Client Encoding` or some part of that string followed by the wild card: `%`
3. Click **Find**. The System Profiles pop-up window appears with the profile displayed.
4. Click in the **Site** column. An ellipsis (...) appears on the right-hand side of the field.
5. Click the ellipsis (...). A pop-up box appears.
6. Click **UTF-8** and then **OK**.
7. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
8. Click **OK**. The transaction message pop-up disappears.
9. Close the window.

Using Character Semantics for Work Area Table Installation

If your data includes multibyte unicode characters, your data may sometimes not fit into character columns in the Oracle LSH Table instances you create because you are using byte semantics (which is the default behavior). You can set this profile to Yes to use character semantics instead of byte semantics across your Oracle LSH installation.

Using byte semantics, a character column's length is the same as the number of bytes the column can contain. For example, a character column with a length of 2 can contain either two 1-byte characters or one 2-byte character, but it cannot contain one 2-byte character and one 1-byte character together. If you try to write more bytes of

data into a column than it can hold, you get an error like `ORA-12899: value too large for column ??? (actual: 3, maximum: 2)`.

Using character semantics, the column length is the same as the number of characters—including multibyte characters—the column can contain. For example, a character column with a length of 2 can store two 1-byte characters, two 2-byte characters, or one of each. Oracle Applications always uses byte semantics, but Oracle Warehouse Builder supports character semantics as well.

Note: The maximum size of Oracle character columns is 4000 bytes regardless of whether you use byte or character semantics. A character column with a length of 4000 can store 4000 bytes—for example, 4000 1-byte characters or 2000 2-byte characters—but the total number of bytes cannot exceed 4000 using either byte or character semantics.

Note: The type of semantics in effect at the time a Table instance is first installed or reinstalled during a Full Work Area installation determines the type of semantics the Table instance uses.

A Table instance first installed using byte semantics (when Use Character Semantics is set to No) continues to use byte semantics even if you change the profile value to Yes and even if you reinstall the Work Area using an Upgrade installation. To apply character semantics to the Table instance you must do an installation of type Full. **Full installation deletes the data in the Table instance.**

Similarly, if you first install a Table instance when Use Character Semantics is set to Yes, it continues to use character semantics even if you set the profile value back to No—until you do a Full installation.

To use character semantics, do the following:

1. Log in. See "[How to Log In to Oracle Applications Profile Forms](#)" on page 3-1.
2. In the **Profile** field, enter `LSH: Use Character Semantics for Workarea Installation`
3. Click **Find**.
4. In the **Site** column:
 - Select **YES** to use character semantics.
 - Select **NO** to use byte semantics (the default value).
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Click the **X** in the upper right corner of the System Profile Values window to close the window.

Gathering Work Area Schema Statistics

You can set a profile to gather statistics on user-defined tables in a Work Area during the post-processing phase of job executions writing to tables. By default, Oracle LSH does not gather table statistics during job execution.

If you set the profile value to **YES**, the system gathers statistics on any user-defined table that has no statistics or stale statistics.

A table is considered to have stale statistics when it has been modified by more than 10% since statistics were last gathered. For example, if a table has 100 records and a job modifies (inserts, updates, or deletes) a total of 8 records, the system does not gather statistics during post-processing of the job. However, if the next job modifies 3 rows in the table, the cumulative number of rows modified since the last time statistics were gathered is 11, which is more than 10% of the original number of rows, and the system gathers the statistics during the post-processing phase of the second job.

To set this value, do the following:

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter `LSH: Gather Work Area Schema Statistics`
3. Click **Find**.
4. In the **Site** column:
 - Select **YES** to gather statistics as described above
 - Select **NO** if you do not want to gather statistics as described above
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Click the **X** in the upper right corner of the System Profile Values window to close the window.

Creating a New Tablespace for Work Area Schema Objects

By default, Oracle LSH internal application tables and user-defined Work Area tables are all contained in the same 8K block size tablespace. As data grows or when there are huge data loads in progress, there is cache contention between the DMLs coming from LSH user interface and the DMLs from backend jobs, and user interface performance may be impacted.

To resolve this, create a new tablespace with a different block size and use that tablespace for all Work Area schemas and objects. Tablespaces with different block sizes do not use the same buffer cache, so the cache contention is eliminated.

For information on moving existing tables into the new tablespace, see My Oracle Support article ID 1518355.1.

To set this value, do the following:

1. Create a new tablespace with a 16K block size (or another size that is different from 8K).
2. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
3. In the **Profile** field, enter `LSH: Tablespace for Workarea schema objects`
4. Click **Find**.
5. In the **Site Level** box, enter the name of the tablespace you created for this purpose. Be sure to use uppercase letters.
6. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.

7. Click **OK**. The transaction message pop-up disappears.
8. Click the **X** in the upper right corner of the System Profile Values window to close the window.

Setting the Default Value for: Force Output Validation Status to Development

Oracle LSH Program definitions have an attribute that determines how all outputs of all instances of that Program definition receive their initial validation status. You can set the default value in a profile.

In a Program definition, if this attribute is set to **Yes**, outputs of instances of the Program definition are always created with a validation status of Development. If set to **No**, the outputs inherit the validation status of the Execution Setup that produced them, which in turn can inherit its validation status from the Program instance.

The shipped default value is **No**.

To change the default value, do the following:

1. Log in. See "[How to Log In to Oracle Applications Profile Forms](#)" on page 3-1.
2. In the **Profile** field, enter LSH: Force Output Validation Status to Development or some part of that string followed by the wild card: %
3. Click **Find**. The System Profiles pop-up window appears with the profile displayed.
4. Click in the **Site** column. An ellipsis (...) appears on the right-hand side of the field.
5. Click the ellipsis (...). A pop-up box appears.
6. Click **Yes** or **No** and then **OK**.
7. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
8. Click **OK**. The transaction message pop-up disappears.
9. Close the window.

Setting the Maximum Number of Nested Domains

In Oracle LSH you define Library Domains to contain object definitions, Application Areas and, if you allow Domain nesting, other Domains. See "Designing an Organizational Structure" in the *Oracle Life Sciences Data Hub Implementation Guide* for further information. Use the Domain Nest Value profile to determine whether Domain nesting is allowed in your implementation and, if it is, how many levels of Domain nesting are allowed:

- If the domain nest value is set to 1, no nesting of Domains is allowed. Your implementation cannot have Domains contained in other Domains.
- If the value is set to 2, which is the default, a top-level Domain can contain a child Domain, but the child Domain cannot contain a Domain.
- If the value is set to 3, child Domains can contain Domains, but those Domains cannot contain other Domains. The maximum setting is 9.

Note: The setting of the domain nest value profile does not affect the total number of Domains allowed, only the number of nested levels allowed. For example, if the value is set to 2, child Domains cannot contain Domains. However, parent Domains can contain any number of child Domains.

To change the maximum number of nested Domains allowed in your Oracle LSH implementation from the default value of 2, do the following:

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. Click in the **Profile** enter LSH: Domain Nesting Levels or some part of that string followed by the wild card: %
3. Click **Find**. The System Profiles pop-up window appears with the profile displayed.
4. In the **Site** column, enter any value between 1 and 9, inclusive.
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Close the window.

Adding the Direct SQL Option to SAS Libraries

This profile allows you to add the SAS option DIRECT_SQL=NOWHERE to your SAS libraries. This SAS option prevents SAS WHERE clauses and PROC SQL-generated or -specified WHERE clauses from being passed to the DBMS for processing.

The default value is null, which has the same effect as a setting of NO: the DIRECT_SQL=NOWHERE is not included in your SAS libraries.

To add the option to your SAS libraries, do the following:

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter LSH: Include Direct SQL option to SAS library
3. Click **Find**.
4. In the **Site** column:
 - Select YES to add the SAS option DIRECT_SQL=NOWHERE to your SAS libraries
 - Select NO if you do not want to add the option to your SAS libraries
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Click the X in the upper right corner of the System Profile Values window to close the window.

Allowing User Preferences for SAS Clients

This profile allows you to set the default User Preference for SAS client across your site and stores individual users' preferences.

Oracle LSH ships with a single value for the SAS IDE Client user preference: SAS PC, which supports PC SAS for Windows. If your users need to use a different SAS client, for example, SAS EG, you can extend the lookup [SAS IDE Client](#) to display additional clients on the Life Sciences Data Hub User Preferences screen. Users can then select the client they prefer.

Note: If you do not set a default value, users are required to set a value in the User Preferences screen in order to launch any SAS client.

To set the default SAS Integrated Development Environment (IDE) Client for your site, do the following:

1. Log in. See "[How to Log In to Oracle Applications Profile Forms](#)" on page 3-1.
2. In the **Profile** field, enter LSH: SAS IDE Client
3. Click **Find**.
4. In the **Site** column:
 - Select SAS_PC to make PC SAS for Windows the default preference across your site.
 - Select a different value—one you have entered through the [SAS IDE Client](#) lookup—to make that client the default.
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Click the X in the upper right corner of the System Profile Values window to close the window.

Set Default SAS Number Format

The default value in Oracle LSH for SAS number format is 8, which does not support precision. To change the default value to BEST, change the value of this profile to Yes.

1. Log in. See "[How to Log In to Oracle Applications Profile Forms](#)" on page 3-1.
2. In the **Profile** field, enter LSH: Default SAS Number Format BEST.
3. Click **Find**.
4. In the **Site** column:
 - Select No to make the default SAS number format value equal to 8. This is the default value.
 - Select Yes to make the default SAS number format value equal to **BEST**, which supports precision in numbers and is the best value if you load files that do not have a SAS number format specified.

Note: If you change this setting, running any existing Load Sets having target Table instances with Number columns that have the old default value of 8 will end with a warning if you load files that do not have a SAS number format specified. This is because the metadata comparison will show a difference between the source and target.

5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Click the **X** in the upper right corner of the System Profile Values window to close the window.

Set Processing Type for OC Data Extract Incremental Adapter

This profile allows you to set the default processing type for the incremental Oracle Clinical data extract adapter, which uses incremental-mode processing and is therefore faster than the regular OC data extract adapter, which uses full-mode processing. You can specify transactional processing with or without audit.

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter LSH: OC Incremental Adapter Default Processing Type
3. Click **Find**.
4. In the **Site** column:
 - Select \$PROCESSTYPES\$TRANSWAUDIT to use transactional processing with audit.
 - Select \$PROCESSTYPES\$TRANSWOAUDIT to use transactional processing without audit.
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Click the **X** in the upper right corner of the System Profile Values window to close the window.

Turn debugging on or off

Enable or disable the recording of debugging statements for database processes. The setting applies to all users. The system reads the value of the profile once per user session, so you may need to restart the application tier for a changed setting to take effect.

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter LSH:ENABLE_DEBUGGING
3. Click **Find**.
4. In the **Site** column, enter:
 - Y to enable debugging.
 - N to disable debugging.
5. In the File menu, select **Save and Proceed**.

Setting Password Requirements

Oracle recommends that you use the profile settings described in this section to provide optimal security in login password usage, including:

- [Setting Password Case Sensitivity \(Required for Business Intelligence Publisher\)](#) on page 3-9
- [Setting Password Length Requirement](#) on page 3-9
- [Setting "Hard to Guess" Requirement](#) on page 3-10
- [Enabling "Forgot Your Password?" Functionality](#) on page 3-10
- [Setting a Limit on Log-In Attempts](#) on page 3-11
- [Setting a Time Limit on Password Reuse after Resetting Password](#) on page 3-11

A password expiration utility is available if the System Administrator requires that all users convert to case sensitive passwords upon the next login. This utility expires all passwords in FND_USER, including that of SYSADMIN and default Vision accounts, and can be run as a SQL Script (\$FND_TOP/sql/AFCPEXPIRE.sql) or as a Concurrent Program (FNDCPEXPIRE_SQLPLUS).

For more information see the *Oracle E-Business Suite System Administrator's Guide - Maintenance* at http://docs.oracle.com/cd/E18727_01/doc.121/e12894/T202991T202995.htm.

Setting Password Case Sensitivity (Required for Business Intelligence Publisher)

This profile must be set to Insensitive if you are using Oracle LSH Programs of type Business Intelligence Publisher to enable users to log in to BI Publisher using their single sign-on credentials.

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter Signon Password Case
3. Click **Find**.
4. In the **Site** column, use the drop-down list to replace the default value, Sensitive, with Insensitive.
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Click the **X** in the upper right corner of the System Profile Values window to close the window.

Note: Passwords for existing user accounts must be reset after you change this setting.

Setting Password Length Requirement

This profile determines the minimum number of characters required in a user's Oracle LSH login password. The default setting is 5. Oracle recommends a setting of 8 or more for use with Oracle LSH.

To set the password length requirement, do the following:

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter Signon Password Length
3. Click **Find**.

4. In the **Site** column, replace the default value, 5, with 8 (or a higher number, if you choose).
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Click the **X** in the upper right corner of the System Profile Values window to close the window.

Setting "Hard to Guess" Requirement

This profile enforces requirements that make it more difficult to guess what another user's password might be. These requirements come as a package; you must either accept or reject the whole. Oracle recommends a setting of Yes (to accept the package) for use with Oracle LSH.

To set the "Hard to Guess" requirement, do the following:

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter `Signon Password Hard To Guess`
3. Click **Find**.
4. Click in the **Site** column. An ellipsis (...) appears on the right-hand side of the field.
5. Click the ellipsis (...). A pop-up box appears.
6. Click **Yes** and then **OK**.
7. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
8. Click **OK**. The transaction message pop-up disappears.
9. Click the **X** in the upper right corner of the System Profile Values window to close the window.

Enabling "Forgot Your Password?" Functionality

For Oracle LSH, Oracle recommends a setting of 40 for the Local Login Mask profile. This setting displays a "Forgot your password?" link on the Login page. If the user clicks this link, the system loads a page where the user can enter his or her username.

The user then receives an email stating, "Password reset requires approval." The user needs to click one of the choices "Approve" or "Reject" that automatically generate an email response. If the user ignores the notification, the request expires in four hours.

To create the local login mask, do the following:

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter `Local Login Mask`
3. Click **Find**.
4. In the **Site** column, replace the default value, 32, with 40.
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.

7. Click the X in the upper right corner of the System Profile Values window to close the window.

Setting a Limit on Log-In Attempts

This profile option determines the maximum number of logins a user can attempt before the user's account is disabled. To reinstate the account a system administrator must unlock the account and reset the password. For example, if the value set is 3, it will lock the account if the user enters incorrect password 3 times.

To set a limit on login attempts, do the following:

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter `Signon Password Failure Limit`
3. Click **Find**.
4. In the **Site** column, enter the limit you choose; for example: 3
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Click the X in the upper right corner of the System Profile Values window to close the window.

Setting a Time Limit on Password Reuse after Resetting Password

This profile will set the minimum number of days that a user must wait after changing his or her password before being allowed to reuse a password. The user can use the new password once and then must wait the number of days you set before he or she can reuse the password.

For example, if the value of this profile is set to 5, a user who changes his or her password cannot reuse the password until 5 days after they reset.

If the profile value is set to the number 0, then there is no restriction on password reuse.

To set a limit on login attempts, do the following:

1. Log in. See ["How to Log In to Oracle Applications Profile Forms"](#) on page 3-1.
2. In the **Profile** field, enter `Signon Password No Reuse`
3. Click **Find**.
4. In the **Site** column, enter the non-negative integer you choose; for example, 5 or 0
5. In the File menu, select **Save and Proceed**. The system displays a message that the transaction is complete.
6. Click **OK**. The transaction message pop-up disappears.
7. Click the X in the upper right corner of the System Profile Values window to close the window.

Oracle LSH: BIP Endpoint

This profile stores the namespace value of Oracle BI Publisher webservice:

`services/PublicReportService_v11`

Note: Do not change this profile in any way. Oracle LSH uses this profile for Oracle LSH BI Publisher Programs.

Attachment File Upload Restriction Default

This profile allows more file types to be uploaded than are allowed under E-Business Suite recommended security settings. Changing the setting is required to support submitting Load Sets and creating Table Descriptors from some types of files, including sasb7dat files.

Adding and Modifying Lookup Values

This section contains the following topics:

- [About Lookups](#) on page 4-1
- [Modifiable Lookups](#) on page 4-1
- [Logging In](#) on page 4-3
- [Querying and Viewing Lookups](#) on page 4-3
- [Adding, Modifying, or Discontinuing a Lookup Value](#) on page 4-4

About Lookups

The Oracle Life Science Data Hub (Oracle LSH) uses predefined lists of values for many items such as drop-down lists in the user interface. These are based on lookups in the seed data shipped with Oracle LSH. You can modify some of these lists of values.

Many database object type attributes used in Oracle LSH public APIs have lists of values based on these lookups (also called reference codelists); the naming convention is to end these attributes' names with `_rc`. You can use the procedures in this section to look up the allowed values for these attributes.

To view or modify Oracle LSH lookups, you must have the LSH Setup Admin privilege.

The changes you can make depend on the access level defined for each lookup:

- **System.** You cannot change anything.
- **Extensible.** You can create additional values and modify them but you cannot modify predefined ones.
- **User.** You can create additional values and change the text displayed in the user interface for existing values (the "meaning"), and discontinue the use of predefined and user-defined values.

Note: If you change the text displayed in the user interface for existing values, the Oracle LSH documentation and online help will not reflect what is in the user interface, which may cause confusion.

Modifiable Lookups

You can modify the following lookups in Oracle LSH:

File Types

(CDR_FILE_TYPES) The access level is **Extensible**. This lookup is used in defining Source Code types; see "Defining Programs" in the *Oracle Life Sciences Data Hub Application Developer's Guide*. The predefined values are Macro, Program, Oracle Reports, and SQL.

Font Name

(CDR_FONT_NAME) The access level is **User**. This lookup is used in Report Set post-processing; see "Defining Report Sets" in the *Oracle Life Sciences Data Hub Application Developer's Guide*. The predefined values are Arial and Courier. If you prefer to use different fonts in your PDF Report Sets, you can add those fonts as values.

OTD Default Papersize

(CDR_OTD_DEFAULT_PAPERSIZES) The access level is **User**. This lookup is used in Report Set post-processing; see "Defining Report Sets" in the *Oracle Life Sciences Data Hub Application Developer's Guide*. The predefined values are A4 and US Letter.

Report Type

(CDR_REPORT_TYPE) The access level is **User**. This lookup is used in Report Set definition; see "Defining Report Sets" in the *Oracle Life Sciences Data Hub Application Developer's Guide*. The predefined values are Figure, Listing, and Table.

SAS IDE Client

(CDR\$SAS_IDE_CLIENT) The access level is **User**. This lookup allows users to set a preference for which SAS client to use. There is one predefined value: SAS PC, which allows users to use PC SAS for Windows. If you want users to be able to use SAS Enterprise Guide, for example, add another meaning value of SAS EG with a code value of \$SAS_IDE_CLIENT\$SAS_EG.

Note: If you do not want your users to use PC SAS and do not want the SAS IDE Client option to appear in User Preferences, delete the SAS PC value. If you want to add it again later, you must enter its code value as: \$SAS_IDE_CLIENT\$SAS_PC.

To delete the value, select it and select **Delete** from the **Edit** menu. Do not delete the lookup itself.

Service Type

(CDR_SERVICE_TYPES) The access level is **Extensible**. This lookup is used in defining services; see [Chapter 1, "Setting Up Services"](#). The predefined values are: BI Publisher, BI Publisher Development, Zip Utility, Informatica, Informatica for Development, OBIEE Business Area Deploy Service, OBIEE Deployment Service, OBIEE Business Area IDE Service, OBIEE Business Area Install Service, Oracle Export, Oracle Warehouse Builder, PLSQL, PLSQL for Development, Oracle Reports for Development, Oracle Reports, SAS, SAS for Development, SQLPLUS, Text for SqlLoader, Workflow, XML Publisher, and XML Reports.

Summary Output Validation Status

(CDR_SOVS) The access level is **User**. This lookup is used in Report Set validation; see "Defining Report Sets" in the *Oracle Life Sciences Data Hub Application Developer's Guide*. The predefined values are Development, Quality Control, Production, Unassigned, and Not Applicable. You can change the displayed text for the existing values (in the Meaning column) but do not add values.

System Validation Steps

(CDR_SYS_VALIDATION_STEPS) The access level is **User**. This lookup is used in object validation; see "Common Development Tasks" in the *Oracle Life Sciences Data Hub Application Developer's Guide*. The predefined values are Development, Quality Control, Production, and Retired. You can change the displayed text for the existing values (in the Meaning column) but do not add values.

Tablespace Names

(CDR_TABLESPACE_NAMES) The access level is **Extensible**. When users define a Table instance, they can select a tablespace for it; see "Defining Tables" in the *Oracle Life Sciences Data Hub Application Developer's Guide*. The predefined values are APPS_TS_ARCHIVE, APPS_TS_MEDIA, and APPS_TS_TX_DATA.

Technology Types

(CDR_TECH_TYPES) The access level is **Extensible**. This lookup is used in Load Set, Data Mart, Business Area, Program, Report Set, and Workflow definition and execution. The predefined values are Oracle Clinical Design Definition, Oracle Clinical DX Views, Oracle Clinical DX SAS Views, Oracle Clinical General, Oracle Clinical Global Meta, Oracle Clinical Labs, Oracle Clinical Study, Oracle, Oracle Export Data Mart, PLSQL, Oracle Reports, Report Set, SAS Program, SAS Macro Catalog, SAS Data Mart, SAS Development, SAS Format Catalog, SAS Load Set, SAS tmp, SQLP, Text, Text Data Mart, Workflow, XML Publisher.

If you add an adapter to an external system you need to add the new technology type to this lookup.

Logging In

To modify lookup values, do the following:

1. Open your Oracle LSH URL and log in as a user with the LSH Administrator application role.
2. From the Navigator drop-down, select **LSH Setup Admin**, then **Setups**, then **Lookups**. An Oracle Applications window opens with the Oracle Life Sciences Data Hub Lookups window displayed.

Querying and Viewing Lookups

To view a lookup, or reference codelist, query for it as follows. Further information is available in the online help. See "[Modifiable Lookups](#)" on page 4-1 for a list of Oracle LSH lookups you can modify.

1. Press the **F11** key. The window enters Query mode and changes the background color of some of the fields.

2. In the **Type** field, enter the name of the lookup you want to see. You can enter part of the name and use the % wildcard. You can enter `cdr%` to retrieve all Oracle LSH lookups or `dme%` to retrieve all DMW lookups—then use the down arrow on the keyboard to view them in alphabetical order.

Note: During its initial development, Oracle LSH was known as CDR. Therefore many internal names contain the string `cdr`. Please think of CDR as a synonym for LSH.

Please think of DME as a synonym for DMW for the same reason.

3. Press **Ctrl+F11** to enter the query. The system displays the lookup and all its current values, including predefined values and those your company has previously added (if any). If your query retrieved more than one lookup, use the Down arrow on your keyboard to view each lookup in turn.

The following information is displayed for each value:

- **Code** is the internal name for the value.
- **Meaning** is the text that appears on the user interface, exactly as it appears here.
- **Description** is an explanation of the value.
- **Tag** is an optional descriptor.
- **Effective Dates.** The **From** column contains the date the value was created in your location. The **To** column may contain a date; if so, this is the date on which the value became or will become unavailable for use. If there is not a date in either column, the value is in effect.

Adding, Modifying, or Discontinuing a Lookup Value

Depending on the access level assigned to a lookup, you may be able to change its displayed text (the "Meaning") and add, modify, and discontinue existing values; see "[Modifiable Lookups](#)" on page 4-1.

The Lookups user interface is part of Oracle Applications functionality not specific to Oracle LSH. Additional information is available in the online help.

Registering Locations and Connections

This section contains the following topics:

- [About Remote Locations and Connections](#) on page 5-1
- [Defining Remote Locations](#) on page 5-2
- [Defining Connections](#) on page 5-4
- [Searching for Remote Locations and Connections](#) on page 5-5

About Remote Locations and Connections

To use an Oracle-technology source data system adapter (the general Oracle Tables and Views adapter and the Oracle Clinical adapters) you must create a database link and define one or more Remote Locations in the Oracle Life Sciences Data Hub (Oracle LSH) user interface for each actual remote location, and define one or more Connections giving a valid username and password for each Remote Location. All passwords are encrypted in Oracle LSH.

A user must select both a Remote Location and Connection to define and to run an Oracle-technology Load Set.

There are two types of Connections:

- **User Connections** are created by users to external databases for which they have a valid username and password. Users can see and use only the User Connections they have created.
- **Shared Connections** are created by an administrator. They also require a valid username and password to the external database, but are available for use to any user who belongs to a user group assigned to the Adapter Area for which the Remote Location is defined.

All the users in a user group assigned to an Adapter Area have access to any Remote Locations defined for that adapter. They can then define their own User Connection based on their personal username and password on the external database, or use a Shared Connection that the administrator has defined; see "[Defining Connections](#)" on page 5-4.

For example, if a user is a member of a user group assigned to the general Oracle Tables and Views Adapter Area and the user has Read privileges on a particular Oracle database, then the user can create a Connection to the Remote Location defined for that database. The user can also use any shared Connections the administrator has defined for the same Remote Location.

The CDR_SYS_ADMIN function role is required to define Remote Locations and Connections.

You can add Remote Locations and Connections at any time.

Security Remote Locations and Connections are defined objects with the same security requirements as other defined objects. To see a Remote Location and Connection during Load Set definition or execution, a user must be a member of a user group with access to the Remote Location and the Connections defined for it.

By default, Remote Locations and Connections inherit the user group assignments of their Adapter Area; see ["Setting Up Security for Adapters"](#) on page 2-4.

In addition, Oracle LSH displays user-specific Connections only to the user who created the Connection.

Note: The tnsnames.ora file in your Oracle LSH database tier must include each remote location.

Defining Remote Locations

If you need to load data into Oracle LSH from another system, you must define a Remote Location.

A Remote Location links the following as a pair:

- One of the Oracle-technology adapters used for loading data into Oracle LSH (Oracle Tables and Views, or one of the set of Oracle Clinical adapters)
- One database from which the adapter can load data into Oracle LSH

Oracle LSH also uses Remote Locations and Connections for storing user credentials for external systems with adapters. See:

- ["Storing the BI Publisher Local Superuser Credentials in Oracle Life Sciences Data Hub"](#) on page 12-6
- ["Creating an Informatica Remote Location and Connection"](#) on page 13-2
- ["Searching for Remote Locations and Connections"](#) on page 5-5

To define a Remote Location:

1. Click the **Remote Location** subtab under the Administration tab. The Maintain Remote Locations screen opens.
2. Click **Add Remote Location**. The Create Remote Location screen appears.
3. Enter values in the following fields:
 - **Name.** Enter a name for the Remote Location. It does not necessarily need to be the same as the name of the database.

Note: Do not use spaces, slashes, or special characters such as single quotes ('). If you use these characters the Remote Location you define does not appear in the list of values where it is needed.

- **Description.** Enter a description of the Remote Location.
- **DBLINK Prefix.** The name of the database link. If another DBLINK Prefix with the same name exists in the database, the system adds an additional

string to make it unique. The DBLINK_NAME is usually the global name or the TNS name of the remote database.

Note: To create a database link to a database whose name is different from the Oracle LSH database, you must set the `global_names` parameter of both databases to `FALSE`. To set the parameter, do one of the following:

- Edit the `init.ora` file to change the parameter value to:

```
global_names=FALSE.
```

- From a SQL prompt, enter:

```
SQL> alter system set global_names=false;
```

- **Connect String.** The name of the string that Oracle LSH must use in the `USING` clause of the create database link SQL statement.
- **Adapter.** Select the Adapter type from the drop-down list.
- **Conversion Multiplier.** (Optional) If you are defining a remote location for a database that uses a character set other than UTF8, you may need to enter a number such as 1.2. Oracle LSH uses this multiplier to adjust the length of VARCHAR2 data type columns in Load Set Table Descriptors.

This may be necessary because when you transfer VARCHAR2 data from a non-UTF8 character set database to Oracle LSH, which is a UTF8 character set database, the data can be longer in the UTF8 database. For example, 60 characters of data from a Western European (WE) database might convert to 62 characters in UTF8. If you have row values equal to or near the defined column length, loading data into Oracle LSH may fail if you do not specify a conversion multiplier greater than 1.

If you enter a multiplier of 1.2, Oracle LSH creates Table Descriptors as follows:

Table 5–1 Example of Column Length Conversion Using a Conversion Multiplier of 1.2

Table Columns in Non-UTF8 Database Table X	Table Columns in Table Descriptor X
VARCHAR2(60)	VARCHAR2(72)
VARCHAR2(10)	VARCHAR2(12)
NUMBER(10)	NUMBER(10)

The multiplier can be any number between 1.00 and 3.00, with a maximum of 2 decimal places. Columns of the Number and Date data types are not affected. If you do not enter a value, the system interprets it as a 1.

Note: If the value of the Conversion Multiplier causes an uploaded Table Column to have a length greater than 4000, which is the maximum length for Oracle columns of data type VARCHAR2, Oracle LSH automatically creates the column with a length of 4000. If the actual data written to the column in any job exceeds 4000 characters, the job fails and the log file gives the reason.

