Oracle Life Sciences Data Hub

Installation Guide

Release 3.4.1
G12930-01
September 2024

Oracle Life Sciences Data Hub Installation Guide, Release 3.4.1

G12930-01

Contents

6 Integrating Other Systems

7 Supported Upgrade Paths

8 Upgrading the Thesaurus Management System (TMS) Database to Release 5.4.1

9 Upgrading to Oracle LSH Release 3.4.1

Preface

This preface contains the following sections:

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.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related resources

All documentation and other supporting materials are available on the Oracle Help Center.

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.

1

Before You Begin

If you are upgrading from a previous release, you may only need to check for the most recent information, assemble the software, and upgrade to 3.4.1.

This section includes the following topics:

Check for the Most Recent Information

It is critical that you check that you have the most current information before you begin the installation process.

- **Latest Release Notes**: For the most recent version of the Release Notes, see Oracle Life Sciences Data Hub and Oracle Life Sciences Data Management Workbench Release Notes (Document ID 2982152.1) on My Oracle Support.
- **Known Installation and Configuration Issues**: For up-to-date information, see Oracle Life Sciences Data Hub, Oracle Life Sciences Data Management Workbench, and Oracle Clinical Development Analytics Known Install and Configuration Issues (document ID 1138053.1) on My Oracle Support.
- **Latest Critical Patch Updates and Technology Stack Updates**: To get the latest information on the technology stack and for the latest quarterly Oracle Critical Patch Update (CPU) certified with Oracle LSH, see *Oracle Life Sciences Applications Supported Technology Stacks* (Document ID 180430.1) on My Oracle Support. This CPU patch includes security fixes and should always be up-to-date.

Get Your Company ID from Oracle

When you install Oracle LSH, you need to enter a parameter value for the company ID. The company ID serves as part of the primary key for all the Oracle LSH objects you define in this instance of Oracle LSH. If your company ever merges with another company and your Oracle LSH data and metadata are merged with the data of another company, the company ID distinguishes the objects created in each original company and prevents duplicate object primary keys.

To ensure that you have a unique number relative to other Oracle LSH customers, Oracle recommends that you use a number assigned to you by Oracle. Company IDs are tracked in an Oracle bug that is not publicly viewable. Contact Oracle Support or ask your consultant to reserve a range of ten numbers for your company. You can use these numbers for:

- Your company ID. If you have multiple instances of Oracle LSH, Oracle recommends using a different company ID for each instance.
- The tech type ID of any adapters your company may create.

Assemble the Software

Use the technology stack product versions mentioned in this document. Even if newer versions of the technology stack products become available, they may not be compatible with Oracle LSH.

Download the Oracle LSH Folder

Oracle LSH, Oracle Life Sciences Data Management Workbench (Oracle DMW), and their technology stacks are contained on the **Oracle Life Sciences Data Management Workbench 3.4.1** media pack for various platforms.

To receive a physical media pack with all the required DVDs, contact Life Sciences Support. To expedite your request, you can call Oracle Support directly or open a Service Request (SR) selecting problem category: **Version Update Request**.

To download the media pack from eDelivery:

1. Go to Oracle Software Delivery Cloud, http://edelivery.oracle.com, click **Sign In**, and log in with your user ID.
2. Select **Download Package** from the **All Categories** drop-down list (or leave All Categories selected). Enter **Oracle Life Sciences Data Management Workbench** in the Search field and click **Search**.
3. Select **DLP: Oracle Life Sciences Data Management Workbench 3.4.1.0.0** and click **Add to Cart**.
4. Click **Checkout**. You see a list of the selected software:
   - Oracle Life Sciences Data Management Workbench 3.4.1.0.0 (Oracle Standard Terms and Conditions)
   - Oracle Life Sciences Data Hub 3.4.1.0.0
   - Oracle Life Sciences Data Management Workbench 3.4.1.0.0
5. From the **Platform** drop-down list, select the appropriate operating system.
6. Click **Continue**.
7. Review the Terms and Restrictions and select **I reviewed and accept the Oracle License Agreement** to continue. (Click **Print** from the top-right corner of the screen to print the agreement.) Click **Continue**. You see a list of zipped files for the Oracle Life Sciences Data Hub <*your operating system*> 3.4.1 release and Oracle Life Sciences Data Management Workbench <*your operating system*> 3.4.1 release:
   - Oracle Life Sciences Data Hub 3.4.1
   - Oracle Life Sciences Data Management Workbench 3.4.1
8. Leave the list of zipped files selected to download the package of Oracle Life Sciences Data Management Workbench 3.4.1.0.0 files or only select the files you need.
9. Click **Download**. Then browse to the location where you want to save the Oracle executable.
10. Double-click the Oracle executable. Leave the default destination or click **Browse** to select another one. Click **Next**. Oracle downloads the zipped files.

11. Move the zipped files to a staging area and unzip them. The full release contains a software folder for Oracle DMW (**p36251041_34100_Generic.zip**) and Oracle LSH (**p36272413_R12_GENERIC.zip**).
12. See Download Software to a Staging Area for details on downloading the software.

Download Software to a Staging Area

Creating a staging area is recommended, but not mandatory.

To set up the staging area successfully, create a directory for each disk in the media pack and then download and expand all the files that comprise a single disk in the media pack to the same location.

See the spreadsheet on the documentation disk for a list of patches and their location on the media pack.

To download patches from My Oracle Support, go to `https://support.oracle.com`.

> **Note:**
>
> See My Oracle Support article 1138053.1, *Oracle Life Sciences Data Hub and Oracle Clinical Development Analytics Known Install and Configuration Issues* for the latest information.

Table 1-1 Software to Download for Oracle LSH and Oracle DMW

Disk or Patch Name	Source	ID Number
Oracle E-Business Suite Release 12.2 software distribution	Media pack	15 disks
Oracle Database 19c for Linux	Media Pack	V982063-01
Oracle Thesaurus Management System 5.4.1	eDelivery	For information on how to download the Oracle TMS 5.4.1 media pack from eDelivery, see *Oracle Thesaurus Management System Installation Guide* for release 5.4.1.
Oracle Life Sciences Data Hub 3.4.1	Media Pack	36272413
Oracle E-Business Suite SDK patch	Media Pack	27723788
NOT ABLE TO ADD HTTPS URL TO FAVORITES LINK IN 12.1.2. HTTP IS PRE-PENDED TO URL	My Oracle Support	11781531
FND_NO_DATABASE_CONNECTION	My Oracle Support	11832737
Oracle Database 19c; for a list of patches required, see My Oracle Support article 1058763.1.	My Oracle Support	2580629.1, 2530680.1, and 2617850.1

If you use Oracle Life Sciences Data Management Workbench, download the following software to a separate application server. For system requirements, see the *Oracle Life Sciences Data Management Workbench Installation Guide*.

Table 1-2 Software to Download for Oracle DMW

Disk or Patch Name	Source	ID Number
Oracle WebLogic Server 12.2.1.4 and Coherence for Linux x86	Media Pack	p30188255_122140_Generic.zip
ADF patch	My Oracle Support	32588679
Oracle Life Sciences Data Management Workbench 3.4.1	Media Pack	36251041

Assemble the Documentation

Installing Oracle LSH is a complex process because it includes installing, upgrading, and patching a number of other products. This book guides you through the process, but refers you to the documentation for other products along the way.

Oracle recommends that you gather all the documentation you will need and read it before you begin the process of installing Oracle LSH.

- Books
- My Oracle Support Articles

Books

The books you need to install the technology stack are included in the media pack.

You can also find PDF and HTML copies online; see Documentation accessibility.

In addition to this guide, you need:

- *Oracle Life Sciences Data Hub System Administrator's Guide Release*
- *Oracle Thesaurus Management System Installation Guide Release 5.4.1*

The rest of the Oracle LSH user documentation is also included. However, you should check Oracle Help Center for the most current versions.

My Oracle Support Articles

The My Oracle Support Web site contains links to the most recent patches and updates for Oracle products. This section lists all the My Oracle Support articles listed in subsequent sections of this guide so that you can copy all of them to one place. See Documentation accessibility.

- System Requirements and Technology
- Upgrading to Oracle Life Sciences Data Hub Release 3.4.1
- Other Document Related to Oracle Life Sciences Data Hub

System Requirements and Technology

System Requirements and Technology Stack references the following My Oracle Support articles:

- 180430.1, *Oracle Life Sciences Applications Supported Technology Stacks*

Upgrading to Oracle Life Sciences Data Hub Release 3.4.1

Upgrading to Oracle LSH Release 3.4.1 references the following My Oracle Support article:

- 1320300.1, *Oracle E-Business Suite Release Notes, Release 12.2*
- 2495027.1, *Oracle E-Business Suite Release 12.2.9 Readme*
- 396009.1, *Database Initialization Parameters for Oracle E-Business Suite Release 12*
- 2580629.1, *Interoperability Notes: Oracle E-Business Suite Release 12.1 with Oracle Database 19c*
- 2530680.1, *Using Oracle 19c RAC Multitenant (Single PDB) with Oracle E-Business Suite Release 12.1* (for an Oracle RAC environment)
- 3019034.1, *How To Apply Patches for LSH/DMW in Downtime Mode*

Other Document Related to Oracle Life Sciences Data Hub

The following related document is available on My Oracle Support:

- *LSH: Life Sciences Data Hub (LSH) Summary of Patches Available (Document ID 1376925.1)*

2

System Requirements and Technology Stack

If you install Oracle LSH for the first time or upgrade to a new version after December 31, 2010 and are using Oracle Enterprise Manager (OEM) 10.2.0.4 or 10.2.0.5 with Oracle LSH, apply OEM patch 8350262.

For further information, see My Oracle Support article number 1217493.1.

This section contains the following topics:

System Requirements

This section includes some general requirements for your Oracle Life Sciences Data Hub installation. For requirements on other products you need to install, see the documentation that came with them.

The general requirements topics include:

Operating Systems

To get the latest information on the technology stack, see *Oracle Life Sciences Applications Supported Technology Stacks* (Document ID 180430.1) on My Oracle Support.

This section includes the folllowing topics:

Database Tier

The Oracle LSH database tier can be installed on the following platforms:

> **Note:**
>
> To get the latest information on the technology stack, see *Oracle Life Sciences Applications Supported Technology Stacks* (Document ID 180430.1) on My Oracle Support.