Note: Your database NLS Length Semantics parameter must be set to BYTE. If it is set to CHAR it may automatically perform conversions of 3x. To check the setting of the parameter, do the following query in SQL*Plus:

```
select * from V$NLS_PARAMETERS where parameter = 'NLS_LENGTH_
SEMANTICS' ;
```

However, it is required to use character semantics. To do so, set a profile value; see ["Using Character Semantics for Work Area Table Installation"](#) on page 3-2.

Note: The following types of Load Sets do not use conversion multipliers: SAS, Text, Oracle Clinical Labs, and Oracle Clinical Randomization. The length of VARCHAR2 columns in OC Labs and Randomization Load Set Table Descriptors is automatically increased.

4. In the **Classification** section, select the following for both the definition and the instance:
 - **Subtype.** Select a subtype according to your company's policies.
 - **Classification Values.** Specify classifications according to your company's policies, to enable users to search for the Remote Location by classification. For instructions, see "Classifying Objects" in the *Oracle Life Sciences Data Hub Application Developer's Guide*.
5. Click **Apply** to save your work. The system opens the main screen for the new Remote Location.

You can change the security and classifications of the Remote Location through the items in the Actions drop-down list. By default, the Remote Location inherits the user group assignments of the Adapter Area.

Defining Connections

A Connection provides a username and password that Oracle LSH uses to access a remote database. Oracle LSH encrypts the passwords.

To use a Remote Location, there must be at least one Connection defined for it. You, the administrator, can define shared Connections available to all users with normal security access to the defined Remote Location. In addition, any Oracle LSH user with an account and password on the remote location can define his or her own Connection through the Preferences link (see "Setting Preferences" in the *Oracle Life Sciences Data Hub User's Guide* for instructions). User-defined Connections are available only to the user who defined them.

To define a Connection:

1. In the main screen for the Remote Location for which you want to create a Connection, click **Create Connection**. The Create Connection screen opens.
2. Enter values in the following fields:
 - **Name.** Enter a name for the Connection.
 - **Description.** Enter a description of the Connection.

- **User Name.** Enter a valid user name for an account on the remote location. Oracle LSH will use this user name to connect to the remote location.
 - **Password.** Enter the password required for the same user account. Oracle LSH encrypts the password for security.
 - **Connection Type.** Select the Connection Type from the drop-down list.
 - **User.** The User option allows only the person who creates the Connection to use it.
 - **Shared.** The Shared option allows multiple users to use the Connection. Security is determined by user group assignment to the Connection.
 - **Remote Location.** The system populates the field with the name of the Remote Location for which you are defining this Connection. See "[Defining Remote Locations](#)" on page 5-2 .
3. In the **Classification** section, select the following for both the definition and the instance:
 - **Subtype.** Select a subtype according to your company's policies.
 - **Classification Values.** Specify classifications according to your company's policies, to enable users to search for the Connection by classification. For instructions, see "Classifying Objects" in the *Oracle Life Sciences Data Hub Application Developer's Guide*.
 4. Click **Apply** to save your work. The system displays the main screen for the new Connection.

You can change the security and classifications of the Connection through the items in the Actions drop-down list. By default, the Connection inherits the user group assignments of the Remote Location. If the user group assignments for the Remote Location and Connection are different, the user must have access to both in order to use the Remote Location/Connection combination.

Searching for Remote Locations and Connections

To find existing Connections, you must go to the Remote Location for which they are defined. To find existing Remote Locations, do the following:

1. Go to the **Remote Location** subtab under the **Administration** tab.
2. Enter a value in one or both of the following fields:
 - **Adapter Type.** From the drop-down list, select the Remote Location's adapter type.
 - **Remote Location Name (Optional).** Enter the Remote Location's name or part of its name plus the % wildcard.
3. Click **Go**. The system displays all Remote Locations that match these criteria in the lower part of the screen.

To see the Connections defined for the Remote Location or to define additional Connections, click the Remote Location's hyperlink in the **Name** column.

Enabling Internal Metadata Table Auditing

This section contains the following topics:

- [About Internal Table Auditing](#) on page 6-1
- [Audited Tables](#) on page 6-2

About Internal Table Auditing

The Oracle Life Sciences Data Hub (Oracle LSH) uses Oracle audit trail functionality to track all data changes in selected internal tables. Auditing tables helps to satisfy regulatory requirements. You must explicitly enable auditing on Oracle LSH tables. Instructions are included in the "Start Journaling Internal Tables" section in the chapter on installing Oracle LSH in the *Oracle Life Sciences Data Hub Installation Guide*.

For each audited table, Oracle creates a shadow table in the same schema in which to store the audit information. The shadow table contains additional columns that contain auditing information, including `AUDIT_USER_NAME`, `AUDIT_TIMESTAMP`, `AUDIT_TRANSACTION_TYPE`, and `AUDIT_TRUE_NULLS`. The name of the shadow table consists of the first 24 characters of the audited table's name plus `_A` (for Audit).

Oracle creates one or more views onto each shadow table. You can access the views in SQL*Plus when you log in as `apps`. Each view's name consists of the first 24 characters of the audited table's name plus `_AC#` or `_AV#`, where C and V are types of views and # represents a number. Combined (C)-type views allow you to see all columns and therefore reconstruct all row values at a particular point in time. View (V)-type views allow access to fewer columns, but provide information about when a value was changed. If there are multiple V-type views (due to size limitations), the system numbers them sequentially.

Oracle also creates three audit triggers and corresponding procedures for each table, one each for Insert (I), Update (U), and Delete (D). The trigger and procedure names both begin with the first 24 characters of the audited table's name followed by, for triggers, `_AI`, `_AU`, or `_AD` or, for procedures, `_AIP`, `_AUP`, or `_ADP`.

For further information, see the *Oracle® E-Business Suite System Administrator's Guide - Security* at http://download.oracle.com/docs/cd/B53825_08/current/acrobat/121sasg.pdf.

Audit Data Handling The automatically generated Insert, Update, or Delete trigger in the After event on the audited table performs the auditing. For each record, the trigger calls the corresponding procedure to compare each audited column to see if its value has changed. If changed, the system creates one row in the shadow table to store the old value(s).

The shadow table stores data in "sparse" format; unchanged values are stored as nulls. When a new record is inserted into an audited table, the system adds a row to the shadow table with a transaction type of Insert (I) with all column values represented as nulls (since null represents their previous, nonexistent, values). When a record is deleted from an audited table, the system adds a row to the shadow table with a transaction type of Delete (D) and all values stored.

Shadow tables do not compress data.

Audit Handling of Changes to Primary and Unique Key Values If a value changes in a column that is part of the table's primary or unique key, Oracle inserts two new rows in the shadow table, one with a transaction type of Delete (D) and one with a transaction type of Insert (I).

For example, because the STATUS_RC column in the CDR_ROLES_B table is part of the unique key, whenever you modify a role there is a change to a unique key column and the system adds two rows to the shadow table, one to delete the old record and another to insert the new one.

Audited Tables

When you enable auditing of internal tables in Oracle LSH, you enable auditing for a predefined set of tables and columns in those tables, listed below.

Where two tables share the same name with either `_B` or `_TL` appended, B means base and TL means translatable. If a table includes columns whose values can be translated and displayed on screen in a different language, the table is split into a base and a translatable table with the same primary key. For every row in the base table, the translatable table includes one row per supported language.

For further information, see the *Oracle® Application Framework Personalization Guide* at http://download.oracle.com/docs/cd/B53825_08/current/acrobat/121fwkpg.pdf.

- **CDR_CONNECTIONS.** This table stores information about user-defined connections to remote databases (both user-specific and shared connections). All columns are audited.
- **CDR_NAMINGS.** This table stores information on every user-defined object. Only the NAME column is audited.
- **CDR_NAMING_VERSIONS.** This table stores information on every version of every user-defined object. Only the VALIDATION_STATUS_RC column is audited.
- **CDR_OBJ_CLA_MAPPINGS.** This table stores information on the assignment of classification values to user-defined objects. The columns CLA_LEVEL_ID, CLA_OBJ_ID, PAR_COMPANY_ID, and PAR_OBJ_ID, PAR_OBJ_VER are audited.
- **CDR_OBJ_UG.** This table stores information on the assignment of user groups to user-defined objects. All columns are audited.
- **CDR_OBJECT_SUBTYPES_B.** This table stores base information on object subtypes. All columns are audited.
- **CDR_OBJECT_SUBTYPES_TL.** This table stores translated language values on object subtypes. All columns are audited.
- **CDR_OUTPUTS.** This table stores information on all outputs generated by user-defined executables. Only the BLINDING_STATUS_RC column is audited.

- **CDR_REMOTE_LOCATIONS.** This table stores information on remote databases registered by users in Oracle LSH. All columns are audited.
- **CDR_ROLES_B.** This table stores base information on user-defined roles. All columns are audited.
- **CDR_ROLES_TL.** This table stores translated language values for user-defined roles. All columns are audited.
- **CDR_SECURITY_DBACCOUNT.** This table stores information on all database accounts defined in the Oracle LSH user interface. All columns are audited.
- **CDR_SOURCE_CODES.** This table stores information on user-defined source code objects. Only the `SHAREABLE_FLAG_RC` column is audited.
- **CDR_SUBTYPE_OPR_ROLES.** This table stores information about the roles and object subtypes associated with each other. All columns are audited.
- **CDR_TABLE_REFS.** This table stores information about all user-defined Table instances. Only the `BLINDING_STATUS_RC` column is audited.
- **CDR_UG_ROLES.** This table stores information about the roles associated with each user group. All columns are audited.
- **CDR_USER_GROUPS_B.** This table stores base information on user groups. All columns are audited.
- **CDR_USER_GROUPS_TL.** This table stores translated language values for user groups. All columns are audited.
- **CDR_USER_UG_ROLES.** This table stores information about the roles and users associated with each user group. All columns are audited.
- **DME_FLAG_DATA_A.** This table stores the history of flag assignments to data in Oracle Health Sciences Data Management Workbench (Oracle DMW).

Setting Up Printers

This section includes the following topics:

- [Setting Up Printers Globally](#) on page 7-1
- [Setting Default Printers](#) on page 7-1

Setting Up Printers Globally

Use standard Oracle Applications technology for setting up printers to work with the Oracle Life Sciences Data Hub (Oracle LSH). Information on setting up printers is included in the *Oracle® E-Business Suite System Administrator's Guide - Configuration* at http://download.oracle.com/docs/cd/B53825_08/current/acrobat/121sacg.pdf.

Setting Default Printers

You can use the Oracle Applications Printer user profile to set default printers at the site, application, responsibility, or user level. Information is included in the *Oracle® E-Business Suite System Administrator's Guide - Maintenance* at http://download.oracle.com/docs/cd/B53825_08/current/acrobat/121samg.pdf.

Setting Up the Classification System

This section contains the following topics:

- [About Classification](#) on page 8-1
- [Creating and Maintaining Hierarchies](#) on page 8-2
- [Assigning Levels to Object Subtypes](#) on page 8-6

See also:

- "Designing a Classification System for Searching and Browsing" in the *Oracle Life Sciences Data Hub Implementation Guide*
- [Creating and Maintaining Object Subtypes](#) on page 9-9

About Classification

The Oracle Life Sciences Data Hub (Oracle LSH) classification system allows you to define a set of labels and use them to label, or classify, every defined object, output, and visualization in Oracle LSH. Users can search and browse for an object or output by its classifications.

The classification system consists of a set of hierarchies. A hierarchy can contain multiple logically related levels (such as Project/Study/Site) where the values, or terms, defined for each level are logically related to a single term in the next higher level (for example, the site Mercy Hospital is related to its study, Study01, which is related to its project, Project Wonderdrug.) an Oracle LSH "hierarchy" can also consist of a single level, in which case its terms function as a list of values.

Classification brings order to the potentially huge number of defined objects and outputs contained in the Life Sciences Data Hub. The Outputs and Visualizations subtabs under the Reports tab display reports, submission forms, and visualizations according to the classification hierarchies you define and the classification terms that Definers assign to outputs, similar to the file system on a PC.

For example, in the Reports tab a user can click on the Project/Study/Site hierarchy. Oracle LSH then displays all the project names defined as terms in the Project hierarchy level. If the user clicks on Project Wonderdrug, for example, Oracle LSH displays folders for all the studies that are part of the Wonderdrug project (including Study01) and also all the reports that are classified to Project Wonderdrug as a whole. If the user clicks on Study01, Oracle LSH displays folders for all the sites that are part of Study01 (including Mercy Hospital) as well as all reports that are classified to Study01 as a whole. If the user clicks on Mercy Hospital, Oracle LSH displays all reports classified to Mercy Hospital.

Further information is available in the chapter "Designing a Classification System" in the *Oracle Life Sciences Data Hub Implementation Guide*.

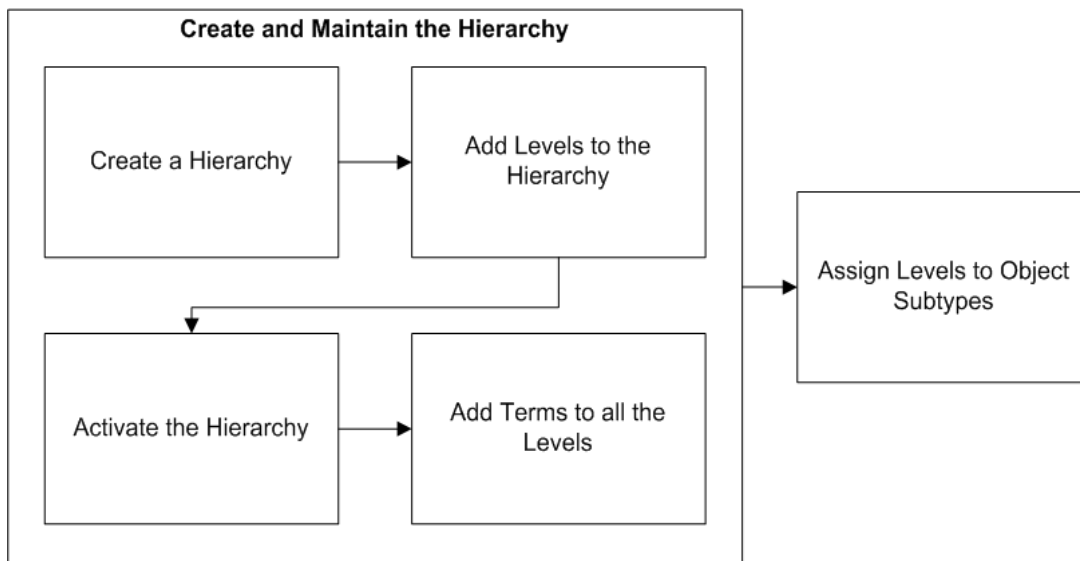
You must create the following:

- **Hierarchies.** A hierarchy consists of one or more named levels in a strict linear hierarchical order. Each level can have only one child level. You can define any number of levels.
- **Hierarchy Levels.** Each hierarchy must include at least one level. If a hierarchy has multiple levels, they must be logically related. For each object subtype, you must specify at least one hierarchy level to which objects of that subtype can be classified.
- **Terms.** A term is a specific value belonging to a hierarchy level. In each hierarchy level you must add meaningful terms to assign to objects. In a multilevel hierarchy, each term must be logically and definitionally related to a term in the level above. When a Definer assigns a term to an object, the object is classified to that term and users can search for the object by its classification.

You can add classification hierarchies and values at any time. [Figure 8-1](#) is a flowchart of the setting up process of the Classification System.

Required Role You must have the Classification Administrator application role to access any of the screens under the Classification tab in the user interface and to perform any of the tasks covered in this section. See ["Setting Up Specialized Administrators"](#) on page 9-7 for information on assigning that role.

Figure 8-1 Flowchart for Setting Up Classification System



Creating and Maintaining Hierarchies

This section contains the following topics:

- [Creating Hierarchies](#) on page 8-3
- [Adding Levels to Hierarchies](#) on page 8-3
- [Activating the Hierarchy and Making it Usable or Unusable](#) on page 8-4

- [Adding Terms to Levels](#) on page 8-5
- [Modifying Hierarchies and Terms](#) on page 8-5

Creating Hierarchies

To create a classification hierarchy, go to the Hierarchy subtab of the Classifications tab and do the following. For information about designing a system of classification hierarchies, see "Defining Classification Hierarchies" in the *Oracle Life Sciences Data Hub Implementation Guide*.

1. Click **Add Hierarchy**. The Create Hierarchy screen appears.
2. Enter values in the following fields:
 - **Name**
 - **Description**
 - **Keyword Hierarchy**. Set a value for the Keyword Hierarchy.
Select **Yes** to make the hierarchy a keyword hierarchy. Keyword hierarchies can have only one level and are not used to display outputs in the Reports screen. Keyword hierarchies can be used in advanced searches.
3. Click **Save and Continue**. The Maintain Hierarchy screen appears with blank rows for defining levels. See "[Adding Levels to Hierarchies](#)" on page 8-3.

Adding Levels to Hierarchies

You must define at least one level for every hierarchy. You must define the topmost level first, then the next lower level, and so on.

To add a level, do the following:

1. Enter values in the following fields:
 - **Name**
 - **Description**
 - **Entry Length**. The number of characters allowed for a term in this classification level. The maximum is 300.
 - **Term Uniqueness**. Select one of the following values to determine the way Oracle LSH enforces the uniqueness requirement for terms in the level:
 - **Unique Within Level**. Each term contained in this level must be unique relative to other terms in the same level.
 - **Unique Within Parent**. Each term contained in this level must be unique relative to the other terms in the level that are related to the same parent term.
 - **Allow Classification**
 - If selected, and if the hierarchy as a whole is set to Active, you can assign this level to object subtypes. Definers can then classify objects of those subtypes to values you define for the level.
 - If cleared, you cannot assign the level to object subtypes and definers cannot classify objects to the level. Oracle LSH uses terms in this level solely as folders in the Reports screen to organize folders in the next lower level.

2. Repeat this procedure to add another level. Click **Add Hierarchy Level** to add additional rows.
3. Click **Apply**. The system saves the changes and the Maintain Hierarchy screen appears.

To add a level after you have clicked **Apply**, click **Update**.

After creating a hierarchy, you must activate it before you can use it. See "[Activating the Hierarchy and Making it Usable or Unusable](#)" on page 8-4. You can remove and reorder a hierarchy level only if its owning hierarchy is not set to Active.

Reorder Hierarchy Levels

To change the order of the levels in a hierarchy, do the following:

1. Click **Reorder**. The Reorder shuttle appears.
2. Select the level you want to move and click the **Up** and **Down** arrows to move it in relation to the other levels.
3. Click **Apply**. The system returns to the Maintain Hierarchy screen with the levels in the new order.

Or click **Cancel** to discard your changes and return to the Maintain Hierarchy screen.

Remove Hierarchy Levels

To remove a level, select it and click **Remove**. If you remove a level, the system adjusts the order number of any remaining levels so that the order numbers start at one (1) and are consecutive.

Activating the Hierarchy and Making it Usable or Unusable

When you have finished defining the hierarchy and its levels, make a hierarchy active and usable.

When a hierarchy's Active flag is set to Yes:

- You can add terms to the levels in the hierarchy.
- However, you cannot add levels to an active hierarchy.
- You can never change the setting back to No.

When a hierarchy's Usable flag is set to Yes:

- You can assign the hierarchy to object subtypes.
- You can set it back to No as long as it is not assigned to any subtypes.

Activate and Make Usable To activate a hierarchy, do the following:

1. In the hierarchy's main screen, click **Update**. The fields become enterable.
2. Set **Active** to **Yes**.
3. Click **Apply**. The system also sets **Usable** to **Yes**.

Make Unusable If you prefer not to make the hierarchy usable at this time, do the following:

1. Click **Update**. The fields become enterable.
2. Set **Usable** to No.

3. Click **Apply**.

You can change the Usable setting to Yes at any time.

Adding Terms to Levels

All terms in levels below the top level must be related to a term in the next higher level. You must begin by defining at least one term in the highest level. You can then add terms related to that term (child terms) in the second level, and then define child terms for those terms in the third level, and so on.

Note: You can add terms only to active hierarchies. See "[Activating the Hierarchy and Making it Usable or Unusable](#)" on page 8-4.

To add terms to classification hierarchy levels, go to the Terms subtab of the Classifications tab and do the following:

1. Navigate to the hierarchy for which you want to define values.
 - If the hierarchy already has terms, you can expand its node (+) to see them. Add a term by clicking the pencil icon in the Maintain Terms column for the parent term (the term in the next higher level that is related to the term you are adding).

If you are adding a term in the highest level, click the Maintain Terms icon for the hierarchy itself.
 - If the hierarchy does not contain any terms, you must begin by creating a term in the highest level by clicking the pencil icon in the Maintain Terms column for the hierarchy.

The Maintain Hierarchy Terms screen opens.

2. If there are no blank rows, click **Add 5 Hierarchy Terms**. Five empty rows appear.
3. Enter values in the following fields for each new term:
 - **Name.** Enter the term itself as you want it to appear in the user interface for classifying objects. The name must conform to the length and uniqueness rules defined for its hierarchy level.
 - **Approved.** Select **Approved** to activate the Term. Only approved terms can be used as default classifications for object subtypes and for object classifications.

You can select the Approved flag at any time. You can clear (disapprove) a term only if the term has not been used and does not have any approved child terms.

For example, if Project Wonderdrug is related to Study01, and Study01's Approval flag is selected, you cannot clear the Approval flag for Project Wonderdrug.
4. Click **Apply** to save your work. The Hierarchy Terms screen appears.

Modifying Hierarchies and Terms

To modify a Classification Hierarchy, click **Update** in the Maintain Hierarchy Screen. Click **Apply** after making changes.

You can modify attributes as follows:

- **Name.** You can modify the name of the hierarchy. The system automatically updates the hierarchy name everywhere it appears in the user interface, including the classification screen for all objects classified to the hierarchy
- **Description.** You can modify the description.
- **Keyword Hierarchy.** You cannot modify the Keyword Hierarchy attribute.
- **Active.** You can change a hierarchy's Active setting from **No** to **Yes** at any time. However, once set to **Yes**, you can never set it back to **No**.
- **Usable.** When you set **Active** to **Yes**, the system automatically changes the hierarchy's Usable setting to **Yes**. You can change it back to **No** only if the hierarchy has not been assigned to any object subtype. A hierarchy is available for use only when its Usable autoboot is set to **Yes**.

Modifying Hierarchy Levels You can modify level attributes as follows:

- **Name.** You can modify the level name. The system automatically updates the level name everywhere it appears in the user interface, including the classification screen for all objects classified to the level or a level below it in the same hierarchy.
- **Description.** You can modify the description.
- **Entry Length.** You can modify the Entry length.
- **Term Uniqueness.** You cannot modify the Term Uniqueness attribute.
- **Allow Classification.** You cannot modify the Allow Classification attribute.

Modifying Terms To modify a term:

1. Go to the **Terms** subtab in the **Classifications** tab.
2. Navigate to the hierarchy that contains the term you want to modify.
3. Click the pencil icon in the Maintain Terms column of the *parent* of the term you want to update.
4. You can modify term attributes as follows:
 - **Name.** You can modify the name, which is the term itself. If you change the name, the system automatically updates the classification of any objects that are classified to the term.
 - **Approved**
 - You can change a term's Approved setting from **No** to **Yes** at any time.
 - You can change the Approved setting from **Yes** to **No** only if the term has not been used either as a default classification for an object subtype or to classify an object and no approved child terms are associated with the approved term.

You can remove a term only if it has not been used in any way and does not have any child values.

Assigning Levels to Object Subtypes

You must assign a classification hierarchy level to an object subtype in order to make it possible for a Definer to classify an object of that subtype at that hierarchy level. You can choose to make object classifications mandatory to a particular level or not, and you can set a default classification or not.

You can create subtypes for any object type in the Subtypes subtab of the Security tab if you have the required privileges.

To add classification levels to subtypes, go to the Subtype subtab of the Classification tab and do the following:

1. From the **Object** drop-down list, select an object type and click **Go**. The system displays the object type in the lower portion of the screen.
2. Expand the object's node by clicking the plus (+) sign. The system displays all the subtypes currently defined for the object type.
3. Click the plus (+) sign in the **Add Level** column on the Subtype's row. The Subtype Hierarchy Level Assignment screen appears. The system displays the Classification: Subtype Hierarchy Level Assignment screen for the object subtype.
4. Select values from the following fields:
 - **Hierarchy**. From the drop-down list, select the classification hierarchy you want to assign to the Subtype. The system populates the Hierarchy Level drop-down list with the levels defined for the hierarchy you select.
 - **Hierarchy Level**. From the drop-down list, select the level of the hierarchy. You can repeat this process to choose more than one level.
 - **Reference Allowed**. Select **Yes** or **No**.
 - **Yes**. If set to **Yes**, Definers can choose to set the classification for objects of this Subtype to Reference. With a classification of Reference, an object takes its classification value from its immediate container: either a Work Area, in the case of object instances, or an Application Area or Domain Library in the case of object definitions. If you select **Yes**, you can set the Default Type to either **Explicit** or **Inherited**.
 - **No**. If set to **No**, Definers cannot choose a classification of Reference. They must specify a particular term. If you select **No**, the system sets the Default Type to **Explicit** and you cannot change that setting.
 - **Default Type**. Select **Explicit** or **Inherited**.
 - **Explicit**. If set to Explicit, to classify an object of this subtype at this hierarchy level, the Definer must manually specify a value.
 - **Inherited**. If set to Inherited, objects of this subtype inherit their classifications for this level from their parent object. Definers can override the inheritance by explicitly specifying a classification.
 - **Mandatory**. Select **Yes** or **No**.
 - **Yes**. Definers are required to set a value for this classification level.
 - **No**. Definers are not required to set a value for the classification level.
5. Click **Apply**. The Classification screen appears.
6. You can select a particular term in this hierarchy level as the default classification for all objects created as this object subtype. Definers can override the default value. Click the plus icon (+) in the Add Default Value, and then query for the term you want to define as the default classification. Click **Apply**.

Setting Up the Security System

This section contains the following topics:

- [About Security in the Oracle Life Sciences Data Hub](#) on page 9-1
- [About Security in Oracle Health Sciences Data Management Workbench](#) on page 9-2
- [Setting Up User Accounts](#) on page 9-3
- [Setting Up Specialized Administrators](#) on page 9-7
- [Setting Up Object Security](#) on page 9-9
- [Granting Security Access to APIs](#) on page 9-16

See also:

- "Designing a Security System" in the *Oracle Life Sciences Data Hub Implementation Guide*.
- [Setting Up Security for Adapters](#) on page 2-4
- [Managing Database Account Privileges for Generic Visualization Business Areas](#) on page 10-3
- [Chapter 5, "Registering Locations and Connections"](#)
- "Managing User Groups" in the *Oracle Life Sciences Data Hub User's Guide*

About Security in the Oracle Life Sciences Data Hub

Before using the Oracle Life Sciences Data Hub (Oracle LSH), you must set up Oracle LSH security. You can continue to add to and modify security components as necessary over time. The basic tasks for setting up security are:

- **Create a user account for each user** and assign one or more application roles to each user account. Application roles allow access to part or the whole of the Oracle LSH user interface or grant blinding privileges. See "[Setting Up User Accounts](#)" on page 9-3.
- **Assign specialized administrative roles to a few users.** Each of these administrators must then perform a specific task: set up the classification system, create Domains to contain and organize objects, grant blind break and unblind privileges, or add users to user groups. See "[Setting Up Specialized Administrators](#)" on page 9-7.
- **Set up a security system for defined objects and outputs.** See "Designing a Security System" in the *Oracle Life Sciences Data Hub Implementation Guide* for information on how to design a security system for the contents of the Life

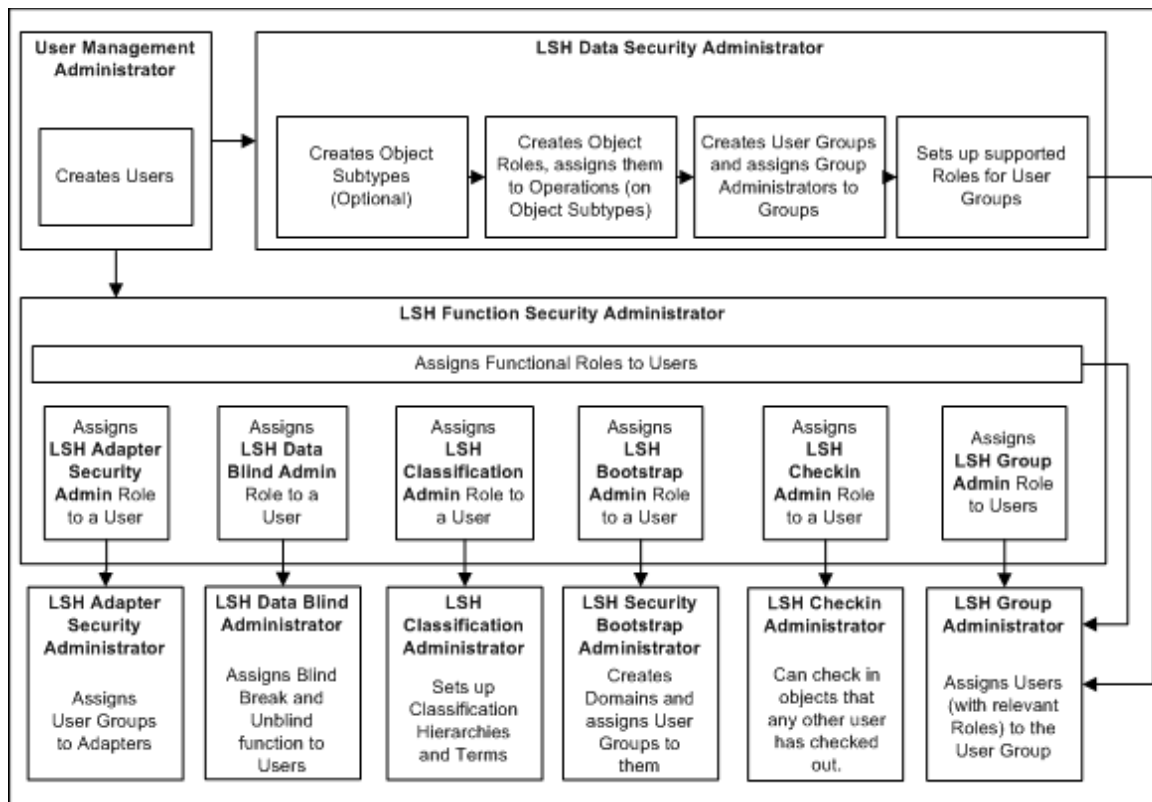
Sciences Data Hub: defined objects, outputs, and through them, data. Instructions are included in this chapter for creating the necessary subtypes, user groups, and object roles, and assigning roles to subtypes and user groups. See "[Setting Up Object Security](#)" on page 9-9.

Figure 9-1 on page 9-2 shows how these tasks relate to each other. The tasks are described in detail in the subsequent sections.

Required Roles To perform most of the tasks described in this chapter, you must be assigned either the LSH Security Administrator role or one of its component roles: LSH Function Security Administrator (for assigning administrative or other application roles to users) or LSH Data Security Administrator (for setting up object security.) Information on assigning these roles is included in the "Create System Administrator and Security Administrator Users" section in the chapter on installing Oracle LSH in the *Oracle Life Sciences Data Hub Installation Guide*.

Creating user accounts is done through a standard Oracle Applications UMX user interface, and you need the UMX sysadmin role to create user accounts.

Figure 9-1 Flowchart of Security Administrator Tasks



About Security in Oracle Health Sciences Data Management Workbench

Oracle Health Sciences Data Management Workbench (Oracle DMW) is installed on top of Oracle LSH and uses its security system, which is itself based in part on the Oracle Applications (Oracle E-Business Suite) security system. For introductory information, see the chapter on administration in the *Oracle Health Sciences Data Management Warehouse User's Guide*.

To set up Oracle DMW security:

- **Assign specialized administrative roles to a few users** to create Domains to contain and organize studies, grant blind break and unblind privileges, and add users to user groups. See "[Setting Up Specialized Administrators](#)" on page 9-7.
- **Create user accounts** for all users, including application roles for all users; see "[Creating User Accounts](#)" on page 9-4 and "[Assigning Application Roles](#)" on page 9-5.
- **Create user groups**; see "[Creating and Maintaining User Groups](#)" on page 9-13.
- **Assign object security roles to user groups.** You can use shipped roles created for use with Oracle DMW or modify them as required.

See the appendix "Predefined Roles" in the *Oracle Health Sciences Data Management Warehouse User's Guide* for more information.

- **Assign users to object security roles within user groups**; see "[Adding Supported Roles to User Groups](#)" on page 9-14.
- **Assign user groups to objects in Oracle DMW**; see the *Oracle Health Sciences Data Management Warehouse User's Guide*.

Setting Up User Accounts

This section contains the following topics:

- [About Users](#) on page 9-3
- [Creating User Accounts](#) on page 9-4
- [Maintaining User Accounts](#) on page 9-4
- [Assigning Application Roles](#) on page 9-5

About Users

Every person who will use Oracle LSH or Oracle DMW must have a user account. A user account includes:

- The user's name and other information; see "[Creating User Accounts](#)" on page 9-4.
- One or more application roles. These predefined roles determine which parts of the Oracle LSH or Oracle DMW user interface a user can see and manipulate. See "[Assigning Application Roles](#)" on page 9-5.

You use the Oracle Applications UMX interface for creating user accounts and assigning application roles.

To reach the User Management screens, log in to Oracle LSH as sysadmin and then select the User Management responsibility. For further information see "Create System Administrator and Security Administrator Users" in the chapter on installing Oracle LSH in the *Oracle Life Sciences Data Hub Installation Guide*.

Note: For complete information, see the *Oracle® E-Business Suite System Administrator's Guide - Security* at http://download.oracle.com/docs/cd/B53825_08/current/acrobat/121sasg.pdf.

Note: The system prevents you from modifying your own user account; the account you used to log in.

Creating User Accounts

System Administrators and Security Administrators can create user accounts in the system.

To register a user:

1. Log in as the system administrator.
2. Select **User Management** from the Navigator drop-down or from the Home page, then click **Users**. The Oracle User Management User Maintenance screen appears.
3. In the Register drop-down, select **External Organization Contact** and click **Go**.
4. Enter values in the following fields:
 - **Email.** The user's email address. Oracle LSH uses this address for corresponding with the user to reset passwords and send notifications.
 - **Name Fields.** Enter the name of the user in the fields. The First Name and the Last Name are mandatory. Prefix, Middle Name and Suffix are optional.
 - **Organization.** Enter or search for the Organization to which the user belongs.

Note: You must set up the list of allowed values as a post-installation step for Oracle Applications. See the *Oracle® E-Business Suite System Administrator's Guide - Security* at http://download.oracle.com/docs/cd/B53825_08/current/acrobat/121sasg.pdf.

- **Phone Number.** The telephone contact details of the user.
- **Account Information--Password.** If you select **Generate Automatically**, the system generates and emails the password to the email account you specified for the user.

If you select **Enter Manually** you must type and confirm the password and inform the user what it is. The user will have to reset the password in either case.

Note: You can set the minimum length and other requirements for passwords; see [Chapter 3, "Setting Profile Values"](#).

5. Click **Submit**. The Confirmation screen appears.
6. Click **OK**. The system creates the user and returns to the User Management screen.

Maintaining User Accounts

In the User Maintenance screen, search for the user you want to update and click the corresponding Update icon. After you make the relevant changes, click **Apply** to save them or click **Cancel** to discard the changes.

Name Fields Type the name of the user in the fields. The First Name and the Last Name are mandatory. Prefix, Middle Name and Suffix are optional.

Email Address The user's email address. Oracle LSH uses this address for corresponding with the user.

Active From The date the user account becomes active.

Active To The date the user account becomes inactive and the user can no longer use Oracle LSH. Leave this field blank to keep the user account active indefinitely. Enter an end time when the user leaves the company.

Roles See "[Assigning Application Roles](#)" on page 9-5.

Contact Information You can only view the user's contact information.

Assigning Application Roles

To assign an application role to a User:

1. In the User Maintenance screen, search for the user to whom you want to assign Roles.
2. Click the Update icon corresponding to the User. The Update User screen appears.
3. In the Roles subtab, click **Assign Roles**. The Search and Select screen appears.
4. Search for all Oracle LSH predefined roles by selecting Search By Role, entering LSH%, and clicking **Go**. The system displays all the predefined Oracle LSH application roles in the lower part of the screen.

For an explanation of application-type application roles required for using Oracle LSH, see "[Oracle Life Sciences Data Hub Application Roles](#)" on page 9-5 and "[Oracle Health Sciences Data Management Workbench Application Roles](#)" on page 9-7. For an explanation of administrative-type application roles, see "[Setting Up Specialized Administrators](#)" on page 9-7.

5. Select each role you want to assign by selecting its check box.
6. Click **Select**. The system adds the roles to the user and the Update User screen appears.
7. You must enter a justification for assigning each role to the user.
8. Click **Apply**. The system assigns the roles to the user and returns to the User Maintenance screen.

Oracle Life Sciences Data Hub Application Roles

Users need one or more of the predefined application roles to perform their work in Oracle LSH. The predefined nonadministrative Oracle LSH application roles follow. For descriptions of the administrative roles, see "[Setting Up Specialized Administrators](#)" on page 9-7.

LSH Application User The role on which all other application roles are based.

LSH Consumer Users assigned the LSH Consumer role have access to the Home and Reports tabs in the user interface. Assign this role to users who need to retrieve information from Oracle LSH and/or run Oracle LSH applications to load and

transform data. Oracle LSH object security determines which operations these users are allowed to perform on particular applications and reports.

LSH Definer Users assigned the LSH Definer role have access to the same tabs as the LSH Consumer, plus the Applications tab. Assign this role to users who must build, test, or validate Oracle LSH applications. Oracle LSH object security determines which operations these users are allowed to perform on particular defined objects and outputs.

LSH Data Blind Break User Only the LSH Data Blind Administrator can assign this role to users. This role does not provide access to any tabs in the user interface. Users assigned to this role typically also have the LSH Consumer role.

Users with this application role can do the following, if they have normal object security access and the blinding-related object security privilege noted:

- Run a job on Tables with a Blinding Status of Blinded that displays the real data, not the dummy data (also requires an object security role with the Blind Break operation on Table instances)
- View the output(s) generated by a job run on real, blinded data (also requires an object security role with the Blind Break operation on outputs)

LSH Data Unblind User Only the LSH Data Blind Administrator can assign this role to users. This role does not provide access to any tabs in the user interface. Users assigned to this role typically also have the LSH Consumer role.

Users with this application role can do the following, if they have normal object security access and the blinding-related object security privilege noted:

- Permanently unblind data in Table instances (also requires an object security role with the Unblind operation on Table instances)
- Change the status of an output from Blinded to Unblinded (also requires an object security role with the Unblind operation on outputs)

Note: The LSH Data Unblind User application role is not required in conjunction with the Read Unblind operation on either Table instances or outputs. Users can have an object security role that includes the Read Unblind operation on Table instances (to run a job on unblinded Table instances) and/or outputs (to view the results of such a job) without having the LSH Data Unblind User application role as well.

XMLP Roles You can assign the BI Publisher roles (XML Publisher is the old name for Oracle BI Publisher) to Oracle LSH Definers to enable them to create or use Oracle LSH BI Publisher Programs. See "[Assigning Application Roles in Oracle LSH](#)" on page 12-3 for more information.

Note: These predefined roles and composite roles should be sufficient for your needs. However, it is possible to create custom composite roles. See the *Oracle® E-Business Suite System Administrator's Guide - Security* at http://download.oracle.com/docs/cd/B53825_08/current/acrobat/121sasg.pdf.)

Oracle Health Sciences Data Management Workbench Application Roles

Each user must have at least one of the following application roles to access the Oracle DMW user interface.

DMW_STUDY_MANAGER is intended for users who run data loads, transformations, and validation checks. It provides access to the Home, Study Configuration, Listings, and Discrepancies pages.

Users with this role can create, modify, and remove studies in the Home page.

DMW_STUDY_CONFIG is intended for users who set up studies by defining models, transformations, and validation checks. It provides access to the Home, Study Configuration, Listings, and Discrepancies pages.

Users with this role can create, modify, and remove studies in the Home page.

Note: There is no functional difference between the application roles DMW_STUDY_MANAGER and DMW_STUDY_CONFIG.

DMW_STUDY_CONSUMER is intended for users who need to review data and raise discrepancies. It provides access to the Home, Listings, and Discrepancies pages.

DMW_LIB_ADMIN is intended for users who create and modify library models and code lists. It provides access only to the Library page.

DMW_SYS_ADMIN is intended for users who do administrative tasks including setting up data sources and defining objects used across studies, such as categories, flags, and tags. It provides access only to the Administration tab.

Note: In addition, some Oracle DMW users should have blinding-related privileges in order to view blinded data and/or unblind data. This requires both a blinding-related application role--[LSH Data Blind Break User](#) or [LSH Data Unblind User](#)--and blinding-related object security privileges through a role in a user group assigned to the object, plus normal object security access to the object. For more information about blinding data in Oracle LSH, see the *Oracle Life Sciences Data Hub Implementation Guide* and the *Oracle Life Sciences Data Hub Application Developer's Guide*.

Setting Up Specialized Administrators

As part of setting up Oracle LSH, the Security Administrator needs to assign the following administrative application roles to users. Each of these administrators has special privileges and responsibilities as follows:

LSH Adapter Security Admin The LSH Adapter Security Admin role is required to see the Adapter Security subtab of the Security tab and to assign user groups to Adapter Domains and Adapter Areas. See [Chapter 2, "Setting Up Adapters to External Systems"](#) for further information.

LSH Classification Admin The Classification Administrator creates and maintains classification hierarchies and terms. See [Chapter 8, "Setting Up the Classification System"](#) for more information.

LSH Data Blind Admin The Data Blind Administrator is responsible for assigning the Blind and Unblind application roles to users. See "LSH Data Blind Admin" in the *Oracle Life Sciences Data Hub Implementation Guide* for more information.

LSH Data Security Admin The Data Security Administrator sets up security for defined objects.

LSH Checkin Admin The Checkin Administrator can check in objects checked out by any user, provided he or she also has the Modify object security privileges for the object. This is required if the user who has checked out the object becomes unavailable for some reason. Note that a Checkin Administrator can install Reports Sets, and Work Areas that contain objects checked out by other users.

LSH Function Security Admin The Function Security Administrator assigns functional roles to users.

LSH Groups Admin Users with this role can be designated the Group Administrator of a user group. This role allows them to see the Groups Administration subtab under the Security tab, which is required for assigning users to user groups. See "[Creating User Groups](#)" on page 9-14 for more information.

LSH Security Admin The Security Administrator role includes both the LSH Function Security Admin role's functions and the LSH Data Security Admin role's functions. Instructions for setting up the first user with this role are included in the "Create System Administrator and Security Administrator Users" section in the chapter on installing Oracle LSH in the *Oracle Life Sciences Data Hub Installation Guide*.

LSH Security Bootstrap Admin This Administrator is responsible for creating Domain and assigning user groups to the Domains. See instructions for creating Domains in "Applications User Interface" of the *Oracle Life Sciences Data Hub Application Developer's Guide*.

LSH System Admin The LSH System Administrator can run the post-installation job and define service locations and services. Instructions for setting up the first user with this role are included in the "Create System Administrator and Security Administrator Users" section in the chapter on installing Oracle LSH in the *Oracle Life Sciences Data Hub Installation Guide*.

LSH Super User The LSH Super User has access to all user interface tabs in LSH: Home, Applications, Reports, Classification, Security, and Administration, as well as the Oracle Applications user and role screens. The Super User role does not include the Bootstrap role or the blinding-related roles.

To create an Administrator:

1. Log in as the Security Administrator.
2. Create a user account. See "[Creating User Accounts](#)" on page 9-4.
3. Assign one of the administrative application roles (listed above) to the user. See "[Assigning Application Roles](#)" on page 9-5.

XMLP Admin The XMLP Admin has administrative privileges in Oracle BI Publisher. Assign this role to an Oracle LSH Administrator who needs to perform administrative tasks (manage /delete reports and folders) in Oracle BI Publisher.

Setting Up Object Security

This section contains the following topics:

- [Creating and Maintaining Object Subtypes](#)
- [Creating and Maintaining Object Security Roles](#)
- [Assigning Subtype Operations to Roles and Modifying Assignments](#)
- [Creating and Maintaining User Groups](#)

Your company must design a security system that meets its particular needs. For information, recommendations, and examples of security system design, see "Designing a Security System" in the *LSH Implementation Guide*.

Each time a user tries to perform an operation on a defined object, the system runs a check that compares the security privileges of the user with the security requirements of the object.

A user can operate on an object only if both these conditions are met:

- The user belongs to an active user group that is assigned to that object, either explicitly or through inheritance.
- The user has a role in that user group that permits the operation on the object's subtype.

The *Oracle Life Sciences Data Hub Implementation Guide* has information on designing an appropriate set of object subtypes, roles, and user groups to meet your company's needs. This section contains the instructions for actually creating these things in Oracle LSH.

Creating and Maintaining Object Subtypes

This section contains the following topics:

- [Creating Object Subtypes](#) on page 9-10
- [Modifying Object Subtypes](#) on page 9-10

See also:

[Assigning Subtype Operations to Roles and Modifying Assignments](#) on page 9-12

Oracle LSH includes a set of predefined object types, such as Tables, Programs, Report Sets, and Variables. Each of these object types has a predefined set of all possible operations that can be performed on objects of that type.

To allow you greater flexibility as you set up your security system, Oracle LSH bases object security on subtypes rather than the predefined object types. You can define different subtypes—for example, Clinical and Financial—for one or more object types and define different classification and security requirements for each subtype of the same object type. Object subtypes have the same predefined set of operations as their object type. Oracle LSH ships with one default subtype for each object type. If you do not wish to add the flexibility (and complexity) of subtypes to your design, you can use the predefined subtype only.

Defining additional subtypes is optional.

When a Definer creates an object, he or she must select an object subtype on which to base the object. The object has the classification and security requirements defined for the selected subtype.

When the Definer creates an object, the system checks the subtype of the parent object and, if a subtype with the same name is defined for the new object, uses that subtype for the new object. If the new object does not have a subtype with the same name, the system uses the predefined Default subtype. The Definer can change it as necessary.

For further information, see "Designing a Security System" in the *LSH Implementation Guide*. For information on using subtypes in classification, see "[Assigning Levels to Object Subtypes](#)" on page 8-6.

Creating Object Subtypes

To create a new object subtype, go to the Subtype subtab of the Security tab and do the following:

1. In the Manage Subtype screen, choose either Operation or Role from the View Subtypes drop-down list.
 - Choose **Role** if you have created a new role and want to add multiple operations to it at the same time. This is the more common situation.
 - Choose **Operation** if you want to add multiple roles to a single operation on the subtype at the same time.
2. Click the node (+) icon in the Manage column of the object type for which you want to add a Subtype. The Create Subtype screen appears.
3. Enter a name for the Subtype (required, maximum 60 characters). You do not need to include the name of the Object Type in the Subtype name.
4. Click **Apply**. The system creates the Subtype and the Manage Subtype screen reappears. The new Subtype is automatically created as:
 - **Not Active**; not available for use in object definition.
 - **Not the Default**; when a Definer creates an object of this type, this Subtype is not the default Subtype of the new object. See "[Default Subtype Behavior](#)" below for further information.

To change either the Active or Default setting, click the pencil icon in the Update column.

Default Subtype Behavior The subtype you set as the default here is not necessarily the one that all new objects of this type default to. Instead, when a user creates a new object, the system reads the subtype of the object's parent, and if the new object has a subtype with the same name as the parent's, the system creates the new object with the same subtype as its parent. If not, the system uses the default subtype.

For example, if a user creates a new Program instance in a Work Area, and the Work Area's subtype is Clinical, the system checks whether there is a Program instance subtype called Clinical. If there is, the system creates the new Program instance with a subtype of Clinical. If Program instances do not have a subtype called Clinical, the system creates the new Program instance with the subtype defined as the default for Program instances.

The user can change the subtype manually.

Modifying Object Subtypes

To modify a Subtype, click the corresponding Update icon in the Manage Subtype screen.

You can modify the following Subtype characteristics:

- **Name**
- **Is Active.** If selected, this Subtype is available for use during definition of objects of this Object Subtype.
- **Is Default.** If selected, this is the default Subtype for objects of this Object Subtype.

Creating and Maintaining Object Security Roles

This section contains the following topics:

- [Creating Object Security Roles](#) on page 9-12
- [Modifying Object Security Roles](#) on page 9-12

A role consists of a name, description, and a set of operations allowed on object subtypes. Users in a user group that have access to a particular object will be able to perform the operations on the types of objects specified by the role(s) they have in the user group.

Predefined roles intended for use with Oracle Health Sciences Data Management Warehouse (Oracle DMW) are shipped with Oracle LSH. These roles appear in the Oracle LSH Manage Roles page. Although they were created for use with Oracle DMW, you can also use these roles in Oracle LSH user groups either as they are or with any modifications you choose. Additional privileges are required to perform Oracle LSH tasks on objects that do not exist in Oracle DMW.

You can also define your own roles for either Oracle LSH or Oracle DMW. See "Designing a Security System" in the *Oracle Life Sciences Data Hub Implementation Guide* for information on designing a set of roles.

Creating Object Security Roles

To create a Role:

1. In the Security tab, click **Roles**. The Manage Roles screen appears.
2. Click **Create**. The Create Role screen appears.
3. Enter values in the following fields:
 - **Role Name.** Enter a unique name for the role to be displayed in the Oracle LSH user interface.
 - **Code Value.** Enter a unique code for the role to be used internally by Oracle LSH.
 - **Description.** Enter a description of the role to help group administrators decide if they want to assign the role to a particular user in their user group.
4. Click **Apply**. The Manage Roles screen appears. You can query for the role by name to check that it was created successfully.
5. Set it to Active so that it is available for use.
 1. In the Manage Roles screen, query for the role. The role appears.
 2. Click the icon in the **Update** column. The Update Role screen appears.
 3. Select **Is Active** and click **Apply**. You can assign subtype operations only to Active roles.
6. Assign operations on object subtypes to the role. Users assigned to the role within a user group will be able to perform the operations you specify on the object subtypes you specify. See "[Assigning Subtype Operations to Roles and Modifying](#)

[Assignments](#)" on page 9-12 for instructions.

Modifying Object Security Roles

In the Manage Roles screen, click the corresponding Update icon to modify a particular Role. After you make the relevant changes, click **Apply** to save them or click **Cancel** to discard the changes.

You can modify the following characteristics:

- **Role Name.** You can change the name of the role that appears in the Oracle LSH user interface.
- **Description.** You can change the description that appears in the Oracle LSH user interface.
- **Is Active.** If selected, you can assign the role to operations on object subtypes and a user group administrator can assign the role to a user in his or her user group.

Assigning Subtype Operations to Roles and Modifying Assignments

This section contains the following topics:

- [Assigning Roles to Operations](#) on page 9-12
- [Modifying Operation Assignments to Roles](#) on page 9-13
- [Notes on Particular Object Types](#) on page 9-13

Subtype operations are predefined. They are the same as the predefined operations on their object type. You must assign at least one role to each operation; if no roles are assigned to the operation, no one will be able to perform the operation on objects of that subtype.

Users who are assigned to a particular role can perform operations on the subtype that are assigned to the same role, if the user is assigned to the role in the context of an active user group assigned to the object.

See "Designing a Security System" in the *Oracle Life Sciences Data Hub Implementation Guide* for information on how Oracle LSH security uses Roles and Operations.

In the Subtypes subtab in the Administration tab, you can see the relationship of roles to object operations two ways. Make your choice in the **View Subtype By** drop-down list:

- Choose **Operation** to add a new role to operations or to add multiple roles to a single operation on a subtype at the same time.
- Choose **Role** if you have already assigned the role to operations and want to view or modify the role's assignments.

Assigning Roles to Operations

To assign roles to operations, go to the Subtype subtab of the Security tab and do the following:

1. Select **Operation** from the **View Subtype By** drop-down list.
2. Expand the node (+) of the object type. All the subtypes appear.
3. Expand the node (+) of the object subtype. All the subtype's operations appear.
4. Click the icon in the Manage column of the operation to which you want to assign or remove a role. The Add Role(s) to Operation screen appears.

5. Double-click on a role in the Available Roles column to move it into the Selected Roles column and assign it to the role. Or move it from Selected Roles to Available Roles to remove the assignment.

You can also select multiple roles by using Shift+Click or Control+Click and use the arrows to move one, a few, or all at once to the column in the shuttle.

You can change the order of the operations using the Up and Down arrows on the right. The display order has no functional effect.

6. Click **Apply**. The system assigns the role(s) to the operation on the subtype and the Manage Subtype screen appears.

Modifying Operation Assignments to Roles

To modify assignments, go to the Subtype subtab of the Security tab and do the following:

1. Select **Role** from the **View Subtype By** drop-down list.
2. Expand the node (+) of the object type. All the subtypes appear.
3. Expand the node (+) of the object subtype. All the roles currently assigned to any operation on the subtype appear.
4. Click the icon in the Manage column of the role whose assignments you want to modify. The Add Operation for Role screen appears.
5. Double-click on an operation in the Available Operations column to move it into the Selected Roles column and assign it to the role.

You can also select multiple operations by using Shift+Click or Control+Click and use the arrows to move one, a few, or all at once to the column in the shuttle.

You can change the order of the operations using the Up and Down arrows on the right. The display order has no functional effect.

6. Click **Apply**. The system adds the operations to the role for the subtype and the Manage Subtype screen appears.

Notes on Particular Object Types

Some object types require some explanation:

- **Execution Setups.** To delete or modify an Execution Setup, a user must also have Modify privileges on the object instance that owns the Execution Setup. Therefore, any role you assign to Remove or Modify Execution Setups, you should also assign to Modify the type executable object whose Execution Setup it can remove or modify.
- **Adapter Domains and Adapter Areas.** These include all the same operations that Domains and Application Areas have. However, only view operations on some objects are required; see "[Creating Roles with the Required Operations for Adapters](#)" on page 2-5.

Creating and Maintaining User Groups

This section contains the following topics:

- [Creating User Groups](#) on page 9-14
- [Adding Supported Roles to User Groups](#) on page 9-14
- [Duplicating User Groups](#) on page 9-15

- [Assigning a Group Administrator to the User Group](#) on page 9-15
- [Modifying User Groups](#) on page 9-16

User groups control the access of users to objects and outputs (and through them, to data). A user has access to an object only if he or she belongs to a user group that is assigned to the object, either explicitly or through inheritance. See "User Groups" in the *Oracle Life Sciences Data Hub Implementation Guide* for further information.

A user group definition consists of a name, description, and a set of object security roles supported by the group (available for assignment to users within the group). Each group must also have an assigned Group Administrator.

You may require the same or very similar user groups, with the same or different users, assigned to different objects. For example, you might have a Study01 user group assigned to your Study01 Application Area. Study02, which has its own Application Area, might require a user group with exactly the same set of roles as Study01. Oracle LSH allows you to copy a user group either with or without users. You can then modify the name or roles assigned as necessary. The Group Administrator can add and remove users and/or change their roles within the copied group.

Group Administrator tasks, adding roles to a group and assigning users to a group and to roles within the group, are covered in "Managing User Groups" in the *Oracle Life Sciences Data Hub User's Guide*.

Creating User Groups

To create a new user group:

1. In the Security tab, click **User Groups**. The Manage User Groups screen appears.
2. Click **Create**. The Create User Group screen appears.
3. Enter values in the following fields:
 - **Group Name**
 - **Description**
 - **Is Active**. If selected, you can assign the user group to an object subtype.
4. Click **Apply**. The Manage User Group screen appears.

You must assign at least one role to a user group. See Step 5 of "[Adding Supported Roles to User Groups](#)" on page 9-14.

Adding Supported Roles to User Groups

The roles you add to a user group become available for the Group Administrator to assign to users within that user group.

To add roles to a user group:

1. In the Security tab, click **User Groups**. The Manage User Groups screen appears.
2. Type the name of the user group for which you want to add roles in the user group box.
3. Click **Search**. All the user groups matching the search criteria appear.
4. Click the required user group. The Group screen appears.
5. Click **Add and Remove Role**. The Add a Default Role to Group screen appears.
6. Double-click on a role in the Available Roles column to move it into the Selected Roles column.

You can also select multiple Roles by using Shift+Click or Control+Click and use the arrows to move one, a few, or all at once to the column in the shuttle.

You can change the order of the Roles using the Up and Down arrows on the right.

7. Click **Apply**. The system assigns all the roles displayed in the Selected Roles column to this user group and returns to the Group screen.

Note: You must assign a Group Administrator to every user group. See "[Assigning a Group Administrator to the User Group](#)" on page 9-15.

Assigning a Group Administrator to the User Group

The Group Administrator is responsible for adding and removing users to a particular user group, and changing users' role assignments within the group.

You must have the LSH Security Admin or LSH Function Security Admin role to assign the LSH Group Admin role to a user.

To assign a Group Administrator to a user group:

1. Select the **Life Sciences Data Hub** responsibility in the navigator and click the **Users Groups** subtab. The Manage User Groups screen appears.
2. Search for and select the required group.
3. Click **Expand All**. All the supported roles appear.
4. Click the **Add User** icon corresponding to the LSH Group Administrator role. The Search and Select Users to Add to Role screen appears.
5. Search for and select the users who need to be LSH Group Administrators for this User Group.
6. Click **Select**. The system assigns the LSH Group Administrator role to the selected users and the Group screen appears.

Note: Group Administrators must have the Group Admin application role assigned. See [Assigning Application Roles](#) on page 9-5.

Duplicating User Groups

You can create copies of a user group in two ways. Either you can duplicate only the user group definition with its supported roles or you can duplicate the definition with its supported roles and also the users assigned to it and their role assignments. When you duplicate a user group, the system appends 'Copy of' to the name of the user group. For example, if you duplicate a user group named *Study01 User Group*, the system names the new user group *Copy of Study01 User Group*.