- Linux x86-64 (64-Bit):

  - Oracle Enterprise Linux 7.4 or later, and 8.x
  - Red Hat Enterprise Linux 7.4 or later, and 8.x

> **Note:**
>
> The database global name cannot be greater than 64 characters.

Application Tier

You can install the Oracle LSH application tier on the following platforms:

> **Note:**
>
> To get the latest information on the technology stack, see *Oracle Life Sciences Applications Supported Technology Stacks* (Document ID 180430.1) on My Oracle Support.

- Linux x86-64 (64-Bit):
  - Oracle Enterprise Linux 7 or 8
  - Red Hat Enterprise AS/ES 6.x, 7.x, or 8.x

Clients

Oracle LSH supports the following browsers on Microsoft Windows operating systems:

- Google Chrome Version 109.0.5414.120 (Official Build) (64-bit)
- Mozilla Firefox Quantum Extended Support Release 102.7.0esr (64-bit)
- Mozilla Firefox Version 109.0.1 (64-bit)
- Microsoft Edge Chromium Version 111.0.5500.0 (Developer Build) (64-bit)

For **Oracle JRE**, Oracle LSH supports the same versions as Oracle E-Business Suite 12.2.9. To get the latest information:

1. Go to My Oracle Support at `https://support.oracle.com` and sign in.
2. Click the **Certifications** tab.
3. In the Search area, enter `Oracle E-Business Suite` for Product and `12.2.9` for Release, and click **Search**.
4. In the Search Results page, expand **Management and Development Tools**.
5. Check the Oracle JRE versions displayed and click the link to see more.

Hardware

Oracle Applications 12.2.9 and Oracle Database 19c can be installed on the same or different servers.

In addition, you need one computer running on Windows for use in installing Oracle Thesaurus Management System (Oracle TMS). You will not need this computer for Oracle LSH after installing Oracle TMS except to install any Oracle TMS patches that may be required in the

future. You need at least one Windows computer if you plan to use Oracle Analytics Server to define Business Areas and create data visualizations (Oracle Analytics Server Answers). You can use the same Windows computer for Oracle Analytics Server and Oracle TMS.

> **Note:**
>
> Oracle LSH does not support a Windows server for SAS.

In an installation where the application and database tiers are installed on different computers, those computers are typically connected by a local area network (LAN), while the application-tier computer is connected to clients in a wide area network (WAN).

Using Real Application Clusters (RAC) to install the database over several nodes is optional.

Oracle LSH uses Oracle XML Publisher (which is bundled with Oracle Applications) to generate PDF-format Report Sets. If you plan to use this feature and if your Report Sets are very large, you may want to dedicate one node to Oracle XML Publisher processing.

For system hardware requirements see the documentation for each component; see Assemble the Documentation.

In addition:

- The TCP/IP network connection to the server should be at least at 1 GB.
- Be sure to allow for growth in database storage capacity.
- Oracle recommends installing a test environment as similar as possible to the production environment, including all operating system and other patches and updates.

Technology Stack

To get the latest information on the technology stack, see *Oracle Life Sciences Applications Supported Technology Stacks* (Document ID 180430.1) on My Oracle Support. At the time of publication of this document, the required technology stack for Oracle LSH consists of the following products:

- **Oracle Applications 12.2.9**

  > **Note:**
  >
  > **Oracle Life Sciences Data Hub has been tested ONLY on Release 12.2.9 of Oracle Applications. Do not install more recent releases unless explicitly instructed to do so by an Oracle Life Sciences Data Hub note or alert on My Oracle Support.**

- **Java Development Kit (JDK) 1.8.0_281** is required for the Oracle LSH Distributed Processing Server.
- **Oracle Database 19c**
- **XML DB**, which is included with the 19c database, is required for Oracle LSH.
- **Oracle Thesaurus Management System 5.4.1 Database Tier** is used internally for the Oracle LSH classification system.

- A **zip utility** and **Java Development Kit (JDK) 1.8.0_281** are required for the Oracle LSH Distributed Processing Server.
- **WinZip** or **7-Zip** is required on clients used by Oracle LSH developers who launch integrated development environments (IDEs) such as SAS or the Oracle BI Administration Tool on their PC. Neither of these utilities is included on the media pack.
  For WinZip, use Pro 11.2 SR-1, WinZip 8.1, or any other WinZip version that includes WZUNZIP.exe.

Integrated External Systems

Oracle LSH includes adapters to support integration with the following external systems.

- Oracle Clinical 5.2.2, 5.4, and 5.4.1
- SAS 9.1.3, SAS 9.2, SAS 9.3, and SAS 9.4: Optional and licensed separately.
- Oracle Business Intelligence Enterprise Edition (OBIEE) 12.2.1.4 can be used to create OBIEE visualizations of Oracle LSH data. Optional and licensed separately. The OBIEE Presentation Server, OBIEE Server, and the OBIEE Administrator's Tool are required if you are using Oracle Analytics Server. The latter two run on Windows only.

3

Installing and Patching Oracle Applications and Oracle Database

This section includes the following topics:

Install Oracle Applications 12.2.0

To install Oracle Applications 12.2.0, see https://docs.oracle.com/cd/E26401_01/doc.122/e22950/T422699i4773.htm#3708191 and *Oracle E-Business Suite Release Notes, Release 12.2* (Doc ID 1320300.1).

> **Note:**
>
> **Oracle LSH has been tested ONLY on Release 12.2.9 of Oracle Applications. If more recent releases have become available, do NOT install them unless explicitly instructed to do so by an Oracle LSH-specific note or alert on My Oracle Support.**

Oracle LSH UTF8 Requirements

Oracle LSH has the following character set-related required settings:

- **Database character set=UTF8**. Oracle Applications Rapid Install prompts you for the database character set. **You MUST set this value to UTF8. A value of UTF8 is REQUIRED. This is the ONLY opportunity you have to set this value, and you CANNOT change it later**.
- **NLS_CHARACTERSET=UTF8**. When you set the database character set to UTF8 during Oracle Applications Rapid Install, it automatically sets NLS_CHARACTERSET to UTF8 as well, which is correct.
- **NLS_LENGTH_SEMANTICS=BYTE**. The default value for NLS_LENGTH_SEMANTICS is BYTE. **Do not change this value**. To use character semantics, set the LHS profile Use

Character Semantics for Workarea Installation to **Yes**. See the chapter on setting profile values in the *Oracle Life Sciences Data Hub System Administrator's Guide* for information.

Upgrade Oracle Applications from 12.2.0 to 12.2.9

To upgrade Oracle Applications from 12.2.0 to 12.2.9, see *Oracle E-Business Suite Release 12.2.9 Readme* (Doc ID 2495027.1).

> **Note:**
>
> After upgrading Oracle Applications from 12.2.0. to 12.2.9, log on to Oracle Applications 12.2.9 as sysadmin. Under the Secure Configuration Console, DO NOT enable the "Hashed Passwords" security guidelines. For more information, see *LSH Is Not Certified with EBS Non-Reversible Hash Password FNDCPASS* (Doc ID 2837193.1) on My Oracle Support.

Upgrade the Oracle Database from 12c to 19c

To use Oracle LSH, you must use Oracle Database 19c.

To upgrade to Oracle Database 19c, see *Interoperability Notes: Oracle E-Business Suite Release 12.2 with Oracle Database 19c* (Doc ID 2552181.1).

> **Note:**
>
> Oracle recommends you to take a full backup after the upgrade.

Perform Oracle Database 19c Post-Upgrade Tasks

After you upgrade the Oracle Database to 19c, perform the following tasks.

- Apply Patches for Oracle Database 19c
- Update PDB GLOBAL_NAME to Uppercase
- Unset optimizer_features_enable
- Set FILESYSTEMIO_OPTIONS to SETALL
- Check job_queue_processes
- Set Parameter Values
- Create New Indexes
- Set the PL/SQL Service Instances Value
- Gather Statistics
- Compile Invalid Objects

Apply Patches for Oracle Database 19c

Verify if the following patches are applied on Oracle Database 19c. If there is any patch that has not been applied already, apply it on Oracle Database 19c.

- Patch 30241807: Fixes the DROP and REVOKE privilege issue. See "Patch 30241807: SELF DEADLOCK FROM DROP USER" on My Oracle Support for details.
- Patch 30937410: Fixes blocked sessions issue. See "Patch 30937410: SESSIONS BLOCKED WAITING FOR GC WAITS FOR A LONG TIME" on My Oracle Support for details.
- Patch 31142377: Fixes concatenation of null CLOB issue. See "Patch 31142377: CONCATENATION OF A NULL CLOB WITH NON-NULL DATA YIELDS WRONG RESULTS WHEN DEST. LOB IS PART OF THE CONCATENATION" on My Oracle Support for details.
- Patch 31142749: Provides OPatch support for RAC environments. See the Readme that came with the patch and document ID 244241.1, *Rolling Patch - OPatch Support for RAC* on My Oracle Support.
- Patch 29252510: Fixes performance issue using SQL*Loader in EBR environments. See "Patch 29252510: ZACN-ESS JOB - PERFORMANCE ISSUE USING SQL*LOADER IN EBR ENVIRONMENTS" on My Oracle Support for details.
- Patch 30392987: Fixes performance issue. See "Patch 29252510: EXECUTE IMMEDIATE WITH AUDITING HAS A LARGE PERFORMANCE IMPACT" on My Oracle Support for details.
- Patch 30233934: Fixes issues with flashback query in PDB failures, incorrect rows returned, and canceling running queries. See "Patch 30233934: QUERY FROM DBA_IND_STATISTICS RUNS SLOW ON RAC" on My Oracle Support for details.
- Patch 32940955: Fixes shared pool memory issue. See "Patch 32940955: DUE TO LARGE "SO PRIVATE SGA" ALLOCATION IN ONE SHARED POOL SUBPOOL (Patch) on My Oracle Support for details.
- Patch 31463613: See "Patch 31463613: ORA-00918: COLUMN AMBIGUOUSLY DEFINED IN 19C DATABASE
- Patch 30808109: See "Patch 30808109: ORA-38802 WHILE COMPILING APPS.FND_GLOBAL.

Update PDB GLOBAL_NAME to Uppercase

To support Oracle Thesaurus Management 5.4.1, update the PDB GLOBAL_NAME in Oracle Database 19c to uppercase.

1. Log in to Oracle Database 19c.
2. Enter:

   `alter session set container="<PDB NAME>"`

   For example: `alter session set container="lsw3qa5"`
3. Enter the customer-created pluggable database (PDB) name and domain in uppercase letters:

   `update GLOBAL_NAME set GLOBAL_NAME='<PDBNAME.DOMAIN>'`

   For example: `update GLOBAL_NAME set GLOBAL_NAME='LSW3QA5.US.ABC.COM'`

> **Caution:**
>
> Do not use spaces or special characters (for example, *, &, @, %) in the global name or you cannot connect to or recover the database.

Unset optimizer_features_enable

To unset optimizer_features_enable:

1. Log in to Oracle Database 19c.
2. Unset optimizer_features_enable.
3. Restart the database.

Set FILESYSTEMIO_OPTIONS to SETALL

In an environment without Oracle Exadata Database systems and Oracle Automatic Storage Management (ASM), you see performance issues if you do not set the FILESYSTEMIO_OPTIONS to SETALL.

1. Log in to Oracle Database 19c.
2. Check that FILESYSTEMIO_OPTIONS is set to SETALL. If not, set it to SETALL. For more details, see Document 396009.1, *Database Initialization Parameters for Oracle E-Business Suite Release 12*, on My Oracle Support.

Check job_queue_processes

1. Log in to Oracle Database 19c.
2. Make sure job_queue_processes set to the recommended value of 1000 at the multitenant container database (CDB) and customer-created pluggable database (PDB) levels. If not, change it.

Set Parameter Values

1. Log in to Oracle Database 19c.
2. Set the following values:
   - CLOB Changes = ALTER SYSTEM SET EVENT='44951 TRACE NAME CONTEXT FOREVER, LEVEL 1024' scope=spfile;

     > **Note:**
     >
     > Back up the earlier events and then append this event (if not already set) in the existing list of events, if any.

   - audit_trial: Check that this parameter uses the same value you used for Oracle Database 11g.

- Execute the following commands:

```
alter session set container="CDB$ROOT"
alter system set "_gc_persistent_read_mostly"=false scope=spfile;
```

- Add SQLNET.ALLOWED_LOGON_VERSION_SERVER=10: and SQLNET.ALLOWED_LOGON_VERSION_CLIENT=10 in SQLNET.ORA of grid home in all RAC Nodes
- Add SQLNET.INBOUND_CONNECTION_TIMEOUT=300 in SQLNET.ORA of grid home and PDB $TNS_ADMIN in all RAC Nodes (source the PDB environment file from ORACLE_HOME)
- Remove the encryption parameter. For details, see *Use of Oracle Database Native Network Encryption with DMW* (Document ID 2783451.1) on My Oracle Support.

The following table contains some more parameters that you must set.

Table 3-1 Parameter Names and their Values

Parameter Name	Parameter Value	Set at Which Container Level (CDB/PDB)?
db_16k_cache_size	16 GB	Only at the CDB level
db_keep_cache_size	12 GB	Only at the CDB level
java_pool_size	At least 2 GB **Note**: Oracle DMW 3.0 and later uses Java database memory with job/install engine architecture.	Only at the CDB level
open_links_per_instance	500	Only at the CDB level
processes	5000	Only at the CDB level
sga_max_size	126 GB (after checking free huge pages)	Only at the CDB level
use_large_pages	ONLY	Only at the CDB level
_enable_NUMA_support	FALSE	Only at the CDB level
_enable_NUMA_optimization	FALSE	Only at the CDB level
_gc_persistent_read_mostly	FALSE	Only at the CDB level
shared_pool_size	26 GB Reset the SHARED_POOL_SIZE to 0 (at the PDB level) by executing the following command from PDB: `alter system reset shared_pool_size scope=both;`	Only at the CDB level
_column_tracking_level	1	At CDB SPFILE
_optimizer_dsdir_usage_control	0	At CDB SPFILE
_optimizer_gather_stats_on_conventional_dml	FALSE	Both at PDB and CDB
_optimizer_use_stats_on_conventional_dml	FALSE	Both at the PDB and CDB levels
_optimizer_gather_stats_on_load	FALSE	Both at the PDB and CDB levels
job_queue_processes	1000	Both at the PDB and CDB levels

Table 3-1 (Cont.) Parameter Names and their Values

Parameter Name	Parameter Value	Set at Which Container Level (CDB/PDB)?
parallel_max_servers	64	Both at the PDB and CDB levels
container_data	CURRENT_DICTIONARY	Both at the PDB and CDB levels
db_cache_size	48 GB	Both at the PDB and CDB levels
open_cursors	1000	Both at the PDB and CDB levels
open_links	50	Both at the PDB and CDB levels
optimizer_features_enable	DEFAULT	Both at the PDB and CDB levels
pga_aggregate_limit	60 GB	Both at the PDB and CDB levels
pga_aggregate_target	30 GB	Both at the PDB and CDB levels
_sql_plan_directive_mgmt_control	0	Both at the PDB and CDB levels
java_jit_enable	TRUE	Both at the PDB and CDB levels

For information on additional parameters and patches if you have installed the JULY 2022 CPU onward, see *DMW Database Parameter Changes And Patches Required Before Applying The July 2022 CPU And October 2022 CPU* (Doc ID 2910163.1) on My Oracle Support.

Create New Indexes

1. Log in to Oracle Database 19c as the SYS user at the container database (CDB) and pluggable database (PDB).
2. Search for the following indexes:
   - i_type3 on type$(roottoid)
   - i_type4 on type$(supertoid)
3. If you cannot locate the indexes in step 2, create them by entering this command:

```
create index i_type3 on type$(roottoid)

/

create index i_type4 on type$(supertoid)

/
```

Set the PL/SQL Service Instances Value

1. Log in to Oracle Database 19c.
2. Set the PL/SQL Service instances value used in the Oracle LSH FORM user interface to 700. None of the service instance configurations should exceed 700.

Gather Statistics

Gather statistics for Oracle LSH and Oracle DMW. For information, see Gathering Schema Statistics for DMW and LSH 3.x and Later Versions (Document ID 2826245.1) on My Oracle Support.

Compile Invalid Objects

Compile all invalid objects at the multitenant container database (CDB) and customer-created pluggable database (PDB) levels.

Apply the Additional EBS Patches

Apply the following additional EBS patches, if not already applied:

> **Note:**
>
> For information on the patch deployment process and how to install the patch, see *How To Apply Patches for LSH/DMW in Downtime Mode* (Document ID 3019034.1) on My Oracle Support.

- `31959538:R12.TXK.C`, `31943873:R12.FWK.C`, and `27222751:R12.FND.C`

Clone the Environment (Optional)

> **Note:**
>
> Make sure you have installed all the pre-required patches and completed the preclone activities before freezing the system for the cloning propose.
> Before cloning the environment, it is recommended to run a full ADOP cycle to make sure the system can switch files systems twice (for example: fs1 > fs2 and fs2 > fs1).

If you plan to create another Oracle LSH environment on the same platform, you can clone your installation at this point. You cannot clone it after you have installed TMS or iAD. Refer to My Oracle Support article 2560690.1, *Cloning Oracle E-Business Suite Release 12.1 with Multitenant Database using Rapid Clone*.

Create Tablespace CDR_BLOB_DATA_TS

> **Note:**
>
> Create the tablespace before installing the 3.4.1 patch.

Create a separate tablespace CDR_BLOB_DATA_TS for the LOB storage. You must calculate an estimated size of the old implementation of the LOB basic file and allocate 25% more tablespace size for the tablespace CDR_BLOB_DATA_TS. This tablespace will store the migrated secure file implementation.

1. Connect to Oracle DMW PDB as SYS or SYSTEM user.

2. Calculate the approximate size of the existing LOB by executing the following SQL statement:

```
WITH lobs AS (
        SELECT /*+ materialize cardinality(2) */
        *
        FROM
        (
        SELECT
        owner,
        segment_name
        FROM
        dba_lobs
        WHERE
        table_name = 'CDR_OUTPUT_BLOBS'
        UNION
        SELECT
        owner,
        index_name segment_name
        FROM
        dba_lobs
        WHERE
        table_name = 'CDR_OUTPUT_BLOBS'
        )
        )
        SELECT
        round(SUM(bytes) / 1024 / 1024) AS size_mb,
        round(SUM(bytes) / 1024 / 1024/1024) AS size_gb,
        round(SUM(bytes) / 1024 / 1024 / 1024 / 1024, 2) AS size_tb
        FROM
        dba_segments
        WHERE
        ( owner,segment_name ) IN (
        SELECT
        *
        FROM
        lobs
        );
```

   The SQL provides the output in MB, GB, and TB.

3. Calculate the tablespace size.

   Size of Tablespace = Estimated Size from step 2 (note the number from SIZE_GB / SIZE_TB) * 1.25 (25% over allocation)

   For example, if the estimated size from step 1 is 100 GB, then Size of Tablespace = 100 * 1.25 = 125 GB.

4. Make sure that there is enough space available in the ASM disk group. In case you do not use the ASM disk group, make sure that there is enough disk space available at the operating system directory where the data files are created.

5. Create the tablespace CDR_BLOB_DATA_TS by executing the following command:

```
CREATE BIGFILE  TABLESPACE CDR_BLOB_DATA_TS DATAFILE
        '<fully qualified datafile name>' SIZE  <Size Calculated in Step 3>
        AUTOEXTEND ON NEXT 25G
```

```
        EXTENT MANAGEMENT LOCAL
        SEGMENT SPACE MANAGEMENT AUTO;
```

6. Assign a quota for the tablespace CDR_BLOB_DATA_TS to the users CDR , APPS, and APPLSYS.

```
ALTER USER CDR QUOTA UNLIMITED ON CDR_BLOB_DATA_TS;

ALTER USER APPS QUOTA UNLIMITED ON CDR_BLOB_DATA_TS;

ALTER USER APPLSYS QUOTA UNLIMITED ON CDR_BLOB_DATA_TS;
```

Ensure Enough Space is Available in Tablespace APPS_TS_MEDIA

Calculate an estimated size of the old implementation of the LOB basic file and allocate 25% more tablespace size for the tablespace APPS_TS_MEDIA.