Duplicate Definition with Roles Only To duplicate only the user group definition with its roles:

1. In the Security tab, click **User Groups**. The Manage User Groups screen appears.
2. Enter a search string—the name of the user group you want to duplicate in the User Group box or text with the wildcard %, or leave blank to retrieve all user groups.
3. Click **Go**. All the user groups matching the search criteria appear.

4. Click the **Select** check box corresponding to the required user group.
5. Click **Duplicate with Roles**. The system creates a copy of the user group and the Manage User Groups screen appears.

Duplicate Definition with Roles and Users To duplicate the definition with its roles and assigned users:

1. In the Security tab, click **User Groups**. The Manage User Groups screen appears.
2. Type the name of the user group you want to duplicate the User Group box.
3. Click **Search**. All the user groups matching the search criteria appear.
4. Click the **Select** check box corresponding to the required user group.
5. Click **Duplicate with Roles, Users**. The system creates a copy of the user group and the Manage User Groups screen appears.

Modifying User Groups

In the Manage User Groups screen, click the corresponding Update icon to modify a particular User Group. After you make the relevant changes, click **Apply** to save them or click **Cancel** to discard the changes.

You can modify the following characteristics:

- **Group Name**
- **Description**
- **Is Active**. If selected, you can assign the User Group to an Object Subtype.

You can also add and remove Roles from the User Group; see "[Adding Supported Roles to User Groups](#)" on page 9-14.

Granting Security Access to APIs

In order to run any of the public APIs for Oracle LSH from outside Oracle LSH, a user must have the Execute privilege granted on the API package `cdr_pub_api_initialization`.

To grant this privilege to a user, do the following:

1. Log in to SQL*Plus on the Oracle LSH database.
2. Enter:

```
Grant execute on cdr_pub_api_initialization to user_name;
```

Creating Database Accounts

This section contains the following topics:

- [About Database Accounts](#) on page 10-1
- [Creating Database Accounts](#) on page 10-2
- [Managing Database Account Privileges for Generic Visualization Business Areas](#) on page 10-3
- [Setting Up TMS Security for Users](#) on page 10-4
- [Resetting the Password](#) on page 10-4
- [Removing a Database Account](#) on page 10-4

You must have the LSH System Admin functional role to do all the tasks described in this section.

About Database Accounts

You must create an Oracle Life Sciences Data Hub (Oracle LSH) database account for users who need access to the Oracle LSH database through an external system or remote database, including:

- [Database Accounts for Use in Definition](#) on page 10-1
- [Database Accounts for Message-Based Submissions](#) on page 10-2

Database Accounts for Use in Definition

You must create an Oracle Life Sciences Data Hub (Oracle LSH) database account for Definers who need to use an integrated development environment (IDE) that requires logging back into the Oracle LSH database to view Oracle LSH data. These IDEs include:

- SAS in Connected mode (read-only access)
- Oracle Reports (read-only access)
- SQL*Plus (read and write access)
- Informatica (read-only access)
- Oracle Business Intelligence (read-only access)

When the Definer launches the IDE, he or she is typically prompted to enter an Oracle LSH database account username and password.

Oracle LSH database accounts maintain a mapping between a Definer's regular Oracle LSH application user account and his or her database account. If the Definer enters the same database account information that is mapped to his or her Oracle LSH user account, the system grants access to the data required by the Program the Definer is working on.

Note: A database account is not required to use Oracle Business Intelligence Enterprise Edition to create visualizations of Oracle LSH data.

Database Accounts for Message-Based Submissions

It is possible to trigger the execution of a job in Oracle LSH by sending an XML message from an external system on a remote database. To set this up, do the following:

- Create an Oracle LSH user account and database account
- Create a database link on the remote database to the Oracle LSH database using the Oracle LSH database account ID and password
- In the XML messages, embed the Oracle LSH user account ID in the appropriate place in the XML message. See "XML Message Requirements" in the *Oracle Life Sciences Data Hub Application Developer's Guide* for information on the required XML schema for these messages.

An Oracle LSH API called CDR_EXE_MSG_API with the procedure Submit Message is available for use in enqueueing messages. See "Using Message-Triggered Submission from External Systems" in the *Oracle Life Sciences Data Hub Application Developer's Guide* for further information.

For general information about enqueueing messages, see the *Oracle® Streams Advanced Queuing User's Guide and Reference* at http://download.oracle.com/docs/cd/B19306_01/server.102/b14257.pdf.

Creating Database Accounts

To create a database account for a Definer, do the following:

1. Click the Database Account subtab under the Administration tab. The Database Account screen opens.
2. Click **Create**. The Create Database Account screen opens.
3. Enter values in the following fields:
 - **User Name.** Click the Search icon and enter search criteria for the Oracle LSH user for whom you are creating a database account.
 - **Database Account Name.** Enter a username for the database account. The text you enter is stored in uppercase.
 - **Password.** Enter a password of 8 characters or more for the Definer to use with the database account.
 - **Confirm Password.** Reenter the password.

Note: For security reasons, the user should reset the password in his or her Preferences screen.

4. Click **Apply**. The system returns you to the Database Account screen.

Managing Database Account Privileges for Generic Visualization Business Areas

Your company can use the Generic Visualization adapter to integrate an external data visualization tool with Oracle LSH; see the *Oracle Life Sciences Data Hub Adapter Toolkit Guide*. After integration, Definers must create Generic Visualization Business Area instances to allow access to specified sets of data through the visualization tool.

Unlike other Business Areas, which are installed in their Work Area's schema, Oracle LSH installs each Generic Visualization Business Area instance in its own schema outside the Work Area schema. There are simplified security requirements for data in this schema.

Users can log in to the integrated visualization tool using an Oracle LSH database account. The system checks if there is an Oracle LSH user account linked to the database account. If there is a linked user account, the system uses it to determine the user's privileges. If there is no linked user account, the system uses the database account itself to determine the user's privileges.

The database account can have one or two privileges assigned:

- **Read Data**. This privilege allows the user to view data that was never blinded and dummy data in Table instances that are currently blinded. All database accounts that should have access to the Business Area instance data should have this privilege.
- **Read Unblind**. This privilege allows the user to view data that has been permanently unblinded.

If a user should be able to view currently blinded data, he or she must have an Oracle LSH user account with all the required Blind Break privileges and a linked database account.

You can grant (and revoke) Read Data and Read Unblind privileges to database accounts for specific Business Area instances. Oracle LSH audits all changes to these permissions.

1. From the **Security** tab, select the **BA DB Privilege Access** subtab.
2. Enter the **Business Area Instance** and click **Go**. You can also use the Search tool to find those Generic Visualization Business Area instances to which you have "Manage GVA BA Database Access" privileges and select one.

For the selected Business Area instance, you can expand the node for **Read Access** or **Read Unblind Access** to view the database accounts currently assigned that privilege.

3. To change assignments, select the plus (+) icon in the **Manage** column for either Read Access or Read Unblind Access. A screen opens displaying accounts available for assignment and those already selected for the privilege.
4. Double-click or use the arrow icons to grant or revoke the selected privilege to one or more accounts:
 - To grant an account the privilege you selected, move it from **Available Users** to **Selected Users**.
 - To revoke the privilege, move the account from **Selected Users** to **Available Users**.

5. Click **Apply**.

Setting Up TMS Security for Users

Users who will run Oracle LSH APIs that insert, delete, or modify Oracle LSH classification hierarchies and terms (LSH Classification Admin tasks) need security access for their Oracle LSH database account to the Oracle Thesaurus Management System (TMS) instance that is installed as part of Oracle LSH.

Use the script `tmsadduser.sql` to add users to the `TMS_ACCOUNTS` and `OPA_ACCOUNTS` tables with TMS superuser privileges:

1. Open a command window on the TMS Middle tier server.
2. Set `local=database`
3. Change directory to the **install** directory of the TMS 4.6 installation.
4. Log on to your LSH database in SQL*Plus as the `system` account.
5. Run the script to add TMS superusers:
start tmsadduser
6. The system prompts you for the following information:
 - **User ID.** Enter the user's LSH database account ID.
 - **User Password.** Enter the user's LSH database account password.
 - **First Name.** Enter the user's given name.
 - **Last Name.** Enter the user's family name.
 - **TMS Password.** Use the password you entered in when you installed the TMS database; see "Installing Oracle Thesaurus Management System" in the *Oracle Life Sciences Data Hub Installation Guide*.
7. Repeat Step 3 for each user.

Resetting the Password

You can reset the password for any account as necessary; for example, if a user forgets his or her password.

To reset a password for a database account, do the following:

1. Click the icon in the Reset Password column for the account.
2. In the Password field, enter the new password at least 8 characters long.
3. In the Confirm Password, enter the new password again.
4. Click **Apply**.

Note: For security reasons, the user should reset the password in his or her Preferences screen.

Removing a Database Account

You can delete an Oracle LSH database account and its underlying database account; for example, when the user associated with the account leaves the company.

To delete one or more database accounts, do the following:

1. Select the checkbox in the **Select** column for each account you want to delete.
2. Click the **Remove** button. A confirmation message appears.
3. Click **Yes** to continue with the deletion. The system deletes the account and the underlying database account/user schema.

Setting Up Oracle Business Intelligence Visualizations

This chapter contains the following topics:

- [Setting Up a Single-Computer Configuration](#) on page 11-1
- [Setting Up a Multiple-Computer Configuration](#) on page 11-3
- [Set Up Database Connectivity](#) on page 11-6
- [Assign Repository Filename\(s\) to each Oracle BI Presentation Server Instance \(10g Only\)](#) on page 11-6
- [Copy and Edit Revised Execution Command Scripts](#) on page 11-8
- [Set the Default Authenticator to Optional \(11g Only\)](#) on page 11-9
- [Starting the WebLogic Server \(11g Only\)](#) on page 11-9
- [Set Up Oracle Business Intelligence Administration Tool](#) on page 11-10
- [Security Configuration](#) on page 11-10

When used with Oracle Life Sciences Data Hub (Oracle LSH), each OBIEE Subject Area is based on a single Oracle LSH Business Area with Table Descriptors that are mapped to Table instances. The OBIEE Subject Area displays the data contained in these Table instances. The Subject Area's name in OBIEE Answers is populated with the Business Area's name. Each Business Area is linked to an OBIEE Presentation Server. Users can launch OBIEE from inside Oracle LSH or through a URL for the Presentation Server linked to the Business Area.

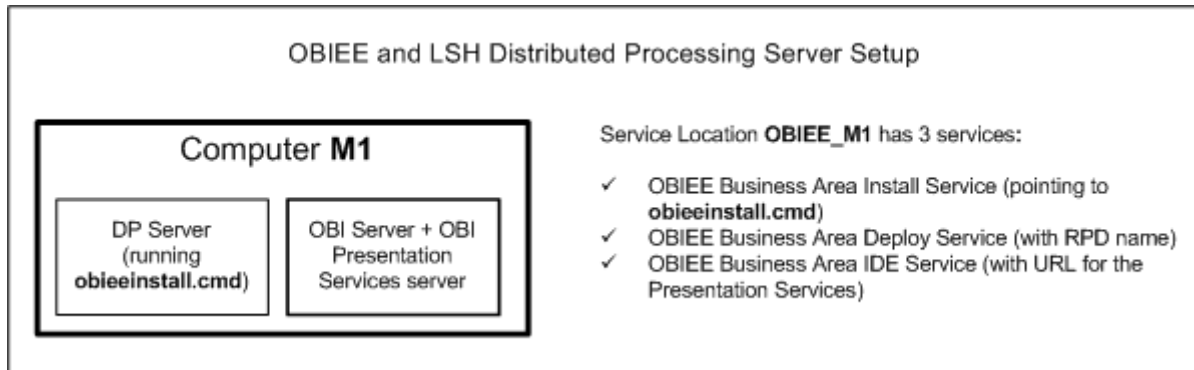
Each Presentation Server can support multiple Business Areas. Oracle LSH automatically merges the OBIEE repository (.rpd) file for each Business Area linked to the same Presentation Server into one large .rpd file.

By default when you install the BI Server, a Presentation Server is also installed on the same computer. You must install the Oracle LSH Distributed Processing (DP) Server on this computer as well. All Business Areas defined in Oracle LSH then point to this Presentation Server. Oracle LSH merges the .rpd files for all Business Areas into a single large .rpd file.

Setting Up a Single-Computer Configuration

In Oracle LSH you must define a service location for the BI Server computer; see ["Defining Service Locations"](#) on page 1-3. [Figure 11-1](#) shows the setup of a single-computer configuration.

Figure 11-1 Single-Computer Setup



You must define all three OBIEE Services for the service location; see "[Defining Services](#)" on page 1-4 for more detailed instructions.

- **OBIEE Business Area Install Service.** In the Execution Command field of this service, enter the path of the `obieeinstall.cmd` script; for example: `C:\LSH_DP_SERVER\obiee\obieeinstall.cmd`.

Note: If you are using OBIEE 10g, the file names have "10g" appended: `obieeinstall10g.cmd` and `obieedeploy10g.cmd`

Note: During installation this script should have been copied into the DP Server Home directory or a subdirectory of it.

- **OBIEE Business Area Deploy Service.** In the Execution Command field of this service, enter the path of the `obieedeploy.cmd` script; for example: `C:\LSH_DP_SERVER\obiee\obieedeploy.cmd`.

In the Details field, enter database connection information and the name of the `.rpd` file you want to use on the Oracle BI Presentation Server; for example: `dsn=abc123,callinterface=OCI10G,RPD=LSH_MASTER`. If you are using OBIEE 11g, add `ver=11g`.
- **OBIEE Business Area IDE Service.** In the Execution Command field, enter the URL of the Oracle BI Presentation Server; for example:
`http://svr:port/analytics/saw.dll?Answers`.

Note: For the OBIEE IDE Service, specify a number of **Service Instances** equal to the maximum number of users you expect to query the repository at the same time, plus a buffer; see "[Service Instances](#)" on page 1-9.

If you do not create enough service instances, users may lose access to the database in the middle of a task. For example, if you define 10 service instances and 11 people log in to the same OBIEE URL, when the last person logs in, the system looks for a session that has been running longer than the defined Reuse Wait time for the service. If there is such a session, the system recycles it even if the user is active, and the original user loses his or her connection to the database.

It is better to solve this problem by defining a large number of service instances than a high reuse wait time because the same user cannot open a new session—for example, if he or she wants to select different launch settings—until his or her original session reaches the reuse wait time.

Setting Up a Multiple-Computer Configuration

This section contains the following topics:

- [Service Locations and Services Required in a Multiple Computer Installation](#) on page 11-4
- [Single and Multiple Repository Files](#) on page 11-6

If you want to install the Oracle BI Presentation Server on a UNIX machine, or on an additional Windows machine, you can install just the Oracle BI Presentation Server and the Oracle LSH DP Server on the second machine, without the Oracle BI Server as shown in [Figure 11-2](#).

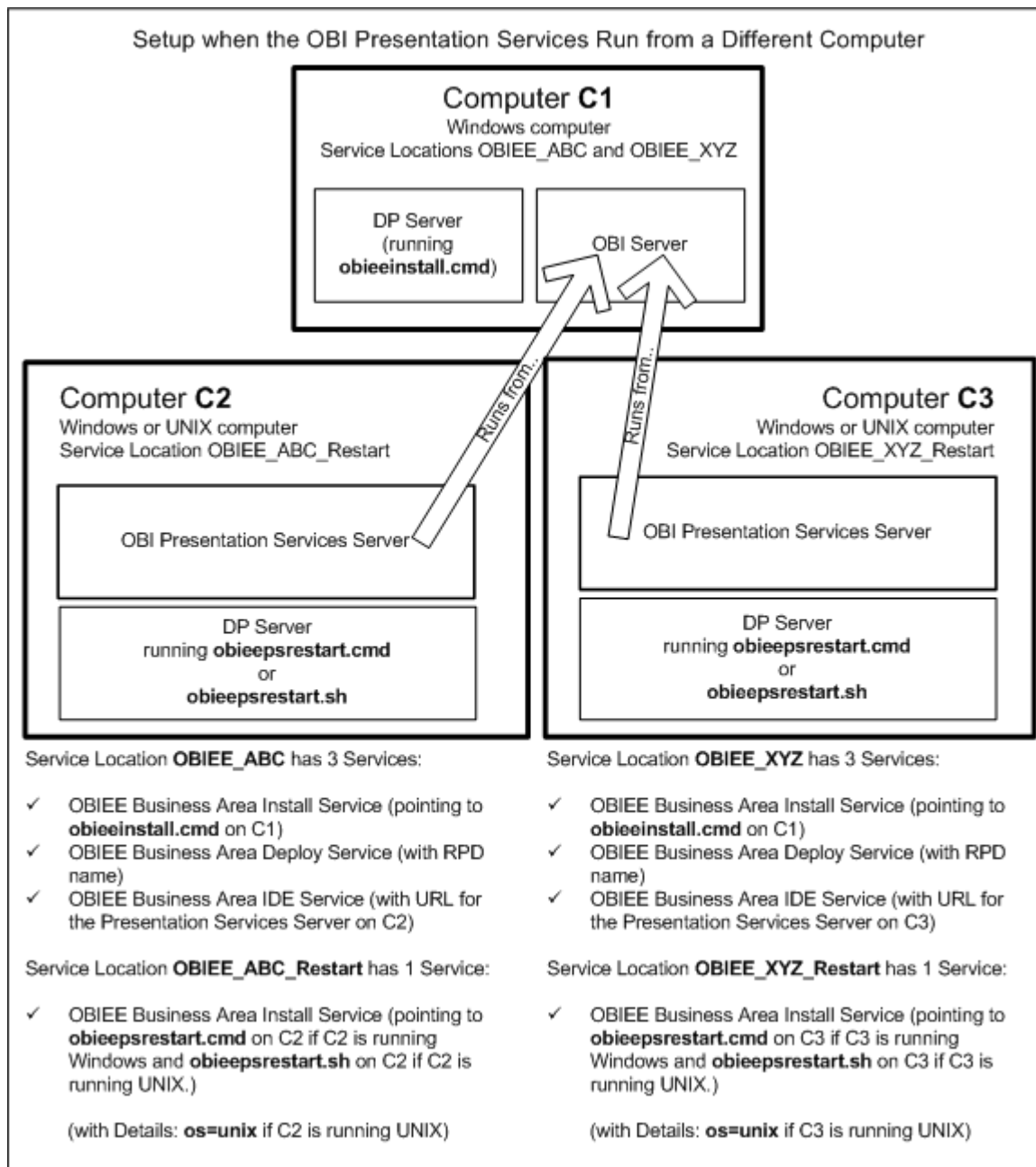
You can choose to have either a single, merged .rpd file or a different one for each Presentation Service instance. In either case, multiple Business Areas can use the same Presentation Server.

Define a pair of service locations for each Presentation Server computer: one on the Presentation Services computer ending in `_RESTART` and one on the BI Server computer with the same name except without the `_RESTART` string; see "[Defining Service Locations](#)" on page 1-3.

Edit the DP Server Start script on each computer with the exact name you give the service location. Instructions are in the *Oracle Life Sciences Data Hub Installation Guide*.

Note: Do not use spaces, slashes, or special characters such as single quotes (') in the service location name. This may cause problems when you define a remote location and connection to password-protect the .rpd file.

Figure 11–2 Multiple-Computer Setup



Service Locations and Services Required in a Multiple Computer Installation

In Oracle LSH you must define a pair of service locations, one with services running on the BI Server computer and the other with services running on the Presentation Services computer, for each separate Presentation Services computer.

Service Location Name The name of the service location for the Presentation Services computer must be the same as the corresponding service location for the BI Server plus the string `_RESTART`.

Note: Do not use spaces, slashes, or special characters such as single quotes (') in the service location name. This may cause problems when you define a remote location and connection to password-protect the .rpd file.

When Definers create a new OBIEE Business Area, they must select a service location to use. The drop-down list displays all OBIEE service locations that do not end in _RESTART.

For example, as shown in [Figure 11-2](#), the service location and service requirements when the BI Server and Presentation Server are on different computers are:

- Give the first service location, which must be for a Windows computer, a name such as OBIEE_ABC and define all three OBIEE services for it:
 - **OBIEE Business Area Install Service.** In the Execution Command field of this service, enter the complete directory path and filename for the Windows execution command script on the BI Server computer; for example : C:\LSH_DP_SERVER\obiee\obieeinstall10g.cmd.
 - **OBIEE Business Area Deploy Service.** In the Execution Command field of this service, enter the complete directory path and filename for the Windows execution command script on the BI Server computer; for example : C:\LSH_DP_SERVER\obiee\obieeinstall10g.cmd.

In the Details field, enter database connection information and the name of the .rpd file you have defined for the Oracle BI Presentation Server; for example: dsn=ilpcr211,callinterface=OCI10G,RPD=Study02

- **OBIEE Business Area IDE Service.** In the Execution Command field of this service, enter the Presentation Services URL, for example: http://C2:9704/analytics/saw.dll?Answers
- Give the second service location, which can be for either a Windows or UNIX computer, a name that is the same as the first service location plus the string _RESTART; for example, OBIEE_ABC_RESTART. Define one service for this service location: the **OBIEE Business Area Install Service**.
 - In the Execution Command field of this service, enter the complete directory path of the correct restart (not install) execution command script for the operating system; for example, on Windows:


```
C:\LSH_DP_SERVER\obiee\obieerestart10g.cmd
```

 or on UNIX:


```
/slot/ems1/appmgr/lshdpserver/obiee/obieepsrestart10g.sh
```
 - If the Presentation Server is a UNIX computer, enter os=unix in the Details field of this service.

Note: In a multiple-computer configuration you can also use the Presentation Server that is installed by default on the same computer as the BI Server. In this case you need one more service location defined the same way as for a single-computer configuration; see ["Setting Up a Single-Computer Configuration"](#) on page 11-1.

Single and Multiple Repository Files

OBIEE repository (.rpd) files reside on the BI Server. The Business Area meta-data and OBIEE Administration Tool logic for all Business Areas that use a particular Presentation Server are merged into a single .rpd file. If you have multiple Presentation Servers, you can choose to have a different .rpd file for each one or a single, merged file. Functionally there is no difference whether you choose to have a single .rpd file or a different one for each Presentation Server.

- To use a single, merged .rpd file, enter the same .rpd file name in the definition of the Deploy service of each pair.
- To have a separate .rpd file for each Presentation Server, enter a different .rpd file name in the definition of the Deploy service of each pair.

Note: You must add an entry to the NQSCONFIG.INI file on the BI Server for each repository file and set up the Analytics Web DSN to use the repository file; see ["Assign Repository Filename\(s\) to each Oracle BI Presentation Server Instance \(10g Only\)"](#) on page 11-6.

Set Up Database Connectivity

To run Oracle Business Intelligence visualizations through Oracle LSH, you must set up connectivity between the Oracle LSH database and the BI Server database. You can use a system ODBC DSN or the Oracle Call Interface (OCI) to connect the Oracle BI Server to the Oracle LSH database server.

Set Up Database Connectivity Using a System DSN Create a valid Oracle driver-based System Data Source Name (DSN) on the computer where the Oracle Business Intelligence server is installed.

Consult Microsoft Windows online help for instructions on creating the System DSN. Use the Test Connection button to test the connection.

Set Up Database Connectivity Using OCI 10g/11g Create an entry in the BI Server's tnsnames.ora file with the name and connection details for the LSH database server.

Test the connection by trying to connect to the Oracle LSH database using SQL*Plus as apps.

Note: You must have an entry for the Oracle LSH database in the tnsnames.ora file in your Oracle Home.

Assign Repository Filename(s) to each Oracle BI Presentation Server Instance (10g Only)

Do the following on each computer with a Presentation Server instance installed:

- [Add an Entry to the NQSCONFIG File for Each Repository File](#) on page 11-7
- [Set up the Analytics Web DSN to Use the Repository File](#) on page 11-7

Add an Entry to the NQSConfig File for Each Repository File

Each time the BI Server is restarted, all the .rpd files defined in NQSConfig.ini are loaded and one of them is defaulted. If an .rpd file is not defined in NQSConfig.ini, it is not available for visualizations.

If you are using a single-computer configuration, you have only one .rpd file. If you are using a multi-computer configuration, you can choose to use one .rpd file for all Presentation Servers or a different .rpd file for each one by entering the same or different value in the Details field of the Deploy service for the relevant OBIEE service location; see the chapter on Oracle Business Intelligence visualizations in the *Oracle Life Sciences Data Hub System Administrator's Guide*. This value determines which .rpd file is used for which Presentation Server.

On the Oracle BI server computer, open the NQSConfig.ini file and make an entry for each repository file under the Repository section that includes both a logical name and the .rpd file name. You must designate one .rpd file as the default, as shown in the following example. Oracle recommends setting the .rpd for the Presentation Server installed on the BI Server as the default. For example:

```
REPOSITORY
logical_name1 = RPD_name1.rpd, DEFAULT;
logical_name2 = RPD_name2.rpd
```

Each .rpd file name must exactly match the .rpd file name defined in the Details field of the Deploy service for one of the the OBIEE service locations.

Note: Neither the logical name nor the .rpd file name is displayed in OBIEE Answers or in Oracle LSH. The Subject Area's name in OBIEE Answers is populated with the Business Area's name.

Set up the Analytics Web DSN to Use the Repository File

On each computer with the Presentation Server installed, set up the Analytics Web DSN to use the Oracle LSH Repository file.

The Presentation Server will try to find its .rpd file during this process, and setup will not succeed if it does not find an .rpd file with the same name that you entered in the NQSConfig.ini file on the Oracle BI server computer and in the Details field of the service location Deploy service definition.

1. Do one of the following:
 - Define and install a Business Area in Oracle LSH; see the *Oracle Life Sciences Data Hub Application Developer's Guide*. This deploys a default .rpd file for the Business Area generated in Oracle LSH.
 - Put a dummy .rpd file in the Server\Repository folder under the Oracle BI installation folder (for example, D:\OracleBI\Server\Repository). You can rename and use the file paint.rpd that comes with OBIEE.
2. In Windows, go to Control Panel, then Administrative Tools, then Data Sources (ODBC), then System DSN, and open the Analytics Web DSN Configuration screen.
3. Enter the name of the Oracle BI Server computer in the Server field of the Analytics Web DSN configuration screen. Click **Next**.

4. Check **Change the default repository to** and enter the logical name of the .rpd file that you set as the default in the NQSConfig.ini file on the Oracle BI server computer.
5. Check **Connect to Oracle BI Server to obtain default settings for the additional configuration options** and supply the Oracle BI Server login ID and password.
6. Click **Next**. The Presentation Server tries to connect to the BI Server and looks for the .rpd file.
7. If the connection succeeds, you can see the name of the Oracle LSH Business Area instance under the Database column. Click **Finish**.

Copy and Edit Revised Execution Command Scripts

You must add local information to the execution command scripts used by the Deploy and Install services, obieedeploy.cmd and obieeinstall.cmd.

Note: If you are using OBIEE 11.1.1.5.0, follow these instructions exactly.

If you are installing OBIEE 10.1.3.4.1 for the first time, use the scripts with the same names but with "10g" appended: obieedeploy10g.cmd and obieeinstall10g.cmd.

If you have already been using OBIEE 10.1.3.4.1 and have upgraded to Oracle LSH 2.2, you do not need to change the scripts you have already edited and used. Only the name has changed.

1. Copy obieedeploy.cmd and obieeinstall.cmd from \$cdr/admin/templates to the Oracle LSH Distributed Processing Server location.
2. Edit obieedeploy.cmd, which has a RPD_DIR environment variable that indicates the path where the LSH master RPD will reside. Unlike OBIEE 10g, in the OBIEE 11g integration, this can be any folder path on the BI Server machine's file system.

```
set RPD_DIR=absolute_path
```

For example:

```
set RPD_DIR=C:\RPD
```

3. Edit obieeinstall.cmd and obieedeploy.cmd to provide location values for environment variables as follows:

- **Path** Enter the absolute path to the BI Server's bin folder:

```
set PATH=absolute_path_to\Oracle_BI1\bifoundation\server\bin
```

- **BI Application** You must enter the exact string shown below:

```
set ORACLE_BI_APPLICATION=coreapplication
```

- **Oracle BI Instance** Enter the path to the BI Server instance:

```
set ORACLE_BI_INSTANCE=drive:\Oracle\fmw\instances\instance1
```

Note: `instance1` is the required name of the first BI Server instance. Normally you need only one BI Server instance to use OBIEE with Oracle LSH.

- **Component Name** The name of the Presentation Server must be the exact string below:

```
set COMPONENT_NAME=coreapplication_obis1
```

- **Component Type** Enter `:OracleServerComponent` (required).

```
set COMPONENT_TYPE=OracleServerComponent
```

- **Oracle Instance** This value must be the same as the Oracle BI Instance.

```
set ORACLE_INSTANCE=drive:\Oracle\fmw\instances\instance1
```

Set the Default Authenticator to Optional (11g Only)

OBIEE 11g has a `DefaultAuthenticator` provider to handle authentication. Since LSH RPDs have their own SQL authentication block configured, the `DefaultAuthenticator` should be made `OPTIONAL` in OBIEE 11g.

After the master RPD has been successfully deployed and the BI Server services successfully restarted, do the following:

1. Open the Weblogic Server Administration Console using the URL specific to your environment.

Note: If the URL does not work, you may need to start the WebLogic Server; see "[Starting the WebLogic Server \(11g Only\)](#)" on page 11-9.

2. On the Home page, under Your Application's Security Settings, click **Security Realms**.
3. Under Change Center in the upper left, click **Lock & Edit**.
4. Under How Do I... click **Configure a New Security Realm** and follow instructions on screen.
5. Click the name of the new security realm. The Settings for the security realm are displayed.
6. In the top row of tabs, click the **Providers** tab.
7. If it is not already selected, click the **Authentication** subtab.
8. Click **DefaultAuthenticator**.
9. Set **Control Flag** to `OPTIONAL` and Save.

Starting the WebLogic Server (11g Only)

If the URL for the WebLogic Administration Console is not working, the WebLogic Server may be down. To start it:

1. On the BI Server computer, right-click Command Prompt under the Start menu and select Run as Administrator.

2. Change directory to the Domain folder under the OBIEE installation folder; for example, e:\oracle\fmw\user_projects\domains\bifoundation_domain.
3. Run the command startWeblogic.cmd under this folder. The system prompts you for the WebLogic Server username and password.
4. Check that the command window displays the message "Server started in RUNNING mode."

Set Up Oracle Business Intelligence Administration Tool

To use the Oracle Business Intelligence Administration Tool, each Definer must do the following on his or her local PC:

- Install Oracle Business Intelligence Client Tools in the location specified by the system administrator; this location must be specified in **cdrconfig.xml**; see the section on installing the client plug-in in the *Oracle Life Sciences Data Hub Installation Guide*.
- Insert the CD-ROM supplied by the system administrator.
- Ensure that **cdrconfig.xml** has the correct directory path for the Oracle BI Administration tool.
- In the Oracle BI Administration Tool, set up connectivity with the Oracle LSH database. See "[Set Up Database Connectivity](#)" on page 11-6 for information.
- Add and set the following environment variables on the PC through advanced system properties:
 - **ORACLE_HOME** Enter the absolute path of the Oracle Home on your PC; for example:

```
C:\Program Files\Oracle Business Intelligence Enterprise Edition Plus Client\oraclebi\orahome
```
 - **ORACLE_INSTANCE** Enter the absolute path to Oracle BI orainst1 on your PC; for example:

```
C:\Program Files\Oracle Business Intelligence Enterprise Edition Plus Client\oraclebi\orainst
```
 - **ORACLE_BI_APPLICATION** You must enter the following value exactly:

```
coreapplication
```

Security Configuration

This section contains the following topics:

- [Data Security](#) on page 11-10
- [Creating an OBIEE Remote Location and Connection for RPD Password Security](#) on page 11-11
- [Using Separate URLs to Control Access](#) on page 11-12

Data Security

Security access to data is controlled in Oracle LSH: people with Oracle LSH user accounts that have access to the Business Area can see data in the corresponding Subject Area in OBIEE Answers. No OBIEE user accounts are required. Users with the

required privileges can see blinded and noncurrent data if they launch OBIEE Answers from within Oracle LSH and request to see such data.

When users launch OBIEE Answers from outside Oracle LSH, through the URL of the OBIEE Presentation Server, they cannot see blinded or noncurrent data even if they have the required privileges within Oracle LSH.

Creating an OBIEE Remote Location and Connection for RPD Password Security

You must define a remote location and connection for each non-restart OBIEE service location in order to provide password protection for the deployed .rpd file in the Administration Tool.

When the Definer installs an OBIEE Business Area in Oracle LSH, the deploy script deploys the latest version of the .rpd file to the BI Server. During the deployment, Oracle LSH passes a user name and password to the deployment script if a remote location and connection are defined for the service location.

When the Definer works on the .rpd file in the Administration Tool, he or she must log in as `administrator` and enter the password you specify in the remote connection definition. You can change this password at any time.

If you do not define a remote location and connection for the service location, no password is required to work on the .rpd file in the Administration Tool.

Create an OBIEE Remote Location

To define a Remote Location:

1. Click the **Remote Location** subtab under the Administration tab. The Maintain Remote Locations screen opens.
2. Click **Add Remote Location**. The Create Remote Location screen appears.
3. Enter values in the following fields:
 - **Name**. Enter the exact same name you have given for the service location.
 - **Description**. Not required.
 - **DBLINK Prefix**. Enter any value. The system does not use this value.
 - **Connect String**. Enter any value. The system does not use this value.
 - **Adapter**. Select OBIEE from the drop-down list.
 - **Conversion Multiplier**. Do not enter a value.
4. **Classification**: The system does not use these values.
5. Click **Apply** to save your work. The system opens the main screen for the new Remote Location.

By default, the Remote Location inherits the user group assignments of the Adapter Area.

Create an OBIEE Remote Location Connection

For each OBIEE Remote Location, create a Remote Connection.

To define a Connection:

1. In the main screen for the Remote Location for which you want to create a Connection, click **Create Connection**. The Create Connection screen opens.
2. Enter values in the following fields:

- **Name.** Enter administrator. (Not case-sensitive)
 - **Description.** Not required.
 - **User Name.** Enter administrator.
 - **Password.** Enter the password you want to require for the .rpd file. Oracle LSH encrypts the password for security.
 - **Connection Type.** The system does not use this value.
 - **Remote Location.** The system populates the field with the name of the Remote Location for which you are defining this Connection.
3. **Classification:** The system does not use these values.
 4. Click **Apply** to save your work. The system displays the main screen for the new Connection.

Using Separate URLs to Control Access

Each Presentation Server must be installed on a different computer and has a different URL. You can use this setup to control what users can see in OBIEE. For example:

- If you have only one Business Area using a particular Presentation Server, users launching OBIEE with that URL can see only the Subject Area that corresponds to that Business Area. You can include the Subject Area in the URL so that users go directly to the Subject Area page; for example:
`http://srv:port/analytics/saw.dll?Answers&SubjectArea=X`
- You can organize your Business Areas so that only related Business Areas use the same Presentation Server; for example, all studies in a single project have a Business Area, and there is another Business Area for the project as a whole. Users launching OBIEE with that URL can see only the Subject Areas related to that project, and can see data only for the Subject Areas that correspond to Business Areas on which they have View privileges.

Setting Up Security for Oracle Business Intelligence Publisher

This chapter contains the following topics:

- [Overview](#) on page 12-1
- [Creating and Assigning Oracle Applications Responsibilities](#) on page 12-2
- [Configuring Security in Oracle BI Publisher](#) on page 12-4
- [Storing the BI Publisher Local Superuser Credentials in Oracle Life Sciences Data Hub](#) on page 12-6

Overview

Oracle LSH Definers can create Oracle LSH Programs of the BI Publisher adapter type to create reports for Oracle LSH Consumers. See "Defining Programs" in the *Oracle Life Sciences Data Hub Application Developer's Guide* for more details.

To integrate Oracle LSH with Oracle BI Publisher, you must do the following:

- **In Oracle Applications:** Create a user-specific responsibility for each Oracle LSH user who will create or use Oracle LSH BI Publisher Programs. This responsibility allows Oracle LSH users access to their database schema from Oracle BI Publisher. See "[Creating User-Specific Responsibilities in E-Business Suite](#)" on page 12-2.
- **In Oracle BI Publisher:**
 - Create a Local Superuser account for use with Oracle LSH. See "[Setting Up the Security Model and Creating the Local Superuser Account in Oracle BI Publisher](#)" on page 12-4.
 - Create a Data Source with the database account credentials of each Oracle LSH user who wants to create or use Oracle LSH BI Publisher Programs. You must assign the user-specific E-Business Suite responsibility to the Oracle BI Publisher Data Source as well as the Oracle LSH user. See "[Creating Data Sources and Assigning them User-specific Roles](#)" on page 12-5 and "[Assigning Application Roles in Oracle LSH](#)" on page 12-3.
 - Create a Shared Folder with the name SYSTEM. See "[Creating the SYSTEM Folder](#)" on page 12-5.
- **In Oracle LSH:**
 - Assign one or more of the XMLP application roles to Oracle LSH Definers. See "[Assigning Application Roles in Oracle LSH](#)" on page 12-3. Oracle BI Publisher was formerly XML Publisher.

Note: Oracle LSH application roles are known as *responsibilities* in Oracle Applications and *functional roles* in Oracle BI Publisher.

- Store the BI Publisher Local Superuser account's username and password in Oracle LSH in a Remote Connection. See "[Storing the BI Publisher Local Superuser Credentials in Oracle Life Sciences Data Hub](#)" on page 12-6.

Creating and Assigning Oracle Applications Responsibilities

This section contains the following topics:

- [Creating User-Specific Responsibilities in E-Business Suite](#) on page 12-2
- [Assigning Application Roles in Oracle LSH](#) on page 12-3

Creating User-Specific Responsibilities in E-Business Suite

You must create an E-Business Suite responsibility for each Oracle LSH user who wants to use Oracle BI Publisher from Oracle LSH. This is required for database access to Oracle LSH users from BI Publisher.

Note: Such Oracle LSH users (Definers as well as Consumers) must also have a corresponding database account. See "[Creating Database Accounts](#)" on page 10-2 for instructions on creating one.

To create the E-Business Suite responsibility, do the following:

Note: For complete information, see the *Oracle® E-Business Suite System Administrator's Guide - Security* at http://download.oracle.com/docs/cd/B53825_08/current/acrobat/121sasg.pdf.

1. Go to your Oracle LSH URL.
2. Log in as sysadmin.
3. Select **System Administrator** (*not System Administration*) from the list of responsibilities in the left-hand column under Navigator. The system refreshes the page and adds a column of links on the right.
4. Under **Security: Responsibility**, select **Define**.
5. The Oracle Applications Forms interface starts. If prompted to accept an applet, click **Grant** or **Grant Always**.
6. In the Responsibilities Form as shown in [Figure 12-1](#), enter the following mandatory values:
 - **Responsibility Name.** Enter a name in this format: BIP_DATASRC_ROLE_LSH_application_username.
 - **Application.** Enter or select Oracle Life Sciences Data Hub.
 - **Responsibility Key.** Same as Responsibility Name but note that there is a 30 character limit for the Responsibility Key.

- **Data Group.**
 - **Name.** Enter or select Standard.
 - **Application.** Enter or select Oracle Life Sciences Data Hub.
- **Menu.** Enter CDR BIP Menu.

Note: You must create the E-Business Suite Responsibilities for each Oracle LSH user who wants to create or use Oracle LSH BI Publisher Programs. Repeat the above steps for each such Oracle LSH user.

Figure 12–1 The Oracle Applications Responsibilities Form Showing a Sample Oracle BI Publisher Role Created for an Oracle LSH Definer

The screenshot shows the Oracle Applications Responsibilities form for user ip1dv102. The form is titled 'Responsibilities' and contains the following fields and sections:

- Responsibility Name:** BIP_DataSrc_Role_john.doe@oracle.
- Application:** Oracle Life Sciences Data Hub
- Responsibility Key:** BIP_DATASRC_ROLE_JOHN.DOE
- Description:** (Empty)
- Effective Dates:** From 28-NOV-2008, To (Empty)
- Available From:**
 - Oracle Applications
 - Oracle Self Service Web Applications
 - Oracle Mobile Applications
- Data Group:**
 - Name:** Standard
 - Application:** Oracle Life Sciences Data Hub
- Menu:** CDR BIP MENU
- Web Host Name:** (Empty)
- Web Agent Name:** (Empty)
- Request Group:**
 - Name:** (Empty)
 - Application:** (Empty)
- Menu Exclusions:** (Empty table with columns Type, Name, Description)

Assigning Application Roles in Oracle LSH

The following application roles are available in Oracle LSH that are specific to Oracle BI Publisher:

Note: Read the *Oracle® Business Intelligence Publisher Administrator's and Developer's Guide* at http://download.oracle.com/docs/cd/E12844_01/doc/bip.1013/e12188.pdf for more information on these roles.

- **XMLP Administrator.** This is the administrator role for the Oracle BI Publisher server. A user with this role can create and delete reports, folders, data sources, and roles in Oracle BI Publisher.

- **XMLP Developer.** This role allows users to build reports in the system.
- **XMLP Scheduler.** This role allows users to schedule reports.
- **XMLP Analyzer Excel.** This role allows users to use the Excel Analyzer feature.
- **XMLP Analyzer Online.** This role allows users to use the online analysis feature.
- **XMLP Template Designer.** This role allows users to connect to the Oracle BI Publisher server from the Template Builder and to upload and download templates.

Besides the regular Oracle LSH application roles, you must also assign one or more of the XMLP roles and the role created specifically for Oracle BI Publisher to Oracle LSH users. See "[Assigning Application Roles](#)" on page 9-5 for instructions on assigning application roles to Oracle LSH users.

Configuring Security in Oracle BI Publisher

This section contains the following topics:

- [Setting Up the Security Model and Creating the Local Superuser Account in Oracle BI Publisher](#) on page 12-4
- [Creating Data Sources and Assigning them User-specific Roles](#) on page 12-5
- [Creating the SYSTEM Folder](#) on page 12-5

You must perform the following tasks in Oracle BI Publisher to integrate it with Oracle LSH. For detailed instructions, refer to the *Oracle® Business Intelligence Publisher Administrator's and Developer's Guide* at

http://download.oracle.com/docs/cd/E12844_01/doc/bip.1013/e12188.pdf.

For your convenience, the following sections contain excerpts from Oracle BI Publisher documentation, slightly modified to include specific information related to Oracle LSH.

Setting Up the Security Model and Creating the Local Superuser Account in Oracle BI Publisher

Do the following in Oracle BI Publisher:

1. Log in to Oracle BI Publisher Enterprise as an administrator.
2. From the Admin tab, select **Security Configuration**.
3. In the Security Model section of the page, select **Oracle E-Business Suite** from the list.
4. Load the dbc file from the Oracle LSH instance. This is typically located under the \$FND_SECURE directory. If you do not have access to this file, contact your Oracle LSH system administrator.
5. Select the **Enable Local Superuser** check box and enter a username and password under the Local Superuser section of the Security Configuration tab. You need to store the Local Superuser credentials in a Remote Connection created specifically for BI Publisher, in Oracle LSH. See "[Storing the BI Publisher Local Superuser Credentials in Oracle Life Sciences Data Hub](#)" on page 12-6.
6. Restart the Oracle BI Publisher server for the security changes to take effect.

Creating Data Sources and Assigning them User-specific Roles

Do the following in Oracle BI Publisher:

1. Log in as the Local Superuser.
2. From the Admin page select JDBC Connection. This will display the list of existing JDBC connections.
3. Select the **Add Data Source** button.
4. Enter the following fields for the new connection:
 - **Data Source Name.** Enter `LSH_DataSrc_LSH_application_username`.
 - **Driver Type.** Select Oracle 9i/10g/11g.
 - **Connection String.** Enter the database connect string.
For example:
`jdbc:oracle:thin@myserver.mycompany.com:port:prod`
 - **User Name.** Enter the Oracle LSH database user credentials that correspond to the Oracle LSH application user account. See "[Creating Database Accounts](#)" on page 10-2.
 - **Password.** Enter the Oracle LSH database user password that corresponds to the Oracle LSH application user account.
 - **Use Proxy Authentication.** Do not select this check box.
5. Click **Test Connection**. If the test is successful, the confirmation message, "Connection established successfully" appears. If connection error occurs, the message "Could not establish connection," appears.
6. Define security for this data source. Use the shuttle buttons to move the corresponding `BIP_DataSrc_Role_LSH_application_username` role from the Available Roles list to the Allowed Roles list.

Note: See "[Creating User-Specific Responsibilities in E-Business Suite](#)" on page 12-2 for instructions on creating the `BIP_DataSrc_Role_LSH_application_username` role for each Oracle LSH user who needs access to Oracle BI Publisher. You must replace `LSH_application_username` with the actual Oracle LSH application user's username.

Repeat the above steps for each Oracle LSH user who will create or use Oracle LSH BI Publisher Programs.

Creating the SYSTEM Folder

Create a Shared Folder named `SYSTEM` in Oracle BI Publisher. Oracle LSH uses this folder to execute Oracle LSH BI Publisher Programs. Only a user with `XMLP_ADMIN` privileges can see this folder.

Note: Folder names are case sensitive in Oracle BI Publisher. Make sure the name of the folder is `SYSTEM`, in all capital letters.

Storing the BI Publisher Local Superuser Credentials in Oracle Life Sciences Data Hub

This section contains the following topics:

- [Creating the Oracle BI Publisher Remote Location in Oracle LSH](#) on page 12-6
- [Creating the Oracle BI Publisher Remote Connection in Oracle LSH](#) on page 12-6
- [The LSH: BIP Endpoint Profile](#) on page 12-7

You must store the Oracle BI Publisher Local Superuser account details in an Oracle LSH Remote Connection. See [Chapter 5, "Registering Locations and Connections"](#) for details on Oracle LSH Remote Connections.

Creating the Oracle BI Publisher Remote Location in Oracle LSH

To create the Remote Location in Oracle LSH, do the following:

1. Click the **Remote Location** subtab under the Administration tab. The Maintain Remote Locations screen opens.
2. Click **Add Remote Location**. The Create Remote Location screen appears.
3. Enter values in the following fields:

Note: Enter the values exactly as specified below, else the Oracle LSH BI Publisher Program will not run.

- **Name.** Enter BIPLOCATION.
 - **DBLINK Prefix.** Enter Dummy.
 - **Connect String.** Enter Dummy.
 - **Adapter.** Select BI Publisher.
4. Click **Apply** to save your work. The system opens the main screen for the new Remote Location.

Creating the Oracle BI Publisher Remote Connection in Oracle LSH

To create the Remote Connection in Oracle LSH, do the following:

1. In the main screen for the Remote Location BIPLocation. The Connection Maintenance screen opens.
2. Click **Create Connection**. The Create Connection screen appears.
3. Enter values in the following fields:

Note: Enter the values exactly as specified below, else the Oracle LSH BI Publisher Program will not run.

- **Name.** Enter BIPConnection.
- **Description.** Enter the details of the Oracle LSH database server in this field in the following format:

`jdbc:oracle:thin:@server:port:SID`

For example: `jdbc:oracle:thin:@srv123.example.com:1234:svr456`

Oracle BI Publisher needs these details to connect back to Oracle LSH.

- **User Name.** Enter the Oracle BI Publisher Local Superuser account's username.
 - **Password.** Enter the Oracle BI Publisher Local Superuser account's password.
 - **Connection Type.** Select `Shared`.
 - **Remote Location.** The system populates the field with the name of the Remote Location.
4. Click **Apply** to save your work. The system displays the main screen for the new Connection.

The LSH: BIP Endpoint Profile

Oracle LSH automatically creates a profile in Oracle Applications, and stores an Oracle BI Publisher webservice namespace value in it. See "[Oracle LSH: BIP Endpoint](#)" on page 3-11 for details. Do not edit this profile because the system uses it to interact with Oracle BI Publisher.

Setting Up Security for Informatica

This section contains the following topics:

- [Setting Up User Accounts](#) on page 13-1
- [Creating an Informatica Remote Location and Connection](#) on page 13-2

You need to set up the following user accounts to integrate Oracle LSH with Informatica:

Setting Up User Accounts

You need the following accounts to integrate Informatica with Oracle LSH:

- [Informatica Administrator Account](#) on page 13-1
- [Accounts for Each Oracle LSH Definer](#) on page 13-1

Informatica Administrator Account

You must create an Informatica administrator account with the name `LSHAdmin`. This Informatica administrative user owns all the security groups and workflows created in association with Oracle LSH.

See the appropriate Informatica documentation for instructions on setting up user accounts in Informatica.

Note: The administrator account name must be exactly `LSHAdmin`, including upper- and lowercase as written.

Accounts for Each Oracle LSH Definer

To work with Informatica, each Oracle LSH Definer needs the following user accounts:

- An Oracle LSH application user account and a corresponding Oracle LSH database user account. See "[Setting Up User Accounts](#)" on page 9-3 and "[Creating Database Accounts](#)" on page 10-2.
- An Informatica user account with the same username as the Oracle LSH application user account. See the appropriate Informatica documentation for instructions on setting up user accounts in Informatica.

You must create Informatica user accounts with **all uppercase** letters for use with Oracle LSH (except for the `LSHAdmin` account).

This is because Oracle LSH application and database user accounts are maintained in Oracle FND, which stores user names in uppercase though it accepts mixed case

at login. In Informatica, user names are case sensitive. When a Definer launches Informatica from Oracle LSH, the system performs a security check that matches the user's Informatica and Oracle LSH user names. If the Informatica user name is not in uppercase, the security check fails and the Definer cannot launch Informatica.

Note: The LSHAdmin account is not an FND account and must be entered in upper- and lowercase as shown.

Creating an Informatica Remote Location and Connection

This section contains the following topics:

- [Creating the Informatica Remote Location in Oracle LSH](#) on page 13-2
- [Creating the Informatica Remote Connection in Oracle LSH](#) on page 13-3

You must store the Informatica administrative user LSHAdmin in Oracle LSH as a Remote Connection. See [Chapter 5, "Registering Locations and Connections"](#) for details on Oracle LSH Remote Connections.

Creating the Informatica Remote Location in Oracle LSH

To create the Remote Location in Oracle LSH, do the following:

1. Click the **Remote Location** subtab under the Administration tab. The Maintain Remote Locations screen opens.
2. Click **Add Remote Location**. The Create Remote Location screen appears.
3. Enter values in the following fields:

Note: Enter the values exactly as specified below, else the Oracle LSH Informatica Program will not run.

- **Name.** Enter INFORMATICAADMIN
 - **DBLINK Prefix.** Enter INFORMATICAADMIN.
 - **Connect String.** Enter INFORMATICAADMIN.
 - **Adapter.** Select Informatica.
4. In the **Classification** section, select the following for both the definition and the instance:
 - **Subtype.** Select a subtype according to your company's policies.
 - **Classification Values.** Specify classifications according to your company's policies, to enable users to search for the Remote Location by classification. For instructions, see "Classifying Objects and Outputs" in the *Oracle Life Sciences Data Hub Application Developer's Guide*.
 5. Click **Apply** to save your work. The system opens the main screen for the new Remote Location.

You can change the security and classifications of the Remote Location through the items in the **Actions** drop-down list. By default, the Remote Location inherits the user group assignments of the Adapter Area.

Creating the Informatica Remote Connection in Oracle LSH

To create the Remote Connection in Oracle LSH, do the following:

1. In the main screen for the Remote Location `INFORMATICAADMIN` click **Create Connection**. The Create Connection screen opens.
2. Enter values in the following fields:

Note: Enter the values exactly as specified below, else the Oracle LSH Informatica Program will not run.

- **Name.** Enter `LSHADMIN`.
 - **User Name.** Enter `LSHAdmin`.
 - **Password.** Enter the password for the Informatica `LSHAdmin` user.
 - **Connection Type.** Select `Shared`.
 - **Remote Location.** The system populates the field with the name of the Remote Location.
3. In the **Classification** section, select the following for both the definition and the instance:
 - **Subtype.** Select a subtype according to your company's policies.
 - **Classification Values.** Specify classifications according to your company's policies, to enable users to search for the Connection by classification. For instructions, see "Classifying Objects and Outputs" in the *Oracle Life Sciences Data Hub Application Developer's Guide*.
 4. Click **Apply** to save your work. The system displays the main screen for the new Connection.

Customizing Object Validation Requirements

This section contains the following topics:

- [About Customizing Object Validation](#) on page 14-1
- [Modifying the Custom Validation Package](#) on page 14-1

About Customizing Object Validation

You can use a package called `CDR_DF_CUSTOM_VALIDATION` to set your own standards for object definitions. The internal procedures `createNaming` and `modifyNaming` call this package each time a user creates or modifies an object definition or instance and commits the changes. By default, the package returns a value of `True` and the standard application logic for creating or updating an object proceeds.

However, you can modify this package to enforce your own rules—for example, naming conventions—so that the package returns `False` and displays an error message if one or more rules are violated. The operation fails and the system does not invoke the standard application logic for creating or updating an object.

You can set a rule for any attribute of any object type (not just the name). You can set different rules for different object types.

To see all attributes for all object types predefined in the Oracle Life Sciences Data Hub (Oracle LSH), enter the following query at a SQL prompt:

```
desc apps.cdr_naming_version_obj_type
```

Modifying the Custom Validation Package

`CDR_DF_CUSTOM_VALIDATION` contains one procedure: `validateNaming`:

```
PROCEDURE validateNaming(  
    pi_oNamingVersionObjType IN CDR_NAMING_VERSION_OBJ_TYPE,  
    po_bRetStatus OUT NOCOPY BOOLEAN,  
    po_vRetMsg OUT NOCOPY VARCHAR2 )
```

The procedure uses the parameters as follows:

pi_oNamingVersionObjType is an input parameter for the object type being validated. See the table `cdr_naming_version_obj_type` for the internal name of each object type.

po_bRetStatus is an output parameter with a Boolean return: True if all rules are satisfied, False if one or more rules are violated. By default this parameter value is hardcoded to True.

po_vRetMsg is an output parameter that supplies the text of the error message to be displayed in the user interface if a new or modified object does not pass one or more validation requirements defined in this procedure.

Synchronizing an Apps Password Change

Much of the internal processing in the Oracle Life Sciences Data Hub (Oracle LSH) is done by the Oracle Warehouse Builder (OWB) in the Oracle Applications schema. When you change the password of the apps account, you must synchronize the change in OWB.

Oracle LSH includes a script for this purpose. To run the script, do the following:

1. Log on to the database as apps.
2. Go to \$CDR_TOP/patch/115/sql.
3. Run the script **cdrowbpwsynch.sql**.
4. When prompted, enter the username and password of the OWB repository owner.
5. When prompted, enter the new apps password.

To change the apps schema password, use the FNDCPASS utility. For information, see the *Oracle® E-Business Suite System Administrator's Guide - Security* at http://download.oracle.com/docs/cd/B53825_08/current/acrobat/121sasg.pdf.



Monitoring Jobs

This section contains the following topics:

- [About Execution in the Oracle Life Sciences Data Hub](#) on page 16-1
- [Querying for Jobs](#) on page 16-1
- [Job Information](#) on page 16-3
- [Job Statuses](#) on page 16-7
- [Execution Statuses](#) on page 16-7

In the Job Execution subtab of the Administration tab you can query for and monitor any job submitted by any user.

The LSH System Admin functional role is required to access this screen and perform the tasks described in this section.

About Execution in the Oracle Life Sciences Data Hub

Oracle Life Sciences Data Hub (Oracle LSH) execution is based on the Oracle Warehouse Builder (OWB) runtime execution platform. Oracle LSH passes every internal and user-submitted job to an OWB operator of the appropriate technology type. If OWB is down, Oracle LSH jobs do not run. Information on restarting OWB is included in "[Starting, Stopping, and Checking the Distributed Processing Service](#)" on page 18-1.

If the job requires execution by one of the integrated processing engines (such as SAS, Oracle Reports, or PL/SQL) Oracle LSH assigns the job to a service defined for that processing engine. The Oracle LSH Distributed Processing (DP) Server on the service location must be running for jobs of these types to run. Information about restarting the DP Server is included in "[Starting, Stopping, and Checking the Distributed Processing Service](#)" on page 18-1.

See "Execution and Data Handling" in the *Oracle Life Sciences Data Hub Application Developer's Guide* for additional information on execution and data handling in Oracle LSH. See "Using, Installing, and Cloning Work Areas" in the *Oracle Life Sciences Data Hub Application Developer's Guide* for information on Work Area installation.

Querying for Jobs

You can view information on any job submitted by any user in the Administration tab. Jobs that you personally have submitted also appear in the Job Execution section of your My Home screen.

To query for one or more jobs, enter values in the following fields:

1. Select one of the following:
 - **Show results when ALL conditions are met.** If selected, the system returns only jobs that meet all the conditions you specify in the fields below.
 - **Show results when ANY condition is met.** If selected, the system returns all jobs that meet any of the conditions you specify in the fields below.
2. Enter a value in at least one of the fields listed here to serve as a search criterion. You can enter any number of criteria.

Range Fields. In each of the following fields you can select Equal To, Greater Than, Less Than, or Between from the drop-down list. You then enter a value in the empty field next to the drop-down list. If you select Between, you must also enter a value in the empty field on the right in the same row.

The range fields include:

- **Submission Date.** Search by the date the job was submitted. Click on the day you want in the calendar.
- **Start Date.** Search by the date the job was started. Click on the day you want in the calendar.
- **Completion Date.** Search by the date the job was completed. Click on the day you want in the calendar.
- **Job ID.** Search by the system-generated ID for the job.

Search Fields. These fields have the Search icon to their right. In each case, you can either enter a value directly in the field or click on the Search icon to bring up a Search screen to search for existing values.

The search fields include:

- **Object Type.** In this field you search for the job by the type of executable object that generated the job.