1. Connect to Oracle DMW PDB as SYS or SYSTEM user.
2. Calculate the approximate size of the existing LOB by executing the following SQL statement:

> **Note:**
>
> Execute the following SQL statement for all of the following tables:
>
> - CDR_INSTALLATION_LOG
> - CDR_OUTPUT_BLOBS
> - CDR_OUTPUT_CLOBS
> - CDR_INSTALL_SCRIPTS
> - DME_DISC_CSV_FILES

```
WITH lobs AS (
        SELECT /*+ materialize cardinality(2) */
        *
        FROM
        (
        SELECT
        owner,
        segment_name
        FROM
        dba_lobs
        WHERE
        table_name = 'enter the table name'
        UNION
        SELECT
        owner,
        index_name segment_name
        FROM
        dba_lobs
        WHERE
```

```
table_name = 'enter the table name'
)
)
SELECT
round(SUM(bytes) / 1024 / 1024) AS size_mb,
round(SUM(bytes) / 1024 / 1024/1024) AS size_gb,
round(SUM(bytes) / 1024 / 1024 / 1024 / 1024, 2) AS size_tb
FROM
dba_segments
WHERE
( owner,segment_name ) IN (
SELECT
*
FROM
lobs
);
```

The SQL provides the output in MB, GB, and TB.

3. Calculate the additional tablespace size required for migration which is equal to the estimated size from step 2 (note the number from SIZE_GB / SIZE_TB) * 1.25 (25% over allocation).

   For example, if the estimated size from step 1 is 100 GB, then additional space required for the tablespace = 100 * 1.25 = 125 GB.

4. Add the datafiles, as required by executing the following command to make sure enough space that was calculated at step 3 is avaiable in the tablespace:

```
ALTER TABLESPACE APPS_TS_MEDIA
            ADD DATAFILE 'fully qualified datafile name'
            SIZE 500M AUTOEXTEND ON NEXT 100M
            MAXSIZE max size limit of the file, typically 30G on Linux OS;
```

   For example, assuming 30 GB of operating system file size limit using ASM storage "+DATA":

```
ALTER TABLESPACE APPS_TS_MEDIA
            ADD DATAFILE '+DATA'
                     SIZE 500M AUTOEXTEND ON NEXT 100M
                     MAXSIZE 30G;
```

5. Assign a quota for the tablespace APPS_TS_MEDIA to the users CDR , APPS, and APPLSYS.

```
ALTER USER CDR QUOTA UNLIMITED ON APPS_TS_MEDIA;

ALTER USER APPS QUOTA UNLIMITED ON APPS_TS_MEDIA;

ALTER USER APPLSYS QUOTA UNLIMITED ON APPS_TS_MEDIA;
```

> **Note:**
>
> Make sure your SYSTEM tablespace has enough space which is managed locally and not managed by dictionary.

4

Installing the Oracle TMS Database Tier

The Oracle Life Sciences Data Hub (Oracle LSH) uses the Oracle Thesaurus Management System (TMS) 5.4.1 database tier internally for its classification system. For more information, see Oracle Thesaurus Management System Installation Guide for release 5.4.1.

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 TMS instance that is installed as part of Oracle LSH. See "Creating Database Accounts" in the *Oracle Life Sciences Data Hub System Administrator's Guide* for further information.

> **Note:**
>
> If you have installed RAC and you have Load Balancing and Failover enabled, the database connection may change from one node to another on the server side. To avoid this problem, shut down all but one database node for the duration of the TMS installation.

5

Installing Oracle Life Sciences Data Hub

$APPL_TOP, $CDR_TOP, and $JAVA_TOP are all on the middle tier. The Distributed Processing (DP) Server Home directory is located on the DP Server. These may all be physically located on the same computer or they may be located on different computers, depending on your installation.

These instructions include copying files from one of these directories to another. Remember that you need to use FTP if they are on different computers.

> **Note:**
>
> During its initial development, the Oracle Life Sciences Data Hub (Oracle LSH) was known 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 as you go through the installation process.

This section includes the following topics:

- Create Auditing Process
- Create CDR_USER Role
- Regenerate the JAR Files
- Set Up Resource Capping
- Turn off EXE DEBUG
- Set the Profile Value for DMW_SEND_BATCHSIZE
- Set Up the Notification Mailer
- Set Up the Distributed Processing Server
- Set Up Client Computers
- Schedule Jobs
  Schedule the various jobs.
- Run the Health Check Scripts

Provide Required Grants to the APPS Schema

- Connect to the PDB as SYS user and execute the following commands:

```
grant execute any procedure to apps with admin option;


grant select any dictionary to apps;


grant administer database trigger to apps;


grant create table, create procedure to apps with admin option;


grant create tablespace to apps;
```

Apply the Oracle Life Sciences Data Hub AD Splicer Patch

Because Oracle LSH is off-cycle from the rest of Oracle Applications, you must use the Applications DBA AD Splicer. You must apply the AD Splicer patch for Oracle LSH, patch number **6114439**, which is on the Oracle LSH media pack, before you apply the Oracle LSH patch.

Follow instructions in the readme file on how to splice Oracle LSH into your Oracle Applications instance **using the AD Splicer**.

> **Note:**
>
> The readme file currently contains contradictory information about how to apply the patch, saying first to use the AD Splicer and not Autopatch, and later saying to apply the driver using Autopatch. **Use only the AD Splicer**. Do not apply the unified driver using Autopatch as it says at the end.

For information on the AD Splicer utility, see the section on the AD Splicer in *Creating a Custom Application in Oracle E-Business Suite Release 12.1.3 and above* (Doc ID 1577707.1).

Ensure to synchronize the run and patch file systems. To do so, execute the following command:

```
adop phase=fs_clone
```

For more information, see File System Synchronization Options.

Create Tablespace CDR_BLOB_DATA_TS

> **Note:**
>
> Create the tablespace before installing the 3.4.1 patch.

Create a separate tablespace CDR_BLOB_DATA_TS for the LOB storage. You must calculate an estimated size of the old implementation of the LOB basic file and allocate 25% more tablespace size for the tablespace CDR_BLOB_DATA_TS. This tablespace will store the migrated secure file implementation.

1. Connect to Oracle DMW PDB as SYS or SYSTEM user.
2. Calculate the approximate size of the existing LOB by executing the following SQL statement:

```
WITH lobs AS (
        SELECT /*+ materialize cardinality(2) */
        *
        FROM
        (
        SELECT
        owner,
        segment_name
        FROM
        dba_lobs
        WHERE
        table_name = 'CDR_OUTPUT_BLOBS'
        UNION
        SELECT
        owner,
        index_name segment_name
        FROM
        dba_lobs
        WHERE
        table_name = 'CDR_OUTPUT_BLOBS'
        )
        )
        SELECT
        round(SUM(bytes) / 1024 / 1024) AS size_mb,
        round(SUM(bytes) / 1024 / 1024/1024) AS size_gb,
        round(SUM(bytes) / 1024 / 1024 / 1024 / 1024, 2) AS size_tb
        FROM
        dba_segments
        WHERE
```

```
( owner,segment_name ) IN (
SELECT
*
FROM
lobs
);
```

The SQL provides the output in MB, GB, and TB.

3. Calculate the tablespace size.

   Size of Tablespace = Estimated Size from step 2 (note the number from SIZE_GB / SIZE_TB) * 1.25 (25% over allocation)

   For example, if the estimated size from step 1 is 100 GB, then Size of Tablespace = 100 * 1.25 = 125 GB.

4. Make sure that there is enough space available in the ASM disk group. In case you do not use the ASM disk group, make sure that there is enough disk space available at the operating system directory where the data files are created.

5. Create the tablespace CDR_BLOB_DATA_TS by executing the following command:

```
CREATE BIGFILE  TABLESPACE CDR_BLOB_DATA_TS DATAFILE
        '<fully qualified datafile name>' SIZE  <Size Calculated in Step 3>
        AUTOEXTEND ON NEXT 25G
        EXTENT MANAGEMENT LOCAL
        SEGMENT SPACE MANAGEMENT AUTO;
```

6. Assign a quota for the tablespace CDR_BLOB_DATA_TS to the users CDR , APPS, and APPLSYS.

```
ALTER USER CDR QUOTA UNLIMITED ON CDR_BLOB_DATA_TS;

ALTER USER APPS QUOTA UNLIMITED ON CDR_BLOB_DATA_TS;

ALTER USER APPLSYS QUOTA UNLIMITED ON CDR_BLOB_DATA_TS;
```

Install Oracle LSH 3.4.1

You must install Oracle LSH 3.4.1 as a patch to Oracle Applications. The patch **36272413** is on the media pack.

> **Note:**
>
> The information on how to install the patch and set up the cleanup job are covered under document IDs 3019034.1 and 2925664.1 respectively on My Oracle Support. Contact Life Sciences Support to get these documents.

1. Locate p36272413_R12_GENERIC.zip in the staging area.
2. Unzip p36272413_R12_GENERIC.zip to $NE_BASE/EBSapps/patch.
3. Apply the patch. For information on how to apply the patch, see *How To Apply Patches for LSH/DMW in Downtime Mode* (Document ID 3019034.1) on My Oracle Support.
4. As an APPS user, run the following script:

> **Note:**
>
> While executing, pass the value as APPLSYS when it asks for parameter: 1.

```
$CDR_TOP/patch/115/sql/dme341ddl.sql
```

5. As a PDB sys user, recompile the APPS invalid object:
   ```
   EXEC UTL_RECOMP.recomp_parallel(16,'APPS');
   ```
6. Set up the cleanup job. For information on how to set up the cleanup job, see document ID 2925664.1 My Oracle Support.
7. Log in to the database as APPS and execute the following script:
   ```
   $CDR_TOP/patch/115/sql/dmeErrLogChanges.txt
   ```

   The script alters the name of the internal table “dme_err_log” and creates a public view on the table. The script creates a trigger on the view to prevent the unauthorized users from inserting or updating records from the script.