Click the Search icon. In the Search screen, enter % in the empty search field and click Go. When the system returns all executable object instance types, click the Quick Select icon next to the one you want.

Alternatively, enter the object type directly in the field from the main screen: Program, Load Set, Report Set, Workflow, or Data Mart.

Note: To find a Load Set, you must enter "Loadset" not "Load Set."

- **Service.** In this field, you search for the job by the service used to run the job. See [Chapter 1, "Setting Up Services"](#) for information on services.

Click the Search icon. In the Search screen, enter % in the empty search field and click Go. The system returns a list of all services as a concatenation of Service Location - Service Type - Priority. For example, where the service location is 41, the service type is PLSQL, and the job priority defined for the service is High:

```
41-$SERVICETYPES$PLSQL-$JOBPRIORITIES$HIGH
```

Alternatively, enter a string following the pattern shown above directly into the field on the main screen.

- **Execution User.** In this field, you search for the job by the username of the person who submitted the job for execution.

Enter the username directly in the field on the main screen.

Alternatively, click the Search icon. In the Search screen, enter some part of the username with the wildcard character % preceding, following, or both. When the system returns all usernames that match, click the Quick Select icon next to the one you want.

- **Domain.** In this field, you search for the job by the name of the Domain that contains the executable object that generated the job.

Enter the name of the Domain directly in the field on the main screen.

Alternatively, click the Search icon. In the Search screen, enter some part of the Domain name with the wildcard character % preceding, following, or both. When the system returns all Domain names that match, click the Quick Select icon next to the one you want.

- **Application Area.** In this field, you search for the job by the name of the Application Area that contains the executable object that generated the job.

Enter the name of the Application Area directly in the field on the main screen.

Alternatively, click the Search icon. In the Search screen, enter some part of the Application Area name with the wildcard character % preceding, following, or both. When the system returns all Application Area names that match, click the Quick Select icon next to the one you want.

- **Work Area.** In this field, you search for the job by the name of the Work Area that contains the executable object that generated the job.

Enter the name of the Work Area directly in the field on the main screen.

Alternatively, click the Search icon. In the Search screen, enter some part of the Work Area name with the wildcard character % preceding, following, or both. When the system returns all Work Area names that match, click the Quick Select icon next to the one you want.

3. **Show All Status.** Select either **Yes** or **No**. If you select **No**, you can specify which statuses you want to retrieve. See "[Job Statuses](#)" on page 16-7 for an explanation of each status.
4. Click **Go**. The system displays the results of the search in the lower part of the screen. See "[General Job Information](#)" on page 16-3.

You can click on the Job ID hyperlink for any job to see execution details for that job; see "[Job Execution Details](#)" on page 16-4.

Click **Clear** to remove the search criteria and the search results in order to begin a new search.

Job Information

Some job information is available in the search results area at the bottom of the Job Administration screen. Additional information is available when you click the Job ID hyperlink for a particular job; see "[Job Execution Details](#)" on page 16-4.

General Job Information

The system displays the following information about jobs in the search results are at the bottom of the Job Administration screen:

Job ID The system-generated ID of the job. You can click on the Job ID hyperlink to view execution details for the job.

Job Status The current status of the job. See "[Job Statuses](#)" on page 16-7 for further information.

Name The name of the executable object—Program, Load Set, Report Set, Workflow, or Data Mart—that generated the job.

Execution Setup The name of the Execution Setup used to generate the submission form for the job.

Work Area The name of the Work Area within which the job was executed.

Object Type The object type of the executable that generated the job—Program, Load Set, Report Set, Workflow, or Data Mart.

Start TS The timestamp of the beginning of the job.

End TS The timestamp of the end of the job.

User The user who submitted the job for execution.

Job Execution Details

If you click on the Job ID hyperlink in the Job Administration screen, the system opens the Job Execution Details screen, displaying the following information for the job:

- [Submission Properties](#) on page 16-4
- [Outputs](#) on page 16-5
- [Subjobs](#) on page 16-6
- [Submission Parameters](#) on page 16-6
- [System Parameters](#) on page 16-6
- [Source Data Currency](#) on page 16-6

Submission Properties

The Job Execution Details screen includes values for the following properties in the upper portion of the screen:

Job Name The system generates a job name that includes the submission date and timestamp.

Job ID The system-generated unique ID for the job.

Technology Type The engine used to run the job concatenated with the object type that generated the job; for example, SAS Program or Text Data Mart.

Execution Status The highest execution status the job has achieved. If the job is in progress, the status displayed is the current status. If the job has ended in failure or been aborted, the system displays the execution status the job had at the time it ended. See "[Execution Statuses](#)" on page 16-7 for a complete list and description of each execution status.

Job Status The current job status. See "[Job Statuses](#)" on page 16-7 for a complete description of each job status.

Error Message If there is an error message associated with the job, a link to it appears here.

Validation Status The validation status of the executable object instance used to generate the job.

Execution Setup The name of the Execution Setup on which the submission form was based.

Execution User The username of the person who submitted the job for execution.

Service Type The type of service that ran the job; see "[Service Types](#)" on page 1-8 for a description of service types.

Service Status In Oracle LSH Release 2.4.8 this field always displays the status "Running."

Submission Time Stamp The date and time the job was submitted for execution.

Start Time Stamp The date and time the job execution started.

End Time Stamp The date and time the job execution ended.

Outputs

In the Outputs subtab, the system displays a row for each output generated by the job, if any.

To view an actual output:

1. Click the hyperlink in the Name column or the View icon next to it. The Output screen opens.
2. Click **View File**.
3. Choose to open or save the file.

In the Outputs subtab, the system displays the following information about each output:

Title The title is relevant only for primary outputs, which display the title. By default the title is the same as the output's name. However, the Definer can specify a different title.

Description The Definer can enter a description of the output.

Primary Flag If Primary Flag is set to **Yes**, the output constitutes a primary reason for the job execution; it is a report on Oracle LSH data.

If set to **No**, the output is a log or error file produced as a result of running the job but not the primary reason for running the job.

Error Flag If Error Flag is set to **Yes**, the output is an error file resulting from the job execution.

Subjobs

If the job is the running of a Report Set or Workflow, it has subjobs, and Oracle LSH lists them in this section. Program, Load Set, and Data Mart executions do not have subjobs.

Name The name of the executable object—Program, Load Set, Report Set, Workflow, or Data Mart—that generated the subjob.

Type The type of object that generated the subjob.

Job Status The current job status of the subjob; see "[Job Statuses](#)" on page 16-7 for further information.

Execution Status The highest execution status reached by the subjob. If the job is in progress, the status displayed is the current status. If the job has ended in failure or been aborted, the system displays the execution status the job had at the time it ended.

Submission Parameters

Submission parameters are Parameters defined for the particular Program, Load Set, Report Set, Data Mart, or Workflow executed by the job.

Name The system displays the name of the Program, Load Set, Report Set, Data Mart or Workflow instance that was executed. Expand its node (+) to view the Parameters.

Type The object type of the object listed in the Name column.

Value The value in effect at submission.

System Parameters

System parameters are predefined for each object type. For further information, see the chapter on each object type in the *Oracle Life Sciences Data Hub Application Developer's Guide*.

Name The system displays the name of the Program, Load Set, Report Set, Data Mart or Workflow instance that was executed. Expand its node (+) to view the Parameters.

Oracle LSH displays the name of the system parameter followed by its reference codelist.

Type The object type of the object listed in the Name column.

Value The value in effect at submission. If a Parameter is not mandatory, it may have no value. When the allowed values are contained in a reference codelist, the system displays the name of the reference codelist surrounded by dollar signs, with the value at submission following. For example, where the value is \$YESNO\$YES, the runtime value is Yes. The value Yes was selected from the YesNo reference codelist.

Source Data Currency

The Source Data Currency subtab provides information on the most recent refresh of the data from all sources read by the job.

Log File

Click View Log to see the log file for the job.

The log file includes information on the execution statuses the job attained. See ["Execution Statuses"](#) on page 16-7.

Job Statuses

Oracle LSH uses the following statuses to track the progress of a job. You must refresh the screen to see updates.

Pending The job has not yet started running.

Started The job has begun pre-processing.

Executing The Program has connected to database and is running.

Finalizing The job has begun post-processing.

Success The job has completed successfully without warnings.

Warnings The job has completed successfully with warnings.

Failure The job has completed with a fatal error.

Aborted The job has been manually aborted.

On Hold The job is waiting for the Work Area quiesce process to complete.

Expired The system removed the job from the queue after the timeout interval passed.

Duplicate The job is a duplicate of another job; the currency of the source data, parameter values, and executable instance version are the same. Therefore the system does not rerun the job unless the person submitting the job chooses to force reexecution. To view the job that this one would duplicate, paste the Duplicate Job ID in the search window and locate the job.

Execution Statuses

Execution statuses provide more detailed information than job statuses. For diagnostic purposes, the system displays the highest execution status a job reached, even if the job ended in failure.

Submitted Submitted is the initial execution status of the job on the job queue. This phase is required.

Initializing The job is waiting for OWB operator processing to begin. This phase is required.

Serializing The job is waiting for another instance of same program to complete. This phase is necessary only if the Program writes to target tables that require serialization. For further information, see "Data Processing Types" in the *Oracle Life Sciences Data Hub Application Developer's Guide*.

Backchain The system is performing backchain processing; examining the executable instances upstream to check if they have backchain Execution Setups defined and if so, if their source data is more current than their target data, and rerunning qualifying

jobs as necessary to maximize data currency. This phase is required only for jobs specifying a data currency of Most Current Available. For a more complete description of backchaining, see "Backchaining" in the *Oracle Life Sciences Data Hub Application Developer's Guide*.

Recovering The system is rolling back data written to tables in the previous execution of same program that did not complete successfully. This phase is required only if the Program writes to targets and last job failed without recovering.

Pre-Execution Function The system is executing a technology-specific pre-execution function. This phase is required only by some technology types.

Obtain Service The system is obtaining a service instance appropriate for the Program's technology type and specified priority, given the service instances currently available for the technology type. This phase is required.

Pending Logon The system is waiting for the Program to start and connect to the Oracle LSH database. This phase is required for all jobs except Workflow master jobs.

Executing The Program has connected to the database and is executing. This phase is required.

Completing The system is performing post-execution table completion processing. This phase is required only if the Program writes to target tables.

Complete The job has successfully completed processing.

Fail Backchain The job failed during backchain processing.

Fail Logon The job failed to connect to the Oracle LSH database.

Failed The job failed at a phase not specified by one of the other failure statuses.

Rolling Back The system is rolling back target tables to their pre-job state due to the job's execution failure.

Rolled Back The system completed rolling back target tables to their pre-job state due to the job's execution failure.

Exporting and Importing Objects

This chapter contains the following topics:

- [About Exporting and Importing Objects](#) on page 17-1
- [Objects Included and Excluded](#) on page 17-2
- [Source Independence](#) on page 17-4
- [Exporting Objects](#) on page 17-4
- [Importing Objects](#) on page 17-8
- [Installing Objects](#) on page 17-11
- [Preserving Customizations](#) on page 17-13

Note: This utility imports and exports metadata. To export data out of Oracle LSH, use Oracle LSH Data Marts. See "Defining Data Marts" in the *Oracle Life Sciences Data Hub Application Developer's Guide*.

About Exporting and Importing Objects

The Oracle LSH export and import utility (also called the release utility) allows you to export all or part of a single Domain to a file and then read from the file to import the included objects to an Oracle LSH instance—either the same one or a different one—for example, to:

- Set up a new Oracle LSH instance within your organization.
- Make an Oracle LSH application that you developed available to others.
- Provide upgrades to previously imported objects.

The export and import utility includes three scripts:

- **cdrruexport.sql** exports objects to a file.
- **cdrruimport.sql** imports objects from a file.
- **cdrruwainstall.sql** installs imported objects.

The scripts are stored in the following location by default:

```
$CDR_TOP/patch/115/sql/script_name.sql
```

You must run these scripts from the command line on the computer where the Oracle LSH database server is installed. There is no user interface to run these scripts.

Required Setup Each script requires access to a local directory on the database server machine from within SQL*Plus. All of the scripts write log files to this directory. The export script writes the export file to the directory and the import script reads from the file in the directory.

But since a SQL script cannot directly refer to a directory created using the operating system (OS), you must create a local database directory from SQL*Plus and map it to the local OS directory. Instructions are included in the sections on running the scripts.

You need to move the file from the export environment to the import environment through an upload and download, FTP, or any method that you use to transfer files.

Required Oracle LSH User Roles Although you run the scripts as the user `apps`, each script has input parameters requiring the username and password of an Oracle LSH user account with certain privileges; see ["Roles and Privileges Required for Exporting"](#) on page 17-6 and ["Roles and Privileges Required for Importing Domains"](#) on page 17-9.

Objects Included and Excluded

This section contains the following topics:

- [Objects Included](#) on page 17-2
- [Objects Excluded](#) on page 17-3

Objects Included

The following object types can be included:

- **All organizational (container) objects:** Domains, Application Areas, and Work Areas. If you export a whole Domain, all nested Domains and other organizational objects contained in it are included. If you export objects at a lower level of the object hierarchy, the utility automatically includes all parent objects up to the top-level Domain.
- **All object definitions of the following types** contained in the included Domains and Application Areas: Tables, Load Sets, Programs, Variables, Parameters, Parameter Sets, and Business Areas.
- **All the object instances of the same types** contained in the Work Areas: Table instances, Load Set instances, Program instances, Parameter Set instances, and Business Area instances.
- **Underlying object definitions for the included object instances** that are contained within the Domain specified for export. That is, if you specify an object instance for export but not its underlying object definition, the system automatically includes the definition if it is included in the same Domain.

Note: If the underlying object definition of an included object instance is located in a Domain outside the one you specify, the object instance cannot be imported correctly and the export log file contains a warning.

- **Secondary objects contained in primary object definitions:**
 - **Table definition:** Columns and Table Constraints

- **Load Set definition:** Table Descriptors, Planned Outputs, and Parameter Set instances
- **Program definition:** Table Descriptors, Parameter Set instances, Source Code definitions and instances, and Planned Outputs

Note: A single line of source code text that has more than 32765 characters does not get exported correctly.

- **Business Area definition:** Table Descriptors, Joins, and Hierarchies
- **Secondary objects contained in the following primary object instance types:**
 - **Table instance:** Table Constraints
 - **Load Set instance:** Mappings
 - **Program instance:** Mappings
 - **Business Area instance:** Mappings

In addition to the above objects, the following Oracle LSH object components are also exported:

- Version labels
- All runnable active Execution Setups for all instances being exported, including backchain Execution Setups

Note: All versions of versioned objects are included, except for Work Areas, for which only the most recent version is included. Domains and Application Areas are not versioned.

Objects Excluded

The following objects are not exported:

- **All validation supporting information** including all supporting outputs and supporting documents.
- **The classification hierarchy** assigned to any object. On import, the default of the target Oracle LSH instance's classification is used for that object type.
- **The subtype of any object.** On import the default subtype of the target Oracle LSH instance is used for each object type.
- **Checkin and checkout comments.**
- **Execution and Work Area installation data** (jobs, outputs, data).
- **All other object types:** Adapter Domains, Adapter Areas, Workflows, Report Sets, Data Marts.
- **Other Oracle LSH metadata:** Users, User Groups, Roles, Subtypes, Adapter Security, Database Accounts, Remote Locations and Connections, Service Locations, Hierarchies, Terms, and Subtypes.

Source Independence

The utility is designed to detect that an object is the same object even if it was imported first from one location and then from another location, and even if it has a different container object.

Recognizing the Same Object Imported from Different Sources The utility preserves each object's original Company ID and Object ID in an import archive table on the target database. (The Company ID is the same for all objects in a database and, with the Object ID, forms the primary key of each object.) Imported objects also receive a new Company ID and Object ID on the target database.

When you export objects, the export process checks if each included object was previously imported—that is, has a record in the import archive table—and if it was, exports the object with its original Company ID and Object ID.

The import script checks the Company ID and Object ID of each object being imported against the Company ID and Object ID of records in the import archive table. If it finds a match it compares the version numbers and upgrades if a newer version is being imported.

For example, if you:

- export an Application Area from Location A to Location B
- and then export the same Application Area from Location A to Location C
- and then modify one of the Programs it contains on Location C
- and then import the Application Area from Location C to Location B

then the import process running on Location B recognizes that the Program is the same as the existing Program originally imported from Location A, compares the version numbers of the matching Programs, and upgrades the Program in Location B to the newer version.

Updating Namespaces The export includes the current parent of each included object, and the parent's current parent, and so on, up to the top-level Domain. The export script checks if each parent object was itself imported—that is, has a record in the import archive table. If the current parent object was imported, it is included in the export with its original Company ID and Object ID. If the current parent was not imported, it is included using its current Company ID and Object ID. This ensures that each object is exported with its current object ownership hierarchy, even if some objects were originally imported and others were not.

Updating Mappings If an object is imported from a different source location than a previous version of the same object, the import process duplicates the object's source and target mappings from the most recent source in the target location.

Exporting Objects

This section contains the following topics:

- [Full and Partial Export](#) on page 17-5
- [Roles and Privileges Required for Exporting](#) on page 17-6
- [Creating the Directory and Running the Export Script](#) on page 17-6
- [Export Log File Content](#) on page 17-7

The export script:

1. Checks in all the objects being exported, with the checkin comment "Checked in by IEU."
2. Creates a zipped file with the name of the Domain as filename (*domain_name.zip*). This file contains the metadata for exportable objects contained in the Domain.
3. Places the log file (*domain_name.log*) separately in the same directory; see ["Export Log File Content"](#) on page 7.

Full and Partial Export

When you run the export script, you must enter a parameter value to indicate whether you want to perform a full or partial export.

- **Full export.** The export script automatically includes the Domain you specify and almost all the objects it contains (see ["Objects Included"](#) on page 17-2 and ["Objects Excluded"](#) on page 17-3).
- **Partial export.** You must populate a database driver table with a list of the objects to be included. The script automatically includes objects contained in the objects you specify, including objects contained in those objects and objects required for those objects to exist, as described in ["Populating the Driver Table"](#) on page 17-5 and within the limits described in ["Objects Included"](#) on page 17-2 and ["Objects Excluded"](#) on page 17-3.

Populating the Driver Table

The driver table must include a single column that contains the object ID of objects you want to include in the export. You do not need to list every object. **Within the limits described in ["Objects Included"](#) on page 17-2 and ["Objects Excluded"](#) on page 17-3**, the utility automatically includes additional objects as follows:

- **Domains.** If you enter the object ID of a Domain in the driver table, the utility exports all definitions in the Domain, all Application Areas in the Domain, and all objects in the Application Areas, including object definitions, Work Areas, object instances in the Work Areas, and the object definitions and container objects required for them to exist; see "Object Definitions and Instances, and their Containers" in the *Oracle Life Sciences Data Hub Application Developer's Guide* for further information.
- **Application Areas.** If you enter the object ID of an Application Area in the driver table, the utility exports the Application Area and all objects contained in it, including object definitions, Work Areas, and object instances in the Work Areas, and the object definitions and container objects required for them to exist.
- **Work Areas.** If you enter the object ID of a Work Area in the driver table, the utility exports all object instances in the Work Area and the object definitions and container objects required for them to exist.
- **Object instances.** If you enter the object ID of an object instance in the driver table, the utility exports all objects required for the instance object to exist, including both primary and secondary definitions that are instantiated by the instances, any source and target Table instances, and all container objects above them in the object hierarchy. See the *Oracle Life Sciences Data Hub Application Developer's Guide* Appendix on "Object Ownership" for further information.

Note: Some objects may be listed explicitly in the driver table and also included in the export because of their relation to another object. The utility ignores such duplication and exports each object only once.

Roles and Privileges Required for Exporting

When you execute the script you must supply an Oracle LSH user account that has the following privileges:

- Modify privileges on all objects types that have to be exported
- The LSH Checkin Admin application role, if there are objects in the Domain that are checked out by a user other than this Oracle LSH application user

These privileges are required because the export script has to check in all checked out objects before proceeding with the rest of the export process.

If the application user does not have sufficient privileges, and the export script needs to check objects in, the export fails.

See [Chapter 9, "Setting Up the Security System"](#) for more information on granting roles and privileges in Oracle LSH.

Creating the Directory and Running the Export Script

To run the export process, do the following from the database server where you have the Oracle LSH source database instance installed:

1. Connect to the database instance as the Oracle Applications schema owner; usually apps.
2. Create an OS directory where you want the output of the export script to be generated; for example, in the Oracle LSH application user's home directory.

The export process creates .log and .xml files that contain Oracle LSH object metadata information that may be sensitive to your organization. Oracle recommends granting full access to the OS directory only to the Oracle user and the user who runs the export process.

For example:

```
mkdir /user/xyz/exp_imp
# Below command removes all rights on the directory from others
chmod o-rwx /user/xyz/exp_imp
# Assuming the user running export owns the directory and Oracle user is part
of a group called orcl
chgrp orcl /user/xyz/exp_imp
# Grant read,write,execute privileges on the directory to the group
chmod g+rwx /user/xyz/exp_imp
# Grant read,write,execute privileges on the directory owner
chmod u+rwx /user/xyz/exp_imp
```

Notes:

- Do not include any spaces in the OS directory name.
 - The OS directory must be empty before running the export script.
-
-

3. Map the OS directory to a database directory using the following SQL command:

```
CREATE DIRECTORY database_directory_name as path_of_the_OS_directory
```

4. From the OS directory, enter the following command with appropriate values. Each parameter is described in text below.

```
sqlplus apps@LSH_source_database_server_name @path_of_cdrruexport.sql LSH_
application_user Database_directory_name Export_type Export_table Export_
```


column Domain_name

The parameters are:

- **LSH Application Username.** Enter a valid Oracle LSH application user's username. The script checks in all checked out objects in this user's name. The script prompts for the password.
- **Database Directory Name.** Enter the name of the database directory you created in SQL*Plus that maps to the OS directory.
- **Export Type.** Enter **F** to perform a Full export or **P** to perform a partial export; see "[Full and Partial Export](#)" on page 17-5.
- **Export Table.** If you are performing a partial export, supply the name of the database driver table that stores the object IDs of the objects you are exporting.
If you are performing a full export, enter any value; for example, x.
- **Export Column.** If you are performing a partial export, supply the name of the database driver table column that stores the object IDs of the objects you are exporting.
If you are performing a full export, enter any value; for example, x.
- **Domain Name.** Enter the name of the Oracle LSH Domain that you want to export. The Domain name is case sensitive.

Note: If the Domain name contains spaces or any special characters, enclose the Domain name in the escape character (\) followed by double quotes (").

For example, if the Domain's name is `Domain 1`, enter it as:

```
\ "Domain 1\"
```

Note: If you include an object instance in the export whose underlying definition is not contained within this Domain, the export of the object instance does not succeed. The export log file contains a warning.

5. The export script creates a zipped file and a log file in the database directory. Both the files have the Domain's name as the filename. Check the *domain_name.log* file for details of the export operation. See "[Export Log File Content](#)" on page 17-7.

Note: The zipped filename contains underscores in the place of any spaces in the Domain name. However, regardless of the zipped filename, the Domain created in the target Oracle LSH instance has the same name as the exported Domain in the source Oracle LSH instance.

Export Log File Content

The script does not run at all if you provide incorrect usernames (database and/or application user) or an incorrect database directory name. Rerun the script with correct parameter values in that case.

The export process fails if the script is unable to check in some or all exportable objects. This is usually because the Oracle LSH application user whose username you provide while running the script does not have sufficient privileges or if some other user is modifying the Domain at the same time when you run the export script. The export log lists details of objects that it cannot check in.

When the export process runs successfully the log file lists the following:

- Names of all Remote Locations and Remote Connections used by all Load Sets being exported. This is useful in manually creating the same in the target Oracle LSH instance.
- Names of Source Code CLOBs and BLOBs that the export script converted into files
- The object name, object type, version number, and container details of:
 - All objects that the export script could successfully check in and include in the export
 - All objects that could not be included in the export and the reason for that.

Note: Object instances that refer to definitions outside the Domain being exported will not work in the target Oracle LSH instance.

You must examine the log file, copy object definitions for such instances into the Domain, update the object instances to point to the copied object definitions, and rerun the export script.

Importing Objects

This section contains the following topics:

- [Upgrading an Existing Domain](#) on page 17-8
- [Roles and Privileges Required for Importing Domains](#) on page 17-9
- [Creating the Directory and Running the Import Script](#) on page 17-9
- [Import Log File Content](#) on page 17-10
- [Applying Security to an Imported Domain](#) on page 17-11

The import script checks to determine whether a Domain with the same name as the Domain being imported already exists. If it does, it checks whether the existing Domain was also created using the import script. If the existing Domain is not an imported Domain, the import script creates a new Domain. If the existing Domain is an imported Domain, it upgrades objects that have new versions, allowing you to deploy new versions of an Oracle LSH application on the target Oracle LSH instance.

Upgrading an Existing Domain

The import script performs the following tasks:

- If any objects are currently checked out in the Domain on the target Oracle LSH instance, the script checks them in with the comment "Checked in by IEU".
- Checks out all objects with the comment "Checked out by IEU" and updates the versions of the object definitions and instances if it finds new versions in the export file.

- Checks in object definitions after versioning and applies the comment "Checked in by IEU."
- Creates a log file in the database directory on the target Oracle LSH instance. The log file has the same name as the Domain: domain_name_import.log. See "[Import Log File Content](#)" on page 17-10.
- You must check the log and decide whether or not to commit changes to the database on account of the upgrade. You can rollback the changes if the import log shows problems during upgrading.

Roles and Privileges Required for Importing Domains

When you execute the script you must supply an Oracle LSH user account that has the following privileges:

- **LSH Checkin Admin** application role. This role is essential for importing or upgrading a Domain.
- **LSH Bootstrap Admin** application role. This role is required only for creating new Domains, not for Domain upgrades. This role allows an application user to create the Domain and objects under it.
- Sufficient privileges to be able to install Oracle LSH Work Areas.

This user does not require any other privileges to run the import because object security is temporarily granted to this user as part of the import process.

See [Chapter 9, "Setting Up the Security System"](#) for more information.

Creating the Directory and Running the Import Script

To import an Oracle LSH Domain, run the following commands from the database server where your target Oracle LSH instance is installed:

1. Connect to the database instance as the Oracle Applications schema owner; usually apps.
2. At least the first time you run the script, create an OS directory where you want the import log to be generated. For example, create it in the LSH application user's home directory so that it is easy to find.

The import process creates a .log file that contains Oracle LSH object metadata information that may be sensitive to your organization. Oracle recommends granting full access to the OS directory only to the Oracle user and the user who runs the import process.

For example:

```
mkdir /user/xyz/exp_imp
# Below command removes all rights on the directory from others
chmod o-rwx /user/xyz/exp_imp
# Assuming the user running import owns the directory and Oracle user is part
of a group called orcl
chgrp orcl /user/xyz/exp_imp
# Grant read,write,execute privileges on the directory to the group
chmod g+rx /user/xyz/exp_imp
# Grant read,write,execute privileges on the directory owner
chmod u+rx /user/xyz/exp_imp
```

Notes:

- Do not include any spaces in the OS directory name.
 - Before running the import script, be sure that you have only the current *domain_name.zip* file in the OS directory.
-
-

3. Map the OS directory to a database directory using the following SQL command:

```
create directory database_dir_name as path_of_the_OS_directory
```

4. Go to the OS directory you created and copy the exported *Domain* file into it.
-
-

Note: If you have not created the exported *Domain* file yourself, you can receive it from the person who has through a download or an email—any medium that you use to transfer files.

5. Run the import script as follows:

```
sqlplus apps@LSH_target_database_server_name @path_of_cdrruimport.sql LSH_
application_user Database_directory_name Zipped_Domain_Filename COMMIT_ON_NO_
ERROR/MANUAL_COMMIT
```

6. The script requires the following command line parameters in this sequence:

- **LSH Application Username.** Enter a valid Oracle LSH application user's username. The import script creates all the imported Domain's objects and performs other required operations on them, such as checking the objects in or out, as this Oracle LSH application user. The script prompts you for a password for this user, when run.
- **Database Directory Name.** Enter the name of the database directory you created in SQL*Plus that maps to an OS directory where you want the import script to generate the import log file; see "[Required Setup](#)" on page 17-2.
- **Zipped Domain Filename.** Enter the name of the zipped export file (*domain_name.zip*) that you copied from the source Oracle LSH instance.
- **COMMIT_ON_NO_ERROR/MANUAL_COMMIT:** You must specify one of these options to the import script:
 - **COMMIT_ON_NO_ERROR.** If you want Oracle LSH to commit the imported Domain's data to the database, enter this option. Oracle LSH commits the data if there are no errors while importing.
If there are errors, Oracle LSH rolls back the data and you must rerun the import script after fixing the errors.
 - **MANUAL_COMMIT.** Enter this option if you want to commit or roll back the data manually after examining the import log.

7. The import script creates a log file in the database directory with a name in this format: *domain_name_import.log*. See "[Import Log File Content](#)" on page 17-10.