Install the CdrRuntime.jar File

1. On the Oracle LSH server, navigate to the $CDR_TOP/jar folder. It contains the CdrRuntime.zip file.
2. Copy the CdrRuntime.zip file to the Oracle database server in any temporary location. Then enter this command to unzip the file and extract CdrRuntime.jar:

   ```
   unzip CdrRuntime.zip
   ```
3. Run the following commands as an Oracle home owner on the Oracle database server to load the java files to the database.

   First, execute the following command:

   ```
   dropjava -force -thin -user apps/<password>
   @(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<DB hostname>)(PORT=<DB port
   number>))(CONNECT_DATA=(SERVICE_NAME=<DB service name>))) CdrRuntime.jar
   ```

   Then, execute this command:

   ```
   loadjava -force -thin -user apps/
   <password>@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<DB host name>)(PORT=<DB
   port number>))(CONNECT_DATA=(SERVICE_NAME=<DB service name>))) CdrRuntime.jar
   ```
4. Run this query to return a set of alter java commands that should be run to compile invalid classes:

   ```
   SELECT

   'alter java class "'

   || object_name

   || '" compile;'

   FROM

   dba_objects

   WHERE

   object_type = 'JAVA CLASS'

   and object_name LIKE '%cdr%'
   ```

```
AND status = 'INVALID';
```

5. Connect to PDB as APPS user and run the statements returned by the above query.
6. Rerun the query mentioned in step 4 and confirm that it does not return any rows.

> **Note:**
>
> If the query returns rows, execute the statements returned again. Repeat this process until the SQL does not return any rows.

Install Online Help

For full installation of Oracle LSH (not an upgrade), install Online Help patch **18551089** (available on the media pack) using ADOP to access the online help topics from HTML versions of the following Oracle LSH guides. For information on how to apply the patch in the downtime mode, see *How To Apply Patches for LSH/DMW in Downtime Mode* (Doc ID 3019034.1).

- *Oracle Life Sciences Data Hub System Administrator's Guide*
- *Oracle Life Sciences Data Hub Implementation Guide*
- *Oracle Life Sciences Data Hub Application Developer's Guide*
- *Oracle Life Sciences Data Hub User's Guide*

Check Non-Default Service

1. Log in to Oracle Database 19c as a sysadmin user.
2. Execute this query to find all the service names with LONG as the service type:

```
select NAME, NETWORK_NAME, CON_NAME, CLB_GOAL
from gv$active_services
where CLB_GOAL='LONG';
```

3. In the list of service names with LONG as the service type, ensure that the non-default service name is not same as the PDB, CDB, or any SYS service name.

Set CDR_DB_SERVICE_NAME

1. Log in to Oracle E-Business Suite as sysadmin user.
2. Expand the **System Administrator Responsibility** folder in the left panel.
3. Expand **Profile** and then select **System**.
4. Open the forms and search for the profile name "CDR_DB_SERVICE_NAME".
5. At the site Level, enter: `ebs_PDB name` identified in the Check Non-Default Service section.
6. Save the form.

Set CDR_SCAN_HOST_NAME

Set CDR_SCAN_HOST_NAME to the database scan name used in the Oracle LSH FORM user interface. You can get the database scan name by executing the following command as a SYS user in CDB: `SQL> show parameter remote_listener`.

You only need to perform the following steps if you use a RAC configuration.

1. Log in to Oracle E-Business Suite as sysadmin user.
2. Expand the **System Administrator Responsibility** folder in the left panel.
3. Expand **Profile** and then select **System**.
4. Open the forms and search for the profile name "CDR_SCAN_HOST_NAME".
5. At the site Level, enter: `SCAN NAME`
6. Save the form.

Set CDR_SCAN_PORT_NUM

Set CDR_SCAN_PORT_NUM to the database scan listener port number used in the Oracle LSH FORM user interface. You can get the database scan name by executing the following command as a SYS user in CDB: `SQL> show parameter remote_listener`.

You only need to perform these steps if you use a RAC configuration.

1. Log in to Oracle E-Business Suite as sysadmin user.
2. Expand the **System Administrator Responsibility** folder in the left panel.
3. Expand **Profile** and then select **System**.
4. Open the forms and search for the profile name "CDR_SCAN_PORT_NUM".
5. At the site Level, enter: `SCAN Port Number`
6. Save the form.

Add clovi.gif to the EBS R12.2 File System

1. Source the PATCH file system environment file from the Oracle LSH application server by using the following command:

   `source /<BASE_LOCATION>/EBSapps.env PATCH`

2. Perform the following:

   > **Note:**
   >
   > Ensure that the destination path where you copy the clovi.gif is correct.

   a. Copy clovi.gif from `$CDR_TOP/media` to `$RUN_BASE/FMW_Home/Oracle_EBS-app1/applications/oacore/html/cabo/images/cache`.

   b. Copy clovi.gif from `$CDR_TOP/media` to `$PATCH_BASE/FMW_Home/Oracle_EBS-app1/applications/oacore/html/cabo/images/cache`.

3. Re-source the environment file, as required.

Create Oracle LSH UI JAR File

1. Create a directory *$FND_TOP*/admin/template/custom and grant the "755" permission.
2. Copy the existing *$FND_TOP*/admin/template/ebsProductManifest_xml.tmp to *$FND_TOP*/admin/template/custom.
3. Modify *$FND_TOP*/admin/template/custom/ebsProductManifest_xml.tmp to add the following additional <classpath> entry after the <classpath> section for the customall.jar file. This should be after </classpath> and before the line </classpath-contents>.

```
<classpath>
      <absolute-path>%s_current_base%/EBSapps/comn/java/classes/oracle/
apps/cdr/jar</absolute-path>
      <relative-path>../../../../java/classes/oracle/apps/cdr/jar</
relative-path>
      <windows-path>@BEA_HOME@/../EBSapps/comn/java/classes/oracle/
apps/cdr/jar</windows-path>
      <libraries>
        <library>cdrlshall.jar</library>
     </libraries>
    </classpath>
```

   Replace *EBSapps* with your corresponding directory name if it is different.
4. Run AutoConfig.
5. Bounce the application tier services.

> **Note:**
>
> These changes must be redone if ebsProductManifest_xml.tmp is patched in future.

6. In order to to synchronize the changes (during the next prepare phase) between both the file systems fs1 and fs2, back up the file $APPL_TOP_NE/ad/custom/adop_sync.drv.
7. Add the following lines (after replacing EBSapps with your correct directory name) to the original "adop_sync.drv" file just before the line "#End Customization".

```
# For LSH 3.x : BEGIN
cp -R %s_current_base%/EBSapps/comn/java/classes/oracle/apps/cdr
%s_other_base%/EBSapps/comn/java/classes/oracle/apps
cp %s_current_base%/EBSapps/comn/java/classes/oracle/apps/cdr/jar/
cdrlshall.jar %s_other_base%/EBSapps/comn/java/classes/oracle/apps/cdr/jar
cp %s_current_base%/EBSapps/appl/fnd/12.0.0/admin/template/custom/
ebsProductManifest_xml.tmp %s_other_base%/EBSapps/appl/fnd/12.0.0/admin/
template/custom
# For LSH 3.x : END
```

8. After the changes are synchronized, make sure you run Autoconfig for the latest template changes to take effect.

Set Profile at the Site-Level

Set the Manual Password Reset Enabled profile to Yes at the site-level.

1. Log on to Oracle E-Business Suite as sysadmin user.
2. Expand the **System Administrator** folder in the left panel.
3. Expand **Profile**, and select System.

   The Find System Profile Values window appears.
4. In the **Profile** field, enter `Manual Password Reset Enabled`.
5. Click **Find**.

   The system returns you to the System Profile Values window with Manual Password Reset Enabled displayed in the Profile Option Name column.
6. Select `Yes` as the value at the site-level.
7. Click **Save**.
8. Repeat steps 3 through 7 to set the values for the following profiles:
   - `LSH: Use Character Semantics for Workarea Installation` - Select Yes
   - `FND: NATIVE CLIENT ENCODING` - Select UTF8
   - `LSH: Gather Work Area Schema Statistics` - Select Yes
   - `LSH: Domain Nesting Levels` - Set the value as 9
   - `RRA: Enabled` - Select Yes

Create System Administrator and Security Administrator Users

For full installations of Oracle LSH (not an upgrade), you must create Oracle LSH user accounts for one or more users and give them the roles required to perform Oracle LSH setup tasks:

- The **LSH System Administrator** can run the post-installation job and define service locations and services in the LSH user interface. (See Run the Post-Installation Programs and Define Service Locations and Services for details.)
- The **LSH Security Administrator** assigns other special security administrator roles (Adapter Security Administrator, Classification Administrator, Data Blind Administrator, Bootstrap Administrator, User Group Administrator) to other users and sets up the object security system (defines object subtypes, roles, and user groups).

The *Oracle Life Sciences Data Hub System Administrator's Guide* includes instructions for these and other security-related tasks. The *Oracle Life Sciences Data Hub System Implementation Guide* includes information on designing an object security system for your organization.

- Log on as sysadmin
- Create a User Account for the Administrator
- Assign a Role

Log on as sysadmin

To log on, do the following:

1. Open your web browser.
2. Enter the eBusiness Suite SSWA (Self-Service Web Application) URL as follows:

   http://<host name>.<domain name>:<HTTP port>/OA_HTML/AppsLogin

   For example:

   http://*appshost*.*your_company*.com:8000/OA_HTML/AppsLogin

   The Applications Login screen appears.
3. Log in as sysadmin.

   An E-Business Suite screen opens.

Create a User Account for the Administrator

You must create a user account for each person to whom you want to assign the LSH Security Administrator or LSH System Administrator role. Oracle LSH uses the standard Oracle Applications UMX interface for creating user accounts.

> **Note:**
>
> For complete information, see the *Oracle Applications System Administrator's Guide —Security*, Oracle Part Number B13923-02. The book is included on the media pack.
>
> You can see it online from the Oracle Technology Network at `http://www.oracle.com/technology/index.html`.