Import Log File Content

The import log file contains the following information:

- Any errors that the system encounters while creating objects in the target Oracle LSH instance.
- The object name, object type, version number, and container details of:
 - All objects that the import script could successfully create or upgrade during the import
 - All object instances that the import script did not upgrade because the instance objects referenced object definition not contained in the imported Domain. The Import Export Utility allows preservation of user customizations through this feature. See "[Preserving Customizations](#)" on page 17-13.
- Remote Locations and Connections that do not exist in the target instance but are referenced by one or more Load Sets in the imported Domain. Such Load Sets may not be installable until you create the Remote Locations and Remote Connections in the target instance.
- Objects that were installable in the source instance but are noninstallable in the target Oracle LSH instance.
- Table instances that were upgradable in the source instance but are nonupgradable in the target Oracle LSH instance.
- If you ran the install script, the import log lists details of all objects successfully installed and errors and warnings, if any. In addition, the following details are logged for an install:
 - Status of all Work Areas imported.
 - Submission ID, job ID, and the Execution Setup location and version number for all backchain submissions created.

Applying Security to an Imported Domain

You must assign at least one user group to a new imported Domain (not required for an *upgraded* imported Domain) to be able to view and use the other objects imported in it. You must have the LSH Bootstrap Admin Role for this operation.

From the Domain's properties screen, do the following:

1. From the **Actions** drop-down list, select **Apply Security** and click **Go**. The system opens the Manage Security screen.
2. Click **Assign Group**. The system opens the Search User Group screen.
3. If you know the name of the user group you want to assign to the Domain, enter it in the **Group Name** field. You can also enter part of a name and the system will return all groups that include the string you enter in their name. If you leave the field blank, the system returns all user groups.
4. Click **Go**. The system displays all user groups that satisfy your query.
5. Select any number of user groups: click their Select check box and click **Apply**.
6. Click **Return** to return to the Properties screen for the Domain.

Installing Objects

You can run the install script to install objects in the imported Domain after you accept an import by committing changes to the target Oracle LSH database instance. If you ran the import script using `COMMIT_ON_NO_ERROR` and there were no errors, the import script committed the changes to the database. If you ran the import script set to

MANUAL_COMMIT you must commit the changes manually after examining the log file.

The install script accepts the same parameters as the import script. You can also run the install script after upgrading a Domain.

You must have the privileges required to install Oracle LSH Work Areas.

The install script performs the following tasks:

- Installs all the installable Work Areas in the imported Domain. For upgraded Domains, only Work Areas that the import script upgraded or added while upgrading, are installed.
- In all successfully installed Work Areas that include imported objects with backchain Execution Setups, the install script submits these Execution Setups and the jobs run in backchain mode.

For upgraded Domains, the install script does not submit backchain Execution Setups in the following cases:

- If the object instance in the upgraded Domain is customized (that is, refers to an object definition outside the Domain)
- If the source object instance does not contain a backchain Execution Setup but the target object instance does
- If the source instance's backchain Execution Setup has not been modified since the last import but the target instance has a new backchain Execution Setup
- Writes details of the installation process to domain_name_import.log stored in the database directory.

See "Installation Requirements for Each Object Type" in the *Oracle Life Sciences Data Hub Application Developer's Guide* for reasons for a Work Area installation to fail.

Running the Installation Script

To run the install script, do the following from the target Oracle LSH database instance:

1. Connect to the database instance as the Oracle Applications schema owner; usually apps.
2. Change to the OS directory you created while running the import script.
3. Make sure the zipped Domain file exists in the OS directory and that it is the latest exported Domain. For a Domain upgrade, the install script first checks whether or not the exported Domain file has the same or higher version numbers for objects. Installation does not continue if the exported Domain file is not the latest.
4. Run the install script from the OS directory as follows:

```
sqlplus apps@LSH_target_database_server_name @path_of_cdrruainstall.sql LSH_
Application_Username Database_Directory_Name Zipped_Domain_Filename
```

Enter the same values for the parameters that you did while running the import script. See "[Creating the Directory and Running the Import Script](#)" on page 17-9.

5. The script adds the install-related messages to the import log file.

Preserving Customizations

If you have made changes to imported objects in the target instance that you want to preserve, use the Move operation to move the customized object definitions into a different Domain.

The Move operation keeps references intact, so that instances of each moved definition continue to point to the same definition in its new location.

The import utility performs a check so that it never overwrites an object instance that points to a definition outside the imported Domain, even if the object instance is included in the import.

The utility does import definitions from the source that correspond to the definitions you have moved, but they have no effect because no object instances use them.

Stopping and Starting Services and Queues

This section contains the following topics:

- [Starting, Stopping, and Checking the Distributed Processing Service](#) on page 18-1
- [Stopping and Starting the Oracle Warehouse Builder Service](#) on page 18-4
- [Stopping, Starting, and Enabling the Job Queue](#) on page 18-5
- [Starting and Stopping Message-Triggered Submission Queues](#) on page 18-6

For information about the initial installation of Oracle Warehouse Builder or the Distributed Processing Server, see "Installing the Oracle Life Sciences Data Hub" in the *Oracle Life Sciences Data Hub Installation Guide*. See [Chapter 1, "Setting Up Services"](#) for information about the Distributed Processing (DP) Server.

Note: During its development, the Oracle Life Sciences Data Hub (Oracle LSH) was known internally as CDR. Therefore many Oracle LSH-related directories, files, scripts, parameters, and so on are named CDR or contain the string `cdr`. Please think of CDR as a synonym for Oracle LSH.

Starting, Stopping, and Checking the Distributed Processing Service

This section contains the following topics:

- [Starting the Distributed Processing Server Service](#) on page 18-2
- [Checking the Distributed Processing Server Service](#) on page 18-3
- [Stopping the Distributed Processing Server Service](#) on page 18-4

The Oracle LSH Distributed Processing (DP) Server passes jobs—executions of user-defined Programs and Data Marts and internal Oracle LSH jobs requiring XML Publisher—to the external engines that process them. You must install and start the DP Server on every computer where you have a processing engine installed. See [Chapter 1, "Setting Up Services"](#) for further information.

You must then start the service on the computer where it is required. The scripts required for starting, checking, and stopping the DP Server are provided with Oracle LSH. See ["Setting Up the Distributed Processing Server"](#) on page 1-3 and the *Oracle Life Sciences Data Hub Installation Guide* for instructions for copying them to the DP Server home directory in each location.

Login Name You must log in as `cdr_dpserver`. This user is created by the post-installation job. Your company should have changed the default password. See

"Run the Post-Installation Programs" in the chapter on installing Oracle LSH in the *Oracle Life Sciences Data Hub Installation Guide* for further information.

Starting the Distributed Processing Server Service

To start the service, do the following:

1. If you have not already edited the script `cdr_apps_dpserver.sh` (or `cdr_apps_dpserver.cmd`), do so. See "Edit the DP Server Start Script" in the chapter on installing Oracle LSH in the *Oracle Life Sciences Data Hub Installation Guide*.

2. Log on as `cdr_dpserver`.

This user is created when you run the post-installation job (see the "Run the Post-Installation Programs" in the chapter on installing Oracle LSH in the *Oracle Life Sciences Data Hub Installation Guide*).

3. Run the script `cdr_apps_dpserver.sh` (or `cdr_apps_dpserver.bat`) by entering the following command:

```
./cdr_apps_dpserver.sh ORACLE_SID DB_HOST DB_PORT RAC_TNS RAC_FLAG
```

where:

- `ORACLE_SID` is the Oracle SID of the database

Note: The Oracle SID is case-sensitive.

- `DB_HOST` is the name of the computer where the `ORACLE_SID` resides
- `DB_PORT` is the SQL*Net Listener port for the `ORACLE_SID`
- `RAC_TNS` is the JDBC connection string of the database server
- `RAC_FLAG` indicates whether you are using an Oracle RAC (Real Application Cluster) database installation. Set to `RAC` if you have a RAC installation. Set to `NO-RAC` if you do not.

The `RAC_FLAG` setting determines which input parameter values the script uses when starting the DP Server.

- If `RAC_FLAG` is set to `RAC`, the script uses only the value for `RAC_TNS`.
- If `RAC_FLAG` is set to `NO-RAC`, the script uses the values for `ORACLE_SID`, `DB_HOST`, and `DB_PORT`.

In either case, it does not matter what value you enter for the unused parameters. See examples below.

- `FW_ENABLED` Set to **Yes** to start the File Watcher Service or **No** if you are not using Oracle DMW.
- `FW_FREQ` (Applies only to Oracle DMW customers.) Refresh frequency in seconds. This value specifies the minimum interval between requests to the database to check if there is a new set of Watcher Configurations. This value cannot be set lower than 60 seconds. A high setting will result in a delay between the user's addition or adjustment of a Watcher Configuration in Oracle DMW and the changes' taking effect in file detection behavior.
- `FW_POLL` (Applies only to Oracle DMW customers.) Polling frequency in seconds. The polling frequency represents the minimum interval at which a File Watcher Service may run to detect if there are any files in the watched

location that should be loaded into Oracle DMW. The minimum value permitted is 60 seconds.

NO-RAC Example when RAC_FLAG is set to NO-RAC:

```
./cdr_apps_dpserver.sh abc123 ab123cd.example.com 1234 NA NO-RAC
```

where:

- abc123 is the Oracle SID
- ab123cd.example.com is the host
- 1234 is the port
- You may enter NA (Not Applicable) or any other value for RAC_TNS.
- NO-RAC is the setting for RAC_FLAG

RAC Example when RAC_FLAG is set to RAC:

```
./cdr_apps_dpserver.sh NA NA NA 'jdbc:oracle:thin:@(DESCRIPTION=(LOAD_
BALANCE=YES) (FAILOVER=YES) (ADDRESS_
LIST=(ADDRESS=(PROTOCOL=tcp) (HOST=ab123cd.example.com) (PORT=1234)) (ADDR
ESS=(PROTOCOL=tcp) (HOST=ef456ghrac.example.com) (PORT=5678))) (CONNECT_
DATA=(SERVICE_NAME=CDRXXX)))' RAC
```

where:

- You may enter NA (Not Applicable) or any other value for ORACLE_SID.
 - You may enter NA (Not Applicable) or any other value for DB_PORT.
 - You may enter NA (Not Applicable) or any other value for DB_HOST.
 - jdbc:oracle:thin:@(DESCRIPTION=(LOAD_
BALANCE=YES) (FAILOVER=YES) (ADDRESS_
LIST=(ADDRESS=(PROTOCOL=tcp) (HOST=ab123cd.example.com) (PORT=1234)) (
ADDRESS=(PROTOCOL=tcp) (HOST=ef456ghrac.example.com) (PORT=5678))) (CO
NNECT_DATA=(SERVICE_NAME=CDRXXX))) is the JDBC connection string of the
database server
 - RAC is the setting for RAC_FLAG
4. The script prompts you for a password. Enter the password for the cdr_dpserver user.

Note: Do not change the value of DB_USER.

Checking the Distributed Processing Server Service

After you have installed and started the service, use the script checkJSapps to check that it is running. Enter the ORACLE_SID as a parameter value.

1. Log on as cdr_dpserver.
2. Run the script **checkJSapps.sh** (or **cdr_apps_dpserver.bat**) by entering the following command:

```
checkJSapps Oracle_SID
```

For example:

```
./checkJSapps abc123
```

where abc123 is the SID.

Note: The Oracle SID is case-sensitive.

Stopping the Distributed Processing Server Service

To stop the service, do the following:

1. Log on as `cdr_dpserver`.
2. Run the script **stopJSapps.sh** by entering the following command:

```
stopJSapps.sh Oracle_SID DPService_Name
```

where:

- *Oracle_SID* is the Oracle SID of the database

Note: The Oracle SID is case-sensitive.

- *DPService_Name* is the Service Location name for the computer where the DP Server is running.

Stopping and Starting the Oracle Warehouse Builder Service

The Oracle Warehouse Builder (OWB) service is required to run any Oracle LSH job and to install Work Areas. The OWB service is started automatically by the Oracle LSH database startup kit during installation.

Note: The OWB server's log files are located in the directory `$OWB_HOME/owb/log`. If you have problems with a Work Area installation, the log file may help you diagnose the problem.

Stopping the OWB Service

To stop the OWB service do the following:

1. Change directory to `$OWB_HOME/owb/rtp/sql`
2. Log into SQL using the `CDR_RTREPOS` username and the password you set up when you installed OWB (see "Installing Oracle Warehouse Builder" in the *Oracle Life Sciences Data Hub Installation Guide*).

Note: If you have upgraded to OWB 11.2.0.1 you must log in as `OWBSYS` instead of `CDR_RTREPOS`.

3. At the SQL prompt, enter:

```
@stop_service.sql
```

When the system displays the words "Not Available" the service is stopped.

Starting the OWB Service

To start the OWB service do the following:

1. Change directory to `$OWB_HOME/owb/rtp/sql`

2. Log into SQL using the CDR_RTREPOS username and the password you set up when you installed OWB (see "Installing Oracle Warehouse Builder" in the *Oracle Life Sciences Data Hub Installation Guide*).

Note: If you have upgraded to OWB 11.2.0.1 you must log in as OWBSYS instead of CDR_RTREPOS.

3. At the SQL prompt, enter:

```
@start_service.sql
```

When the system displays the word "Available" the service has been restarted.

Stopping, Starting, and Enabling the Job Queue

A job queue is required on the database to pass jobs from Oracle LSH to OWB. The job queue is enabled by the Oracle LSH post-installation concurrent processing job. You must explicitly start it using the instructions below.

Note: The scripts are located in \$cdr/patch/115/sql/.

This section includes the following topics:

- [Starting the Job Queue](#) on page 18-5
- [Stopping the Job Queue](#) on page 18-5
- [Enabling the Job Queue](#) on page 18-5

Starting the Job Queue

To start the job queue, connect as apps and run this script:

```
begin
cdr_exe_job_queues.start_jobq_process_disabled;
end;
/
commit;
```

Stopping the Job Queue

You can stop and disable the job queue by connecting as apps and running this script:

```
begin
cdr_exe_job_queues.stop_processing_queues;
end;
/
commit;
```

Enabling the Job Queue

After you have started the job queue (or it has been started during installation) you must enable the job queue by connecting as apps and running a script as follows:

```
begin
cdr_exe_job_queues.start_jobq_process_enabled;
end;
/
```

```
commit;
```

Starting and Stopping Message-Triggered Submission Queues

This section contains the following topics:

- [Starting a Message-Triggered Submission Queue](#) on page 18-6
- [Stopping a Message-Triggered Submission Queue](#) on page 18-6

If you plan to use message-triggered job submission, you must start a job queue on the database to pass jobs to Oracle LSH from XML messages from external systems. See "Using Message-Triggered Submission from External Systems" in the *Oracle Life Sciences Data Hub Application Developer's Guide*.

Starting a Message-Triggered Submission Queue

Log on as the APPS user and enter the following to start the queue:

```
begin
cdr_exe_msg_queues_admin.start_processing_queues;
cdr_exe_msg_queues_admin.enable_job_processing_queue;
end;
```

Stopping a Message-Triggered Submission Queue

Log on as the APPS user and enter the following to stop the queue:

```
begin
cdr_exe_msg_queues_admin.stop_processing_queues;
end;
```

Troubleshooting

This section contains the following topics:

- [Job Execution Problems](#) on page 19-1
- [Database Core Dumps during Object Creation](#) on page 19-3
- [Query Exceeds 200 Rows](#) on page 19-3
- [Table Auditing \(Journaling\) Not Working](#) on page 19-3
- [Enabling a Trace on a Job](#) on page 19-4

Note: See also "Troubleshooting Oracle Applications Performance Issues," Note 169935.1 on My Oracle Support.

Note: During its development, the Oracle Life Sciences Data Hub (Oracle LSH) was known internally as CDR. Therefore many Oracle LSH-related directories, files, scripts, parameters, and so on are named CDR or contain the string cdr. Please think of CDR as a synonym for Oracle LSH.

Job Execution Problems

This section includes the following topics:

- [Job Remains at Submitted or Executing](#) on page 19-1
- [Oracle Warehouse Builder Processes Do Not Start](#) on page 19-2
- [Job Runs But Does Not Upload Outputs](#) on page 19-2
- [Job Log File Does Not Contain Info after a Long or Failed Execution](#) on page 19-3
- [Job Fails to Obtain Service](#) on page 19-3
- [Report Set Jobs Do Not Produce Any Output when using SAS 9.2](#) on page 19-3

Job Remains at Submitted or Executing

If a job does not progress beyond a status of Submitted or Executing, try the following procedure:

1. Log in as Apps.
2. Run the following SQL script:

```
SELECT q.JOB_PROCESSING_FLAG_RC, q.LOG_MESSAGE_ID,TO_CHAR(q.LOG_STAMP),
q.message FROM cdr_job_queues_log q ORDER BY q.LOG_STAMP DESC
```

3. Check the resulting message. If the message is similar to the following, proceed to Step 4:
 - dequeued from control Q: _JOBCONTROL_STOP
 - Caught Exception (-20200) (ORA-20200: *** ERROR: cdr_exe_owb_execution.startedJob: SQL error: ORA-06508: PL/SQL: could not find program unit being called ORA-25228: timeout or end-of-fetch during message
4. Stop the OWB service.

To do this, connect as cdr_rtrepos and run stop_service.sql (see ["Stopping the OWB Service"](#) on page 18-4).
5. Stop the job queue.

To do this, connect as apps and run stopq.sql (see ["Stopping the Job Queue"](#) on page 18-5).
6. Start the OWB service.

To do this, connect as cdr_rtrepos and run start_service.sql (see ["Starting the OWB Service"](#) on page 18-4).

You should get a message that the service is available.
7. Start the job queue.

To do this, connect as apps and run startq.sql (see ["Starting the Job Queue"](#) on page 18-5).
8. Rerun the query given in Step 2.

You should get a message:

```
dequeued from control Q: _JOBCONTROL_ENABLE
```

It may take some time for the message to appear.
9. Rerun the job.

Oracle Warehouse Builder Processes Do Not Start

If Oracle Warehouse Builder (OWB) processes do not start, check the init.ora file for the setting of the job_queue_processes parameter. Oracle recommends a setting of 10, but you may require more. If you have too few job queue processes started, OWB processes may not be able to start.

Job Runs But Does Not Upload Outputs

If a job runs on an external engine (SAS, Oracle Reports, or XML Publisher) and generates outputs but doesn't upload the outputs to Oracle LSH, check the DP Server log. The DP Server log directory is on the computer where the DP Server is installed, usually in the following location:

```
$CDR_HOME/<DBName>/log
```


Job Log File Does Not Contain Info after a Long or Failed Execution

If a job executes for a long time or fails and the job log does not give enough information, check the Oracle Warehouse Builder (OWB) log.

The OWB log directory is located on the machine where the database is installed, usually in the following location:

```
$OWB_HOME/Log
```

Job Fails to Obtain Service

If a job that requires an external engine (SAS, Oracle Reports, or XML Publisher) fails at *Obtainservice* (or you get a No Data Found error at this point) run the script `checkJSapps` to see if the DP Server is running; see "[Checking the Distributed Processing Server Service](#)" on page 18-3.

If the DP Server is not running, follow instructions at "[Starting the Job Queue](#)" on page 18-5.

Report Set Jobs Do Not Produce Any Output when using SAS 9.2

If you are using SAS 9.2 and find that your Report Set jobs fail and do not generate any output, you must install the one-off patch 6993271. See "Apply Required Oracle Applications Patches" in the *Oracle Life Sciences Data Hub Installation Guide* for details on how to install the patch.

Database Core Dumps during Object Creation

If your database core dumps during the creation of Domains and other objects, the cause may be a bug in Oracle Text (bug number 5587976). At the time of publication of this document, patch 5587976 fixes this issue on Linux x86. If you are running on a different platform, work with Oracle Support to get a backport to RDBMS 10.1.0.4 for your platform.

Query Exceeds 200 Rows

If you get the error message "Query has exceeded 200 rows. Potentially more rows exist, please restrict your query" in a situation where you cannot restrict your query—for example, you have more than 200 Domains created, or more than 200 Applications in a Domain, or more than 200 Work Areas in an Application Area, or more than 200 object instances in a Work Area—then you may want to reset the Oracle Applications profile FND: View Object Max Fetch Size. This profile is set to 200 by default. See [Chapter 3, "Setting Profile Values"](#) for further information.

However, use caution because changing this value affects all your Oracle Applications queries as well as all your Oracle LSH queries.

Table Auditing (Journaling) Not Working

Document number 105624.1 on My Oracle Support is a troubleshooting guide for auditing (also called journaling).

Enabling a Trace on a Job

To help diagnose performance issues, you can create a trace file for any job. Do the following:

1. In the Output Description Suffix field of the Submission screen for the job in the Oracle LSH user interface, enter the following string: `!enable_trace!`

When you submit the job, the system creates a trace file for the job. The trace file name contains the identifier `LSHJOB`.

2. (Optional) Enter a string in the Output Title Suffix field. The system adds that string to the trace file name immediately before the `.trc` extension.
3. Run the job.
4. Use the following query to find the location of the file:

```
select value from v$parameter where name='user_dump_dest'
```

Object Ownership

In the Oracle Life Sciences Data Hub (Oracle LSH), object definitions and instances are "owned" by their immediate container object. The following diagrams show the Oracle LSH data model for object ownership:

- [Figure A-1, "Object Ownership within a Domain"](#)
- [Figure A-2, "Object Ownership within an Application Area"](#)
- [Figure A-3, "Object Ownership within a Work Area"](#)

Object ownership has ramifications in many areas of Oracle LSH, including security (user group assignments) and classification; it is possible for objects to inherit the user group assignments and classifications of their owning objects. In addition, Oracle LSH enforces unique names for objects of the same type in the same container object.

Domains

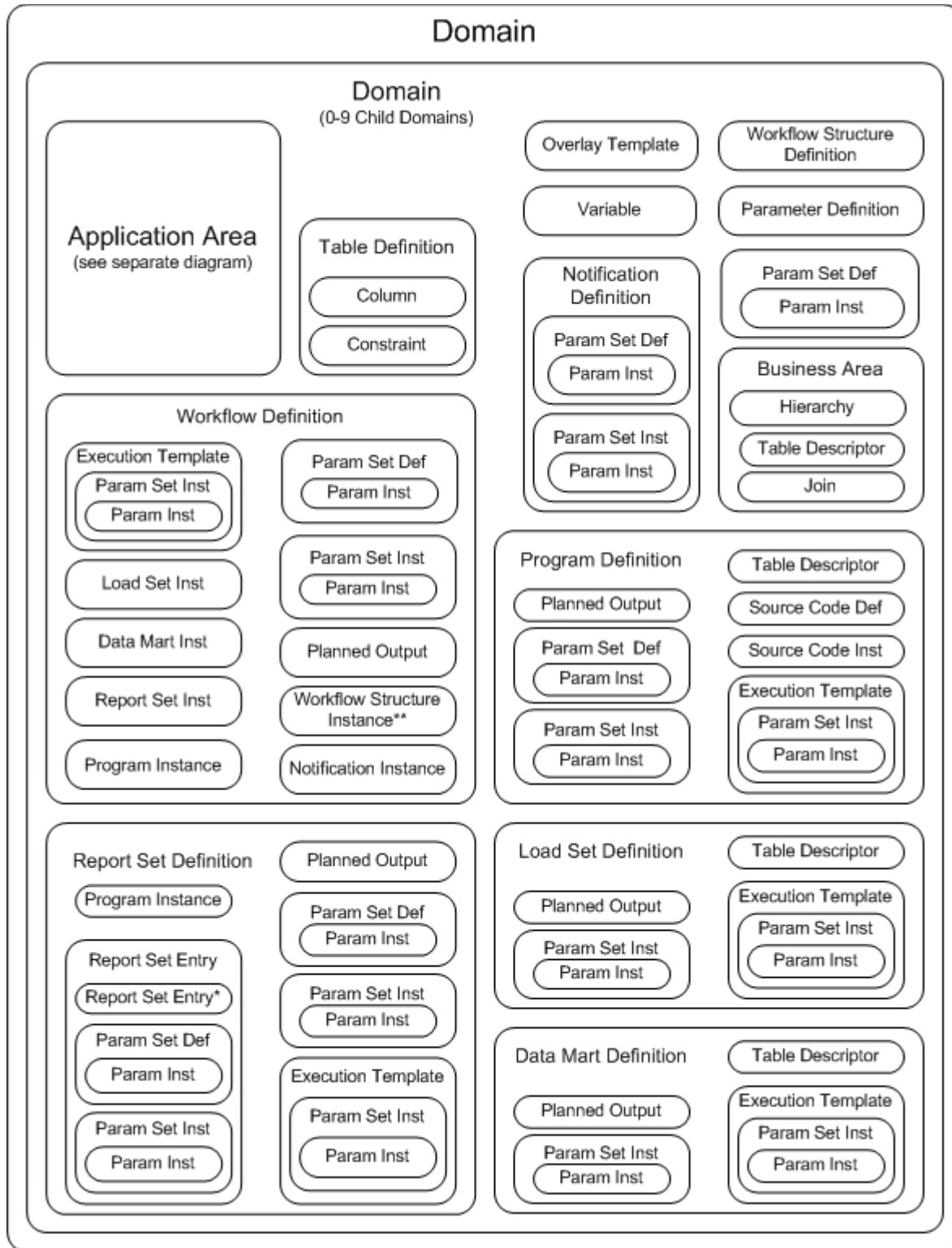
Domains contain Application Areas and object definitions. Depending on the setting of the Object definitions contained directly in a Domain constitute the Domain library.

In addition, Domains can contain child Domains that themselves contain child Domains, up to nine (9) levels, depending on the value of the Domain Nest Value profile setting for your LSH implementation. A Domain can contain any number of Domains at a single level. For example, if the Domain Nest Value profile is set to one (1), a top-level Domain can contain any number of child Domains, but those child Domains cannot contain any child Domains of their own.

[Figure A-1](#) shows all layers of object ownership within a Domain, except for objects contained in the Application Area. See [Figure A-2](#) for object ownership within an Application Area.

Note: Report Set Entries inside Report Set Definitions can be nested indefinitely.

Figure A-1 Object Ownership within a Domain



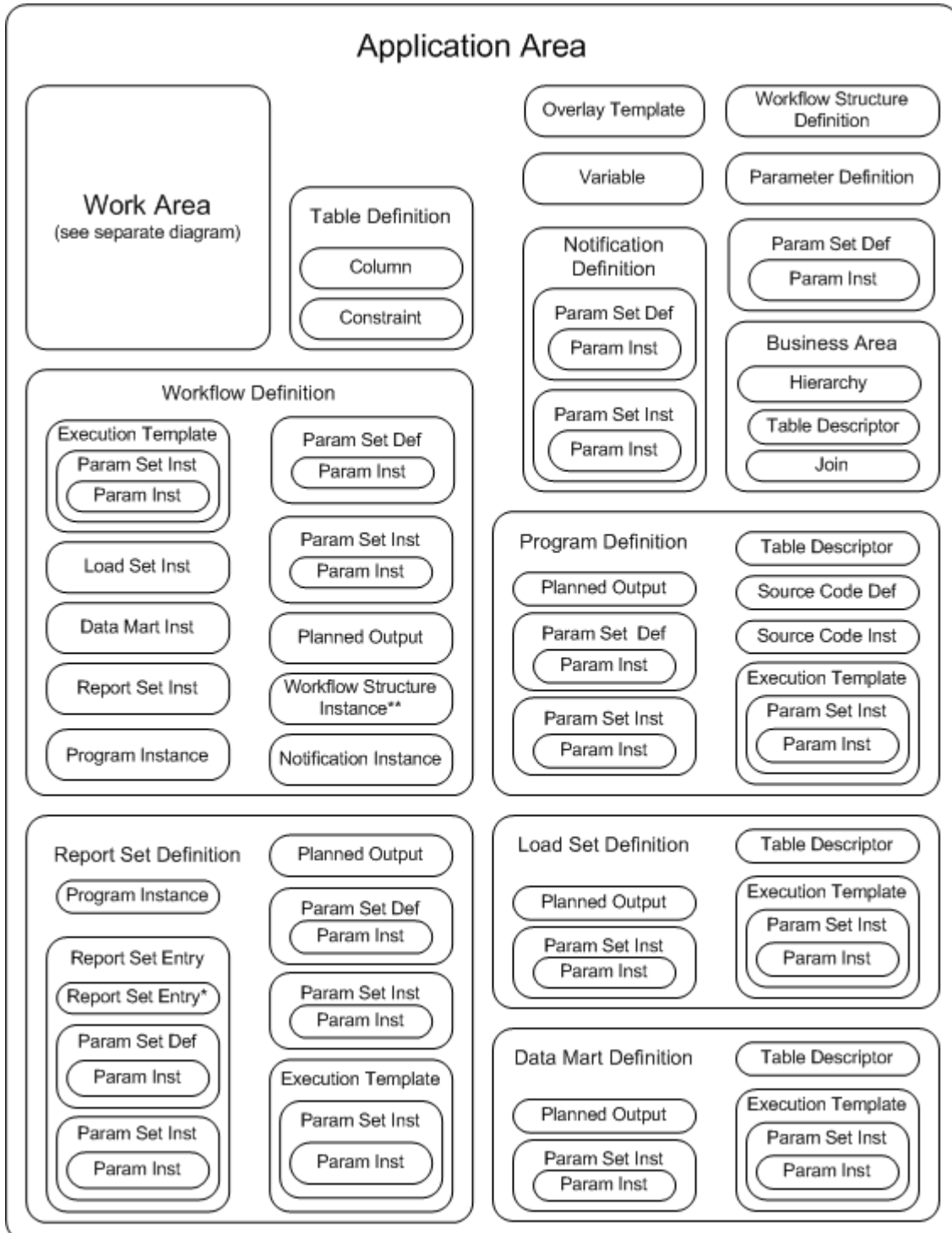
Application Areas

Application Areas contain exactly the same objects as Domains, except that Domains contain Application Areas and Application Areas contain Work Areas. Both Domains

and Application Areas can include all object definition types. Figure A-2 shows all possible layers of object ownership within an Application Area.