1. Select **User Management** by clicking on it in the Navigator (near the bottom of the list on the left). New links appear in the second column, including **Users**.
2. Click **Users** under User Management. The User Maintenance screen appears.
3. From the **Register** drop-down list, select **External Organization Contact** and click **Go**. The Register Business Contact screen appears.
4. Enter values in the following fields:
   - **Email Address**. Enter the user's email address. Oracle LSH uses this address for corresponding with the user.
   - **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.
   - **Organization**. If you have a multi-organizational setup, enter or search for the Organization the user belongs to.
   - **Phone Number**. The telephone contact details for the user.
   - **Extension**. The extension of the provided telephone number of the user.
   - **Account Information**. Select **Generate Automatically** for Oracle LSH to generate and email the password. Or select **Enter Manually** and type and confirm the password.

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

Assign a Role

Assign administrator roles as follows:

- **LSH System Admin**. You must assign the LSH System Admin role to at least one user.
- **LSH Setup Admin**. You must assign the LSH Setup Admin responsibility to at least one user.
- **LSH Adapter Security Admin**. You must assign the LSH Adapter Security Admin role to at least one user.
- **LSH Data Security Admin**. This role allows a user to create all the objects required by the Oracle LSH object security system: object subtypes, object roles, and user groups.
- **LSH Function Security Admin**. This role allows a user to create user accounts and assign functional roles to them. Functional roles control which parts of the Oracle LSH user interface a user can view or allow special privileges.
- **LSH Security Administrator**. This role is a combination of the LSH Data Security and Functional Security Admin roles.

> **Note:**
>
> For further information on Oracle LSH security and security roles, see the *Oracle Life Sciences Data Hub System Administrator's Guide*.

To assign a functional role to a user:

1. Go to the User Maintenance screen.

   If you are already in the User Management tab, click the Users subtab.

   Or log in as sysadmin, click the **User Management** responsibility in the navigator, click **Users**, and click the Users subtab.
2. Search for the user to whom you want to assign roles. The system displays the search results in the lower portion of the screen.
3. Click the Update icon corresponding to the user. The Update User screen appears.
4. Click **Assign Roles**. The Search and Select screen appears.
5. Search for all Oracle LSH predefined roles by selecting Search By **Roles and Responsibilities**, entering `LSH%`, and clicking **Go**. The system displays all the predefined Oracle LSH functional roles in the lower part of the screen.
6. Select each role you want to assign by checking its box. See Assign a Role for further information.
7. Click **Select**. The system displays additional fields. Enter values as follows:
   - **Justification**. You must enter text in this field. Describe the reason this person needs this role.

- **Active From**. The system automatically enters the current date. If you prefer to have the user's privileges begin at a later date, you can select the date you want by clicking the calendar icon.
- **Active To**. Leave this field blank to avoid having the user's privileges automatically expire on the date you specify. When the user leaves the company or changes roles, you can enter an expiration date here. If you want to set an end date for the user's privileges associated with this role, use the calendar icon to specify the end date.

8. Click **Apply**. The system assigns the role(s) you specified plus any necessary base roles to the user.

Run the Post-Installation Programs

You must run the Oracle LSH LOB Loader and post-installation concurrent programs.

> **Note:**
>
> If you are using RAC, shut down all but one database node before running the post-installation programs. If you leave more than one node up, the jobs may run successfully but you may get errors.

- Log On to Oracle Applications
- Load the Adapter Files
- Set the Tech Type Value
- Run the Post-Installation Concurrent Program
- Perform Additional Database Task
- Configure Automated Workarea Tablespace Creation
  Configuration required to enable the automated workarea tablespace creation.

Log On to Oracle Applications

To run the jobs, you must log on to Oracle Applications as an Oracle LSH user with the following roles:

- LSH Setup Admin
- LSH Adapter Security Admin

1. Open your Web browser.
2. Enter the eBusiness Suite SSWA (Self-Service Web Application) URL as follows:

   http://<host name>.<domain name>:<HTTP port>/oa_servlets/AppsLogin

   For example:

   http://*appshost*.*your_company*.com:8000/oa_servlets/AppsLogin

   The Applications Login screen appears.
3. Log in as a user with LSH Setup Admin and LSH Adapter Security Admin privileges. The Oracle Applications Home page appears.
4. Select the **LSH Setup Admin** Responsibility by clicking on it. **Lookups** appears in the second column.

5. Click **Lookups**. (If necessary, click **Grant This Session** or **Grant Always**.) The Oracle Life Sciences Data Hub Lookups window opens.
6. From the **View** menu, select **Requests**.

Load the Adapter Files

Oracle LSH includes predefined adapters that control the interaction between Oracle LSH and other systems. The Oracle LSH LOB Loader concurrent program finds all the adapter SQL files and loads them into a table in Oracle LSH.

To run the Oracle LSH LOB Loader:

1. Follow steps in Log On to Oracle Applications.
2. Click **Submit a New Request**. The Submit a New Request window opens.
3. Select **Single Request** and click **OK**.
4. Click the gray LOV button on the right of the **Name** field. The Reports List of Values opens.
5. Select **LSH LOB Loader Concurrent Program** and click **OK**.
6. Click **Submit**.
7. Click **OK**. A window pops up with the job ID and asks if you want to submit another request.
8. Click **No**.

To monitor the concurrent program's progress:

1. Click **Find**.
2. Click **Refresh Data** periodically to update the execution phase and status displayed on screen.

When the status is Complete you can view the log file by clicking the **View Log** button.

> **Note:**
>
> - Always check the log file because the phase may be Complete and the status Normal, but the program may not have successfully completed all its tasks.
> - If you see this message in the log file: "ORA-20001: APP-FND-02901: You do not have access privilege to any operating unit. Please check if your profile option MO: Security Profile includes any operating unit or the profile option MO: Operating Unit is set. has been detected in MO_GLOBAL.INIT," see *ORA-20001: APP-FND-02901 Errors Running Collections or Plan in 12.2 OR in Trying to View Request Log from Legacy Collections Self Service In R12.0/12.1* (Document ID 981828.1) on My Oracle Support.
> - If you see this message in the log file: "ERROR: LOBLoaderCP.runProgram() Exception String index out of range: -1", see *Problems Encountered During the Installation of LSH 2.2, Upgrade to LSH 2.2, and Execution of its Verification Tests* (Document ID 1327829.1) on My Oracle Support.

Set the Tech Type Value

In a fresh installation of Oracle LSH, you need to set the Tech Type value as described below. These steps are not required in an upgrade.

1. Connect to PDB as **apps** user.
2. Run the following SQL statement:

```
select CDR_TECH_TYPES_ID_SEQ.nextval from dual;
```

Run the Post-Installation Concurrent Program

Run the Job: To run the Oracle LSH post-installation concurrent process:

1. Click **Submit a New Request** under one of the following circumstances:
   - in the Requests window you used to monitor the Oracle LSH LOB Loader concurrent process
   - after following the steps in Log On to Oracle Applications
2. Select **Single Request** and click **OK**. The Submit Request window opens.
3. Click the gray LOV button on the right of the **Name** field. The Reports List of Values opens.
4. Select **LSH: Post Installation Program** and click **OK**. The Parameters pop-up window appears.
5. Enter values for the following Parameters:

> **Note:**
>
> Carefully set the following parameters. You cannot change the values for some of the parameters after you run the job.

   - **Company ID**. The company ID serves as part of the primary key for all the Oracle LSH objects you define in this instance of Oracle LSH.
   - **Owning Location**. Enter the name of your Oracle Applications instance.
   - **Object Sequence Start Value**. Enter a single digit. The system will end all object IDs with this digit to further distinguish objects created in this Oracle LSH instance.
   - **Object Sequence Start Value**. Leave blank. The system will end all object IDs with the number 1.
   - **Database Host Name**. Enter the machine name of the database server instance.
   - **Database Port Number.** Enter the port number of the database server instance.
6. Click **Submit**. A window pops up with the job ID and asks if you want to submit another request.
7. Note the job ID and click **No**.

Monitor the Process: To monitor the concurrent process's progress:

1. Click **Find**. Use the job ID to search for the process.

2. Click **Refresh Data** periodically to update the execution phase and status displayed on screen.
3. When the phase is Complete, click the **View Log**.

> **Note:**
>
> Always check the log file, because the phase may be Complete and the status Normal and yet the process may not have successfully completed all its tasks. In such case, contact Life Sciences Support.

Check the log file to make sure it did the following:

- Set the company ID
- Set the owning location
- Recreated the cdr_object_id_seq with the start value you provided
- Inserted one record each in the cdr_namings and cdr_naming_versions tables for the instance domain

> **Note:**
>
> The job does the above only the first time it runs.

- Set the profile to check if the post-installation has been run for this site.

- Change a Post-Installation Job Parameter Value

Change a Post-Installation Job Parameter Value

To create an adapter, the post-installation job creates objects—including a Work Area and Program—inside an Adapter Area in an Adapter Domain, and installs the Work Area. These objects are created using the post-installation job parameters values that you entered when you ran the job.

> **Note:**
>
> If you change the values of some of these parameters after running the job, it may cause problems. The problems vary depending on which parameter you change, and are given for each parameter listed in this topic.

However, you can call the function FND_PROFILE.SAVE in SQL*Plus to change most parameters. This function takes the following parameters. You must enter single quotes around each value.

- **x_name** is the profile name
- **x_value** is the profile value that you want to set
- **x_level_name** is the name of the level at which the value should be set

> **Note:**
>
> You must set each value at the Site level. To do this, enter `'Site'` for the function parameter x_level_name each time you call the function.

To call FND_PROFILE.SAVE:

1. Log in to SQL*Plus as apps.
2. At the SQL prompt, enter:

```
begin
FND_PROFILE.SAVE ('x_name''x_value' 'x_level_name');
end;
/
```

You need a line calling FND_PROFILE.SAVE for each post-installation job parameter you need to change. The function parameter values required to change each post-installation parameter are included below. When you are ready to commit the changes, enter:

```
commit;
```

The details for each post-installation job parameter are:

- **Object ID Sequence**. You cannot change this value.
- **DB Host Name**. If the Oracle LSH database moves to a different machine or the name of the current machine changes, you can call FND_PROFILE.SAVE to change the name. However, changing this value will result in subsequent adapter Work Area installations and upgrades failing when you upgrade to a new version of Oracle LSH. Provide the following function parameter values:

  ```
  FND_PROFILE.SAVE('CDR_DBHOST_NAME','New_Host_Name', 'SITE');
  ```

- **DB Port Number**. If the DB port number changes, you can call FND_PROFILE.SAVE to change it. However, changing this value will result in subsequent adapter Work Area installations and upgrades failing when you upgrade to a new version of Oracle LSH. Provide the following function parameter values:

  ```
  FND_PROFILE.SAVE('CDR_DBPORT_NUM','New_Port_Number', 'SITE');
  ```

- **Owning Location**. There are currently no restrictions in changing this parameter value. Provide the following function parameter values:

  ```
  FND_PROFILE.SAVE('OWNING_LOCATION','New_Owning_Location', 'SITE');
  ```

- **Company ID**. Never change the company ID. The company ID is part of the unique key for every object in Oracle LSH, including the adapter objects that were created by running the post-installation job. Changing this value could lead to the system not working in many places, including job execution, the user interface not listing submitted jobs, and more.