Note: Report Set Entries inside Report Set Definitions can be nested indefinitely.

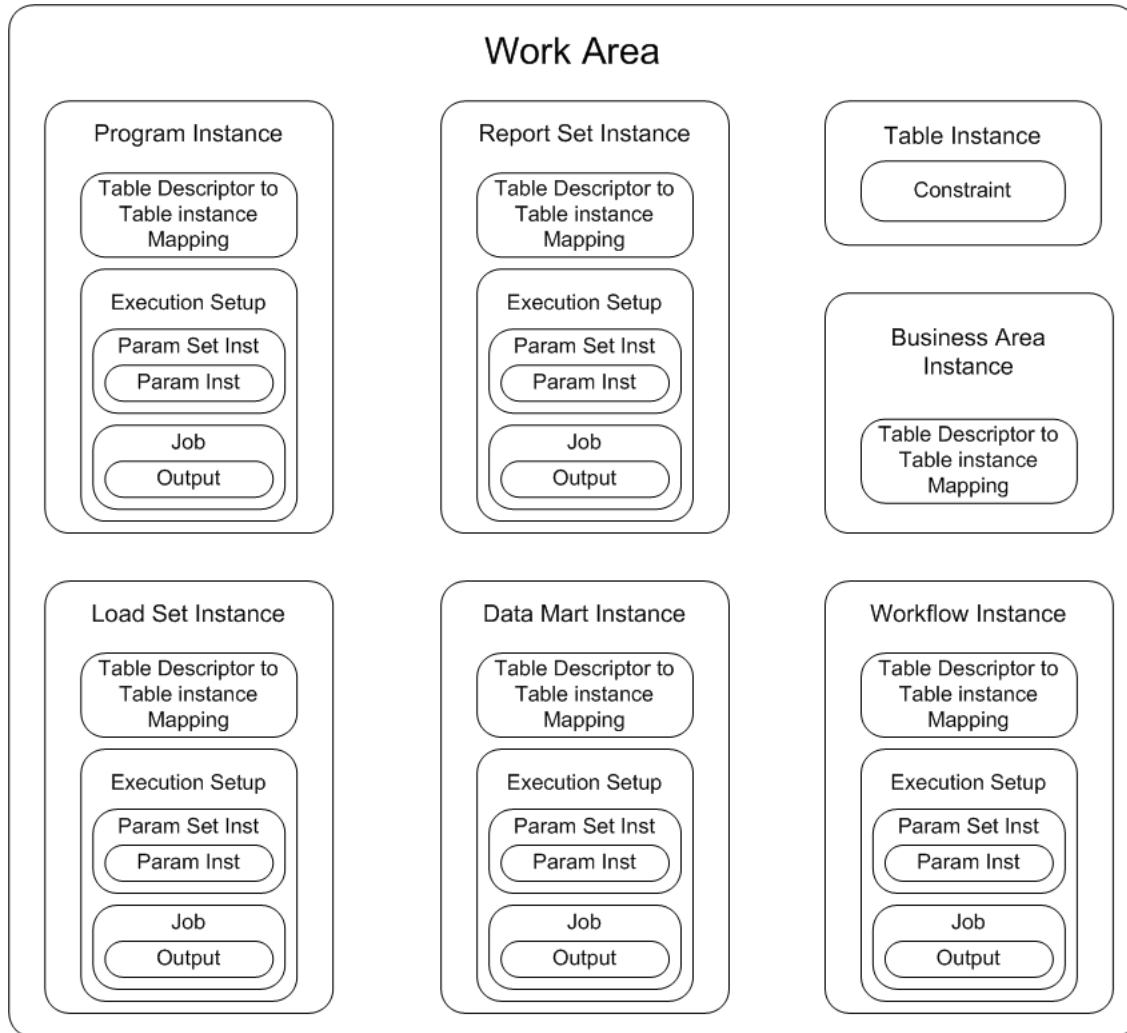
Figure A-2 Object Ownership within an Application Area



Work Areas

Work Areas contain object instances, each of which points to an object definition located either in an Application Area or in a Library. This diagram shows only object ownership, so you cannot see the pointers to the definitions. Most object instances point to an object definition with the same type name; for example, Program instances point to Program definitions. There are two exceptions: both a Parameter and a Table Column point to a Variable as their definition source. [Figure A-3](#) shows all possible layers of object ownership within an Application Area.

Figure A-3 Object Ownership within a Work Area



Glossary

activity

An element of a Workflow; either an executable object such as a Program, a structural object such as a Fork, or a Notification.

adapter

An interface between Oracle LSH and another system. Oracle LSH includes adapters that allow you to load data into Oracle LSH from other systems, to transform data by creating Programs in integrated development environments, and to create visualizations of Oracle LSH data for display in other systems.

application

The set of all the defined objects required to perform a particular business function such as loading and analyzing data for a study or producing a particular set of reports.

Application Area

A container that is used to develop and manage the Oracle LSH objects required for a single business application (such as a study, project, or set of reports); contains one or more Work Areas where the set of object instances (such as Table and Program instances) necessary for the business application are installed.

Application Area library

A set of object definitions contained directly in an Application Area.

audit

A record of each change made to each record in a Table instance defined with a processing type that includes auditing, with the timestamp of the change and the user ID of the person who made the change. You can recreate the state of data in an audited table as it was at any point in time.

blind break

A single execution of a job on data defined as blinded in Oracle LSH, so that the sensitive blinded data is displayed in the job output, but is not permanently unblinded. Blind breaks may be required for patient medical emergencies or for reporting purposes during the course of a clinical trial.

blinded data

Data that must be hidden in Oracle LSH because its display would reveal patient treatment patterns in a double-blind clinical trial. If you specify that a table supports blinding, Oracle LSH creates a partition, allowing you to load the blinded data in the "real data" partition of the table and, optionally, to load dummy data in the "dummy

data" partition. Oracle LSH requires standard object security privileges to run a Program on dummy data but requires additional security privileges to run the same Program on real (blinded) data.

browse

Navigate through the user interface looking for an output or definitional object.

Business Area

Oracle LSH definitional object used by a visualization tool such as Oracle Business Intelligence Enterprise Edition to create ad hoc, onscreen visualizations of Oracle LSH data.

check constraint

A check constraint applies to a particular Table column and specifies allowed data values for that column.

child object

An object contained in, or owned by, another object.

classification

(1) The act of associating an Oracle LSH object definition or object instance with one or more values in one or more classification hierarchies, for the purpose of categorizing the object so that it can be found during searching and browsing. (2) The actual classification hierarchy value or set of values associated with an Oracle LSH object definition or object instance. (3) The Oracle LSH subsystem of single- and multi-level hierarchies and their values used in searching and browsing.

classification hierarchy

Hierarchical or flat structure consisting of one or more levels, with one or more values in each level. If a hierarchy contains multiple levels, they must be logically related and the values in each level must be logically related to values in adjacent levels (each value must be related to one and only one value in the next higher hierarchy level, but can be related to many values in the next lower level). Used in searching and browsing.

cloning

A special copy operation for Work Areas that results in an identical Work Area, including all its contained object instances and mappings. All object instances in the clone have the same version number and validation status as their counterparts in the original Work Area, and the two Work Areas share a user-specified label.

constraint

A constraint on an Oracle LSH Table or Column similar to an Oracle column or table constraint. See also [check constraint](#), [primary key constraint](#), and [unique key constraint](#).

Consumer

An Oracle LSH user who retrieves (consumes) information from the Oracle LSH database by running and viewing reports and creating onscreen data visualizations.

container object

An object that contains, or owns, another object; also called a parent object or organizational object.

Data Mart

(1) A large quantity of data extracted from one or more user-defined Oracle LSH Table instances in one of several formats. (2) The definitional object that generates the Data Mart output.

data source

An Oracle LSH Table instance or [source data system](#) table, view, or data set from which an Oracle LSH Program reads data.

data transformation

See [Program](#).

date data type

The Oracle date data type is defined as the number of days since the beginning of the Julian Calendar in a floating point numeric. Oracle LSH stores all date values in this format but displays dates and accepts entry of dates in one of several European, US or Standard formats (such as DD-MON-YYYY) based on user preference.

Definer

An Oracle LSH user who develops applications by defining objects in Oracle LSH, typically a programmer.

definition

See [object definition](#).

definitional object

Objects defined by the user in Oracle LSH, including Oracle LSH Tables, executables, and their components; includes both object definitions and object instances. See also object definition and object instance.

development environment

An installed Work Area whose usage intent is set to Development and whose database schemas are used for the purpose of developing Oracle LSH object definitions and instances. Oracle LSH supports maintaining separate Work Areas and schemas as development, quality control, and production environments so that data from a development or quality control environment is never mixed with production data.

Domain

A container that is used to group Application Areas and/or other Domains and/or to store and control a library of object definitions.

Domain library

A set of object definitions contained directly in a Domain; intended as a collection of valid, production-quality object definitions suitable for reuse.

dummy data

Alternative data with blinded information obfuscated that you can load, processed instead of the real (sensitive, blinded) data for the purpose of testing Programs and producing reports prior to unblinding the real data.

Entry

See [Report Set Entry](#).

execution

(1) The running of a particular Oracle LSH job; for example, running a Program to generate an output. (2) The Oracle LSH runtime subsystem that manages the execution of jobs.

Execution Setup

A defined object that is a component of each Oracle LSH executable object instance (Programs, Report Sets, Data Marts, Load Sets, Workflows) whose purpose is to control the execution of the executable object. It includes input Parameters whose values are either bound or settable at submission. The Execution Setup serves as the basis for the submission form of an executable object.

Execution Template

An Execution Setup that has been explicitly made available for use as a template to other instances of the same executable object definition version. Definers can use Execution Templates as the basis for an Execution Setup in another instance of the same executable object definition.

full reload processing

Processing mode for executables writing to Reload Tables in which the current state of all rows in a data source are reinserted by a Program or other executable object, and rows that are not reloaded are soft-deleted. Oracle LSH automatically detects which rows are new inserts, which are updates, which are unchanged and, implicitly, treats any rows that are not reloaded as deleted.

IDE

See [Integrated Development Environment \(IDE\)](#).

incremental reload processing

Processing mode for executables writing to Reload Tables in which only those rows that have been added or modified since the last time the same Program or other executable object was run are processed. Unchanged rows can also be reloaded and are ignored by the processing. Incremental reload processing does not delete records and therefore can be used to selectively load different subsets of data without impacting other data in the target table.

index

A non-unique index provides a way to optimize access to data in a table via columns included in the index. See also unique indexes [unique key constraint](#) and [primary key constraint](#).

instance

See [object instance](#).

installation

The process of converting selected Oracle LSH object instances in a single Work Area into installable components such as PL/SQL packages and DDL scripts and instantiating those components in an Oracle LSH schema.

Integrated Development Environment (IDE)

In Oracle LSH, an external source code development software application that can be launched directly from the Oracle LSH user interface for the purpose of writing or editing Program source code, integrated with Oracle LSH by a system that includes: a

syntax-directed editor, graphical tools for program entry, integrated support for compiling and running the program, and a mechanism for relating compilation errors back to the source.

library

See [Domain library](#) and [Application Area library](#).

Load Set

an Oracle LSH executable whose purpose is to move data and metadata into Oracle LSH from an external source data system that is connected to Oracle LSH by an adapter customized for the source data system. The adapter handles constraints on the structure of the target Table Descriptors, provides the Parameters, and supplies the source code for the installed Load Set.

mapping

(1) (noun) A mapping defines the relationship between a Table Descriptor in a Program or other executable object and a Table instance in a Work Area that the executable object reads from or writes to. Each column of the Table Descriptor and the Table instance must be mapped to a column of a compatible data type and length in the other, or to a constant. (2) (verb) Creating a mapping.

master job

A job launched by the submission of an Execution Setup. When a user submits a Load Set, Program, Data Mart, Report Set, or Workflow, the resulting job is a master job. Some master jobs have subjobs: Report Set executions include Program executions as subjobs. Workflow executions may include Program, Load Set, Data Mart, and/or Report Set executions as subjobs.

narrative

A block of text that annotates a generated report or report set. May precede the body of the report (pre-narrative) or follow it (post-narrative).

Notification

Message routed within Oracle LSH or (optionally) by email; contains a subject name, text, and one or more hyperlinks to Oracle LSH outputs or other objects; can be used for collecting informal approvals or comments or for broadcasting text.

object definition

The definitional metadata details of an Oracle LSH object, stored in an Application Area or Domain library and created (normally) through an instance of the definition that is contained in a Work Area. The metadata varies depending on the definition's object type, but includes a name, subtype, validation status and classification(s). For example, Table definitions also contain Columns and Constraints; Program definitions contain Source Code, Parameters, Planned Outputs, and Table Descriptors.

object instance

An instance of Oracle LSH object definition; contains a pointer to an object definition, which contains most of the definitional details, and a few attributes including a name, subtype, validation status, classification(s), and (for executable instances only) mappings. Object instances are located in a Work Area and installed in the database.

object subtype

Subcategory of a predefined object type that serves to differentiate classification and security requirements among objects of the same object type; for example, Financial

Report Set and Clinical Report Set. Object subtypes have the same operations allowed as their object type. One subtype for each object type is shipped with Oracle LSH; an Administrator may define additional object subtypes.

object type

A predefined category of Oracle LSH definitional objects shipped with Oracle LSH; for example, Report Sets or Tables. Object instances constitute a separate object type; that is, a Report Set instance is a different object type from a Report Set definition and can have different classification and security requirements defined at the subtype level.

operation

An action that can be performed by a user on an object; for example, view or modify. Operations are predefined for object types. The Oracle LSH Administrator defines roles and associates them with operations on object subtypes as part of the Oracle LSH security system.

Oracle LSH schema

A set of Oracle database schemas (one primary schema and one or more auxiliary schemas) that owns all permanent objects that are populated by the installation of a single Oracle LSH Work Area, including tables, views, packages, or schema stores. An Oracle LSH schema also contains all data loaded into its tables.

organizational object type

Category of Oracle LSH object types used solely as containers of other objects; includes Domains and Application Areas.

output

In Oracle LSH, a file generated by the execution of a Program, Workflow, Report Set, or Data Mart; either the primary output reporting on or containing data, or a secondary output such as a log file or an error file. Also called an actual output to distinguish it from a Planned Output definition.

Overlay Template Definition (OTD)

A layout template for PDF outputs of Oracle LSH Report Sets; can include boilerplate text, graphical elements such as logos and lines, and placeholders for Parameter values for data (such as Study) and publishing specifications (such as font style and size).

Parameter

A defined object that acts as a simple scalar variable and is based on an Oracle LSH Variable definition. In a Program or other Oracle LSH executable object, an input Parameter supplies a runtime value to the executable object and an output Parameter holds a value generated during execution. In addition to a pointer to an Oracle LSH Variable, a Parameter contains a list of allowable values, rules for validating the supplied value, a mandatory/not mandatory setting, and classification(s). Parameter values can be propagated within the context of a Report Set or a Workflow.

Parameter Set

A definitional object created automatically by Oracle LSH to contain the Parameters contained in a single primary definitional object.

parent object

An object that contains, or owns, another object; also called a container object.

Planned Output

An object definition contained in an Oracle LSH executable object definition that serves as a placeholder for an actual output to be generated during execution and specifies the classification(s) of the actual output. Primary Planned Outputs report on or contain data; secondary Planned Outputs include log files and error files.

primary key constraint

A column or set of columns whose value(s) identify a row in a Table as unique; often a unique ID. The system enforces the following constraints on the data contained in primary key columns: values cannot be null, and the value or combination of values in primary key column(s) must be unique for each row.

privilege

In Oracle LSH, a privilege is an operation on an object subtype assigned to a user through a role in a user group. A user has the privilege necessary to perform an operation on an object when he or she is assigned to a user group that has access to the object, and within that user group is assigned to a role that allows the operation on the object's subtype.

production environment

An installed Work Area whose usage intent is set to Production and whose database schemas contain only valid Oracle LSH objects interacting with real clinical or other data in an audited and regulatory-compliant manner.

Program

an Oracle LSH defined executable object that functions as a computer program to transform data and/or generate one or more reports. Contains Source Code and one or more Table Descriptor(s); may also contain one or more Planned Outputs and Parameters.

quality control (QC) environment

An installed Work Area whose usage intent is set to Quality Control and whose database schemas are used for the purpose of testing Oracle LSH object definitions, adapters, and loads of data and metadata before using them in a production environment. Oracle LSH supports maintaining separate Work Areas and schemas as development, quality control, and production environments so that data from a development or quality control environment is never mixed with production data.

report

(1) The primary output generated by a Program; contains a set of clinical or other data for in-house review or submission to regulatory agencies. Reports may be displayed as listings, figures, tables, or text and may be generated in one of several file formats. (2) A predefined output generated by the system on demand; for example, a coversheet for a report.

Report Set

(1) A collection of reports generated by a single execution process and integrated with a table of contents. May be generated in PDF format using custom templates with graphics and including bookmarks and hyperlinks. (2) The primary Oracle LSH definitional object that produces the Report Set output.

Report Set Entry

One of the items in the table of contents of a Report Set; may contain other Report Set Entries, narratives, and/or one or more reports generated by a single Program; a Report Set chapter or subchapter.

role

(1) User-defined component of the Oracle LSH security system corresponding to a professional title or function; assigned to operations on object subtypes, to user groups, and to individual users within those groups. To perform a particular operation on an Oracle LSH object, a user must have a role assigned to that operation for the object's subtype in a user group with access to the object. (2) Predefined application roles shipped with Oracle LSH that control access to portions of the Oracle LSH user interface.

runtime

The point in time when a user submits an executable object for execution.

runtime subsystem

The Oracle LSH subsystem that manages the execution of installed executable object instances as well as the tracking of, and access to, the progress and results of the executions.

search

Facility for finding existing defined objects and outputs in Oracle LSH, using criteria including object type, name, and classifications (including keywords). Oracle LSH includes simple and advanced search options.

snapshot

A set of data, or data and objects, at a particular point in time.

Source Code

(uppercase) A secondary definitional object owned by an Oracle LSH Program containing a file of user-written computer instructions ([source code](#)) whose purpose is to perform a task.

source code

(lowercase) The computer instructions used by a Program to perform a task. Source code may be written in an Integrated Development Environment (IDE) in the context of an Oracle LSH Program or written outside Oracle LSH and then uploaded and stored in Oracle LSH; in Oracle LSH, source code is stored in the definitional object called [Source Code](#).

source data system

An application external to, but integrated with, Oracle LSH that collects data loaded into or accessed by Oracle LSH through an adapter.

Source Table Descriptors

A [Table Descriptor](#) used to link a Program to a Table instance containing data used as input to the Program.

staging table processing

Data processing type used for Programs that write to Tables defined as Staging. If the Table is defined as Not Audited, each time the Program that writes to it is executed,

the system hard-deletes the data resulting from the previous execution. If the Table is defined as Audited, the system saves a copy of the complete set of data resulting from the previous execution, but operates only on current data.

structure

See [Workflow Structure](#) and [Table structure](#).

subjob

A subjob is a job that is executed as part of a master job. For example, Report Set executions are master jobs that include Program executions as subjobs, and Workflow executions are master jobs that can include Load Set, Program, Report Set, and/or Data Mart executions as subjobs.

subtype

See [object subtype](#).

Table

an Oracle LSH metadata representation of a table-like object (Oracle view or SAS dataset); includes a description of the object as a whole, its columns, and constraints that apply to one or more of its columns. Appears as an Oracle view to all installed Programs except those based on SAS technology, where it appears as a SAS dataset or SAS view.

Table constraint

See [constraint](#).

Table Descriptor

An instance of an Oracle LSH Table defined within an Oracle LSH executable object—Load Set, Program, Workflow, Report Set, or Data Mart—or in a Business Area, for the purpose of linking the executable object or Business Area to an installed Table and its data by mapping.

Table structure

The number of Columns a Table definition contains and the data type and length of each one.

Target Table Descriptors

A [Table Descriptor](#) used to link a Program to a Table instance to which the Program writes data.

test environment

See [quality control \(QC\) environment](#).

transform, transforming Program

See [Program](#).

transactional processing

Data processing type used to write to Table instances defined as transactional. Programs writing to transactional Tables must use explicit Insert, Update, and Delete commands in their source code.

transition

An element of a Workflow that defines one workflow activity as sequential to another and specifies the condition, if any, for the execution of the second.

unblind

Change a Table instance's blinding status from Blinded to Unblinded.

unique key constraint

A column or combination of columns that identifies a row in a Table as unique; similar to a primary key except that null values are allowed in columns that are part of the unique key. You can explicitly disallow null values at the column level.

usage intent

An attribute of Work Areas indicating the Work Area's purpose; either Development, Quality Control, or Production. Oracle LSH enforces rules based on the interaction of a Work Area's usage intent attribute and the validation status of the Work Area and the object instances contained in it.

user group

Unit defined by an Oracle LSH Administrator for use in security that includes one or more users and one or more roles, and is assigned to one or more Oracle LSH objects. To perform an operation on an Oracle LSH object or output, a user must belong to a user group assigned to that object and be assigned a role within that user group that permits the operation on the object's subtype.

validation

(1)Process through which object definitions and instances reach the stage where they can be declared stable and valid according to an Oracle LSH customer's policies. (2) Process required by regulatory agencies to certify that a computer system used in clinical trials is stable and functions properly.

validation status

An object attribute that indicates the life cycle stage of the object: Development, Quality Control, Production, or Retired. Oracle LSH enforces rules based on the interaction of a Work Area's usage intent attribute and the validation status of the Work Area and the object instances contained in it.

Variable

an Oracle LSH defined object equivalent to a SAS variable or Oracle table column that serves as a source definition for Oracle LSH Parameters and Table Columns. Oracle LSH creates Variables from Oracle table columns and SAS dataset variables the first time a particular Oracle table or SAS dataset is loaded into Oracle LSH. The user can define Variables during Parameter or Table Column definition.

version

Restorable, saved state of an object definition.

visualization

An ad hoc graphical and/or text presentation of data created in an interactive onscreen environment.

Work Area

A container in an Application Area where Definers create the object instances required to support the business purpose of the Application Area. Work Areas have two special operations: **installation**, during which some or all of the object instances contained in the Work Area are instantiated or upgraded in an Oracle LSH schema and the Oracle LSH schema is created if necessary; and **cloning**, which creates a duplicate Work Area that shares a label with the original Work Area.

Workflow

An executable Oracle LSH defined object that includes other Oracle LSH executable defined objects such as Load Sets, Programs, Data Marts, and Report Sets, as well as Notifications; executed as a whole in a defined sequence that can include conditional branching.

Workflow Structure

Predefined object that controls the execution of the Workflow; a Workflow Structure can detect the completion status of the previous activity and fire the next one or more activities as specified for their type: Start, End, And, Or, Fork.

XML Publisher

Oracle product integrated with Oracle LSH for use in generating PDF-format Report Sets and enabling the use of custom overlay templates. Also used to produce internal PDF-format reports.

A

adapters
 creating, 2-4
 overview, 2-1
 security, 2-4
 structure, 2-2
assigning repository filenames, 11-6
assigning subtype operations, 9-12
audited tables, 6-2

B

BIP endpoints, 3-11

C

classification
 overview, 8-1
clearing Oracle Applications cache, 2-6
codelists, 4-1
configuration settings, 3-1
connections, 5-1
 defining, 5-4
 searching, 5-5
creating data sources, 12-5
creating database accounts, 10-2
creating hierarchies, 8-3
custom validation package, modifying, 14-1
customizing object validation, 14-1

D

data security, 11-10
database accounts, 10-1
 creating, 10-2
 for definers, 10-1
 for submitting messages, 10-2
 removing, 10-4
database core dumps, 19-3
DBA privileges for GV BAs, 10-3
default authenticator for 11g, 11-9
default values
 output validation status, 3-5
defining service locations, 1-3
defining services, 1-4
direct SQL options for SAS, 3-6

Distributed Processing Server
 checking, 18-3
 job processor, 1-2
 listener, 1-2
 overview, 1-1
 setting up, 1-3
 starting, 18-2
 stopping, 18-4

E

E-Business suite responsibilities
 creating, 12-2
enabling job trace files, 19-4
excluded objects, 17-3
execution command scripts
 editing, 11-8
execution statuses, 16-7
execution, overview, 16-1
exporting objects, 17-4

F

failing report set jobs, 19-3

G

gathering statistics, 3-3
granting access to APIs, 9-16

H

hierarchies, 8-2
 activating, 8-4
 adding levels, 8-3
 adding terms, 8-5
 assigning levels, 8-6
 creating, 8-3
 modifying, 8-5

I

importing objects, 17-8
included objects, 17-2
Informatica
 creating remote connections, 13-3

- creating remote locations, 13-2
- Informatica administrator accounts, 13-1
- installing objects, 17-11
- internal table auditing, 6-1

J

- job execution details, 16-4
- job execution problems, 19-1
- job failing to obtain service, 19-3
- job information, 16-3
- job log files not containing information, 19-3
- job outputs not uploading, 19-2
- job queue
 - enabling, 18-5
 - starting, 18-5
- job statuses, 16-7
- jobs stuck at submitted or executing status, 19-1

L

- log-in attempts, 3-11
- lookups, 4-1
 - modifying values, 4-4
 - querying and viewing, 4-3

M

- message-triggered submission
 - starting, 18-6
 - stopping, 18-6
- modifiable lookups, 4-1
 - logging in, 4-3
- modifying lookup values, 4-4
- multiple-computer configuration, 11-3
 - requirements, 11-4

N

- nested domains, 3-5

O

- object export and import overview, 17-1
- object security, 9-9
- object subtypes
 - creating, 9-9
 - creating roles, 9-11
 - modifying, 9-10
 - modifying roles, 9-12
 - security roles, 9-11
- Oracle Applications, 12-2
 - roles, 12-3
- Oracle Applications profile
 - logging in, 3-1
- Oracle Applications responsibilities, 12-2
- Oracle BI admin tool, 11-10
- Oracle BI Publisher
 - creating remote connections, 12-6
 - creating remote locations, 12-6
- Oracle BIP Endpoint profile, 12-7

- Oracle BIP integration
 - overview, 12-1
- Oracle BIP security, 12-4
 - creating the local Superuser account, 12-4
- Oracle BIP security model, 12-4
- Oracle LSH Definer accounts, 13-1
- Oracle Warehouse Builder Service
 - not starting, 19-2
 - starting, 18-4
 - stopping, 18-4
- output validation status, 3-5

P

- particular object types, 9-13
- presentation server URLs, 11-12
- preserving customizations, 17-13
- printers
 - defaults, 7-1
 - setting up, 7-1

Q

- query exceeding 200 rows, 19-3
- querying for jobs, 16-1

R

- reference codelists, 4-1
- remote locations, 5-1
 - defining, 5-2
 - searching, 5-5
 - security, 5-2
- removing a database account, 10-4
- repository files, 11-6
- resetting DBA passwords, 10-4
- RPD password security, 11-11
- running the installation script, 17-12

S

- security
 - overview, 9-1
- service types, 1-8
- services
 - defining, 1-4
 - instances, 1-9
 - overview, 1-1
 - prioritization, 1-9
 - types, 1-8
- setting native client encoding, 3-2
- setting passwords, 3-8
 - case sensitivity, 3-9
 - forgotten passwords, 3-10
 - password length, 3-9
 - requirements, 3-10
- setting up database connectivity, 11-6
- setting up TMS security, 10-4
- single-computer configuration, 11-1
- source independence, 17-4
- specialized administrators, 9-7

- starting the WebLogic server for 11g, 11-9
- Stopping the Job Queue, 18-5
- subtype operations
 - assigning roles, 9-12
 - modifying assignments, 9-13
- synchronizing apps password changes, 15-1
- system configuration, 3-1

T

- Table auditing problems, 19-3
- time limits on password reuse, 3-11
- troubleshooting, 19-1

U

- user accounts, 9-3, 13-1
 - assigning roles, 9-5
 - creating, 9-4
 - maintaining, 9-4
- user groups, 9-13
 - adding supported roles, 9-14
 - assigning an administrator, 9-15
 - creating, 9-14
 - duplicating, 9-15
 - modifying, 9-16
- user preferences for SAS clients, 3-6
- using character semantics, 3-2

W

- work area table installation, 3-2