Perform Additional Database Task

1. Log in as apps account.
2. Execute the following script:

```
SET serveroutput ON
declare
  status1 boolean;
```

```
begin
  status1 := cdr_profiles_pkg.SAVE
('DMW:Set Based Processing Enabled','$YESNO$NO','SITE');
  if status1 then
   dbms_output.put_line('Success');
  else
   dbms_output.put_line('Failure');
  end if;
end;
/
commit;
```

3. Make sure that executing the following query returns the result as $YESNO$NO:

   ```
   select cdr_profiles_pkg.value('DMW:Set Based Processing Enabled') from dual;
   ```

   If the result of the query is not $YESNO$NO, contact Life Sciences Support.

Configure Automated Workarea Tablespace Creation

Configuration required to enable the automated workarea tablespace creation.

1. Verify that all the following objects are created. To do so, execute the following:

   ```
   select OWNER, OBJECT_NAME, OBJECT_TYPE, STATUS, CREATED, LAST_DDL_TIME
   from dba_objects
   WHERE OBJECT_NAME IN
   ('CDR_WA_TBLSPC_NM_CNTR','CDR_WA_METADATA','CDR_WA_TLSPC_AUDT','CDR_WA_META
   DATA_U1', 'CDR_CREATE_WA_TABLESPACE');
   ```

2. Configure the metadata in the metadata table CDR_WA_METADATA. To do so, log in to the database as APPS and execute the following script:

   ```
   ($CDR_TOP/patch/115/sql/cdrwatblspcinsmtdt.sql
   ```

   The script prompts you for the different inputs as shown below.

   ```
   **********  DO YOU WANT TO ENABLE AUTO TABLESPACE CREATION WHEN THERE IS SPACE ISSUE
   (YES / NO) ***********
   AUTO TABLESPACE CREATION: (YES/NO) <ENTER REQUIRED VALUE>
   **********  ENTER DATAFILE DIRECTORY WHERE DATAFILES WILL BE CREATED **********
   DATAFILE DIRECTORY <ENTER COMPLETE DATAFILE PATH WHERE NEW TABLESPACES WILL BE
   CREATED >
   **********  ENABLE SENDING EMAIL ALERT (YES / NO ) ***********
   SEND ALERT EMAIL (YES / NO) <ENTER REQUIRED VALUE>
   ```

   After the script is executed successfully, verify the content of the metadata table CDR_WA_METADATA.

3. To enable the automated workarea tablespace creation, perform the following steps:

   a. As APPS user, execute the following command:

      ```
      exec Cdr_Create_WA_Tablespace.P_WA_BIGFILE_TBLSPC(<NUMBER OF
      TABLESPACES>);
      ```

Execute the following SQL to verify if the new tablespace has been created successfully:

```
select *
                from dba_tablespaces
                where tablespace_name like 'CDR_WA_OBJ_BFL_TS_%';
```

Execute the following SQL to verify if the new tablespace-related entry is present at metadata table CDR_WA_TBLSPC_NM_CNTR:

```
select *
from  CDR_WA_TBLSPC_NM_CNTR;
```

b. Schedule the weekly monitor job. Execute the following command:

> **Note:**
>
> You can set the `repeat_interval` as per the requirement.

```
BEGIN
DBMS_SCHEDULER.create_job (
job_name => 'WA_BFL_TBLSPC_GRP_MONITOR',
job_type => 'PLSQL_BLOCK',
job_action => 'begin Cdr_Create_WA_Tablespace.P_WA_TBLSPC_FREESPACE();
end;',
start_date => SYSTIMESTAMP,
repeat_interval => 'FREQ=WEEKLY; BYDAY=SAT;',
enabled => TRUE);
END;
/
```

- (Optional) <Enter one of the user's choices while performing this step.>
- (Optional) <Enter another of the user's choices while performing this step.>

4. <Enter the next step.>
   a. (Optional) <Enter a substep.>
   b. (Optional) <Enter a substep.>

Gather Statistics on Schemas

For details on gathering statistics on schemas (CDR and APPS), see Gathering Schema Statistics for DMW and LSH 3.x and Later Versions (Document ID 2826245.1) on My Oracle Support.

Schedule the Context Index Refresh Program

The Oracle LSH Context Index Refresh Program refreshes context server indexes in Oracle LSH and TMS. You must set it up to run regularly so that user-entered metadata is continuously updated and available for use in Oracle LSH.

To schedule the Context Index Refresh Program:

1. Click the **Submit a New Request** button either:
   - in the Requests window you used to monitor the Oracle LSH Post-Installation Program
   - after following steps under Log On to Oracle Applications
2. Select **Single Request** and click **OK**.
3. Click the gray LOV button on the right of the **Name** field. The Reports List of Values opens.
4. Select **LSH Context Index Refresh Program** and click **OK**.
5. Click **Schedule**. The Schedule pop-up opens.
6. In the Run the Job section, select **Periodically**. Additional fields appear.
7. Schedule the job to run every two minutes by typing the number **2** in the blank field and selecting **Minutes** from the drop-down list in the Run Every line.
8. Click **OK**. A warning message appears stating that selecting a schedule without an end date will result in the request's being resubmitted until cancelled.
9. Click **OK**. The Submit Request screen appears.
10. Click **Submit**. A window pops up with the job ID and asks if you want to submit another request.
11. Click **No**.
12. To monitor the concurrent program's progress:
    a. Click **Find**.
    b. Click **Refresh Data** periodically to update the execution phase and the status displayed on screen.
    c. When the status is Complete, click **View Log** to view the log file. The log file should not have any errors.

Start Journaling Internal Tables

Oracle keeps an audit trail of all data changes in some of its internal metadata tables in shadow journaling tables. However, you must explicitly turn this feature on. Journaling tables help to satisfy regulatory requirements.

- Set AuditTrail:Activate Profile to Yes
- Define the Audit Installation

Set AuditTrail:Activate Profile to Yes

Set the **AuditTrail:Activate** profile to Yes at the Oracle LSH Application level.

1. Log on as sysadmin; see Log on as sysadmin for details.
2. Click the **System Administrator** responsibility link. A new column of links appears.
3. Under **Profile** in the right-hand column, click **System**.

   If you receive a message asking if you want to install an applet, do so.

   The **Find System Profile Values** window appears.
4. In the Display section, uncheck **Site** and check **Application**.
5. In the **Application** field, enter`%life%` to bring up Oracle Life Sciences Data Hub.

6. In the **Profile** field, enter `AuditTrail:Activate` in the **Profile** text box at the bottom of the window.
7. Click **Find**. The system returns you to the **System Profile Values** window with `AuditTrail:Activate` displayed in the **Profile Option Name** column and `Oracle Life Sciences Data Hub` displayed in the **Application** column.
8. Select `Yes` as the value for the column **Application** and click the **Save** icon or save from the File menu (File > Save).

Define the Audit Installation

To start journaling, do the following:

1. Log on as sysadmin; see Log on as sysadmin for details.
2. Click the **System Administrator** responsibility link. A new column of links appears.
3. In the new column of links, scroll down to **Security: Audit Trail**. In this section, click **Install**.

   If you receive a message asking if you want to install an applet, do so.

   The **Audit Installations** window appears.
4. Click the Search icon (flashlight/torch) in the toolbar. The **Find Audit Installations** pop-up appears.

> **Note:**
>
> If the icon is inactive, the wrong window is in focus. If necessary, retrieve the **Audit Installations** window:
>
> a. In the **Navigator - System Administrator** window, **Functions** tab, expand the **Security** node.
> b. Expand the **Audit Trail** node.
> c. Click **Install**.The **Audit Installations** window appears.
>
> Then click the Search icon in the toolbar.

5. Click in the **Oracle Username** field to display the ellipsis (...), then click the ellipsis to display the list of values. Find and select `CDR`.
6. Click **Find**. The system returns you to the **Audit Installation** window with `CDR` displayed in the **Oracle Username** column
7. Select the **Audit Enabled** checkbox and click the save icon.

Increase Memory Available for MetaData Reports

To allow predefined Oracle LSH metadata reports to run, you must change the Options setting for each of the concurrent programs that runs a set of reports. In addition, you may want to change the Priorities setting to the highest possible setting for the quickest display of the reports. The short names of the concurrent programs, with the type of reports they run, are:

- CDR_MDATA_DEFN_CP (Definitions reports)
- CDR_MDATA_INST_CP (Instances reports)

- CDR_MDATA_LIBR_CP (Library reports)
- CDR_MDATA_SECU_CP (Security reports)
- CDR_MDATA_WA_CP (Work Area reports)

See the *Oracle Life Sciences Data Hub System Administrator's Guide* chapter on System Reports for information on the reports.

To change the setting, do the following for each set of reports:

1. Log on as sysadmin; see Log on as sysadmin for details.
2. Click the **System Administrator** responsibility link. A new column of links appears.
3. In the new column of links, scroll down to **Concurrent: Program**. In this section, click **Define**.

   If you receive a pop-message asking if you want to install an applet, do so.

   The **Concurrent Programs** window appears.
4. Press the F11 key to enter Query mode.
5. In the **Short Name** field, enter the short name of one of the sets of metadata reports (see list above). For example, enter: `CDR_MDATA_DEFN_CP`
6. Press Ctrl+F11 to enter the query. The system populates all the fields with the current information for that set of reports.
7. In the **Options** field, enter: `-Xmx512M`
8. (Optional) In the **Priority** field, enter: `1`

   1 is the highest possible setting and 100 is the lowest possible setting.
9. From the **File** menu, select **Save**.
10. Repeat the procedure until you have changed the settings for all metadata report sets.

Grant Security Rights to Seeded Adapters

In order to ensure that seeded adapters have the security rights they need to call APIs, do the following:

1. Log in to the application server.
2. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

   ```
   source /<BASE_LOCATION>/EBSapps.env RUN
   ```
3. Connect to PDB as apps user.
4. Run $CDR_TOP/patch/115/sql/cdradaptergrants.sql

Create Auditing Process

> **Note:**
>
> You can execute the cdraudinstall.sql script only once in the environment.

1. Log in to the application server.
2. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

   ```
   source /<BASE_LOCATION>/EBSapps.env RUN
   ```

3. Connect to PDB as apps user.
4. Run $CDR_TOP/patch/115/sql/cdr34audpostprocess.sql.
5. Run $CDR_TOP/patch/115/sql/cdrauditttabpkreg.sql.

Create CDR_USER Role

Create the CDR_USER role using the following script:

1. Log in to the application server.
2. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

   ```
   source /<BASE_LOCATION>/EBSapps.env RUN
   ```

3. Connect to PDB as apps user.
4. Run $CDR_TOP/patch/115/sql/cdrcreatecdruserrole.sql

Regenerate the JAR Files

1. Log on to Oracle LSH with the same user that was used for submitting the Post Install and the Lob Loader programs.
2. On the Home page, expand the **Life Sciences Data Hub** link.
3. Click on any link available to you, such as Home or Applications.
4. If the page is rendered properly, do not regenerate the JAR files. If the page is not rendered properly and throws some error, regenerate the JAR files by performing the following steps:
   a. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

      ```
      source /<BASE_LOCATION>/EBSapps.env RUN
      ```

   b. Run adadmin (provide SYSTEM and APPS password when prompted).
   c. Select **Generate Applications Files**.
   d. Select **Generate product JAR Files**.
   e. Type ALL for all products when prompted.
   f. Type Yes for the force generation of JAR files when prompted.
   g. Restart the Oracle LSH middle tier services.

Set Up Resource Capping

Resource capping of Oracle DMW and Oracle LSH jobs is required to ensure that you have enough resources available for the UI activity especially when jobs demand a high CPU time. If you do not set up the resource capping, the UI operations may suffer due to the CPU queue length and resource crunch. The system routes the data processing jobs to a database service

specifically defined for that purpose. A resource plan which limits the CPU usage of these jobs to 75% is applied on the sessions that connect to this service.

1. Connect to PDB as SYS user.
2. Create a database service. Perform one of the following:
   - For a non-RAC standalone database, run the following command. This service will be used for executing the data processing jobs (for example, XFM , Oracle InForm and so on).

     ```
     begin
     dbms_service.create_service('service_name','service_name');
     end;
     /
     begin
     dbms_service.start_service('service_name');
     end;
     /
     ```

     For example:

     ```
     SQL> begin
     dbms_service.create_service('dmwforjobsd31ctst','dmwforjobsd31ctst');
     end;
     /
     PL/SQL procedure successfully completed.

     SQL> commit;
     Commit complete.

     SQL> begin
     dbms_service.start_service('dmwforjobsd31ctst');
     end;
     /
     PL/SQL procedure successfully completed.

     SQL> commit;
     Commit complete.
     ```

   - For a RAC database, run the following command:

     ```
     srvctl add service -d CDB NAME -s service_name -pdb PDB NAME -preferred
     CDB Instance1,CDB Instance2
     srvctl start service -db CDB NAME -service service_name
     ```

     For example:

     ```
     srvctl add service -d dmw3dvc -s service_name -pdb dmw3dv -preferred
     dmw3dvc1,dmw3dvc2
     srvctl start service -db dmw3dvc -service service_name
     ```

3. Log in to the LSH application server.
4. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

```
source /<BASE_LOCATION>/EBSapps.env RUN
```

5. Navigate to `$CDR_TOP/patch/115/sql`.
6. Copy dmwRMSetup.sql to the database server stage location.
7. Log in to the database server and source the PDB environment file.
8. Connect to PDB as SYS user and execute the dmwRMSetup.sql script:

```
$ sqlplus sys/<password>@<PDB name> as sysdba

SQL> @dmwRMSetup.sql
```

   The script prompts for the database service created for the data processing jobs.
9. Enter the *service_name* you have created.

   The script prompts for the Oracle DMW PDB name.
10. Enter the Oracle DMW PDB name.
11. Type `Y`, and press **Enter**.

    The script creates a resource limit plan for the data processing jobs.

Turn off EXE DEBUG

1. Connect to PDB as apps user.
2. Enter:

```
SQL> select FND_PROFILE.VALUE_SPECIFIC('LSH:ENABLE_DEBUGGING') from dual;
 FND_PROFILE.VALUE_SPECIFIC('LSH:ENABLE_DEBUGGING')

--------------------------------------------------------------------------------
-----
 $YESNO$NO



SQL> select FND_PROFILE.VALUE_SPECIFIC('DMW:USER_DEBUGGING') from dual;
FND_PROFILE.VALUE_SPECIFIC('DMW:USER_DEBUGGING')
--------------------------------------------------------------------------------
-----
<NULL>
```

> **Note:**
>
> After you install Oracle DMW, you sign in and navigate to the Administration page and click the Tracing & Logging tab to see that you disabled logging and tracing in Active Sessions.

Set the Profile Value for DMW_SEND_BATCHSIZE

1. Open your Oracle LSH URL.

2. Log on with the system administrator account. An E-Business Suite screen opens.
3. On the left pane, expand the System Administrator (not System Administration) node, then the Profile node, and then click System.

   A new browser screen opens with several windows and the Find System Profile Values window is on the top.
4. In the **Profile** field, search for DMW_SEND_BATCHSIZE, and click **Find**.
5. In the **Site** column, enter a value of **10**.
6. In the **File** menu, select **Save and Proceed**.
7. Click **OK**, and close the window.

Set Up the Notification Mailer

To enable Oracle LSH to send Notifications to users' email address (as well as their Oracle LSH My Home page) you must set up an email account for the purpose of handling Notification responses and configure the Notification Mailer in the Oracle Workflow user interface. Follow instructions under Implementing Notification Mailers in the *Oracle Workflow Administrator's Guide*.

Set Up the Distributed Processing Server

The Distributed Processing (DP) Server is the mechanism Oracle LSH uses to communicate with the external processing engines that run some Oracle LSH jobs.

Install the DP Server on each computer where you have installed an external processing engine (such as SAS) and where you have installed XML Publisher. If you install multiple external processing engines on the same computer, you can install the DP Server once on that computer.

For information about the DP Server, see "Setting Up Services" in the *Oracle Life Sciences Data Hub System Administrator's Guide*. For information on integrating particular external systems with Oracle LSH, see Integrating Other Systems.

> **Note:**
>
> For Oracle DMW, the DP Server is required for File Watcher, for loading SAS and text data files. SAS files require the SAS processing engine and text files require the SQL*Loader, which is installed with Oracle Database.

Setting up the DP Server includes the following steps. You must do them in the following order:

- Create the Distributed Processing Server User Account
- Install the Distributed Processing Server
- Secure Distributed Processing Server Files
- Set NLS_LANG to UTF8
- Copy and Edit Files
- Define Service Locations and Services
- Start the DP Server

- Start the Message Queue
- Restart and Enable the Job Queue

Create the Distributed Processing Server User Account

You must run a script to create the Distributed Processing (DP) Server database account `cdr_dpserver` and set its password. Use this account to start the DP Server.

> **Note:**
>
> When you start the DP Server on each service location, you need this password. You should change the default password for use within your company.
>
> To change a password:
>
> 1. Log in to SQL*Plus.
> 2. Enter the following:
>
>   `alter user old_password identified by new_password`

To run the script:

1. Log in to the application server.
2. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

   `source /<BASE_LOCATION>/EBSapps.env RUN`
3. Go to $CDR_TOP/patch/115/sql.
4. Log in to SQL*Plus as apps
5. Run the script:

   `cdrcreatedpserveruser.sql`

   At the prompt, enter the password you want to use for the cdr_dpserver account.
6. Exit from SQL*Plus.

Install the Distributed Processing Server

On each computer where you have installed one or more processing engines for use with Oracle LSH, do the following to install the Oracle LSH Distributed Processing (DP) Server:

1. Create a home directory for the DP Server. It can be located anywhere on the computer where the DP Server resides. Oracle recommends naming it `DPServer_Home`.
2. In the DP Server Home directory, create two subdirectories: **lib** and **log**.

   The lib directory will hold the jar files the DP Server uses. The log directory will hold 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.
3. Change to the lib directory.
4. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

```
source /<BASE_LOCATION>/EBSapps.env RUN
```

5. Copy **DPServer.zip** from $CDR_TOP/jar to the `DPServer_Home/lib` directory.
6. Using GNU zip or another utility, unzip **DPServer.zip** into the lib directory. The DPServer.zip file contains the following files:
   - DPServer.jar
   - fileWatcherServer.jar
   - xmlparserv2.jar
   - aqapi.jar
   - jmscommon.jar
   - jta.jar
   - ojdbc8.jar
   - orai18n-mapping.jar
   - ucp.jar
7. Change directories to the DPServer_Home directory.
8. Create a working directory with a meaningful name for each service that will run on this machine. For example, if you will run SAS jobs on this computer, create a directory such as SASWORK. If you will also run Oracle Reports jobs on this computer, create another directory with a name like REPWORK.

   Each time one of these engines runs a job, the DP Server creates a directory containing the files required for the job and gives the directory the job ID as a name. When you define services in the Oracle LSH user interface, specify that you want the DP Server to create these job directories in the working directories you have created. For more details, see Define Service Locations and Services.
9. Set the TNS alias in the tnsnames.ora file to the global_name of the database server. This is required because the DP server runs jobs, such as SAS programs, that connect to the database server using the global_name.
10. On the DP Server machine, create a symbolic link from the location where SAS is installed to user home:

    ```
    ln -s  SAS_executalble_path/sas_u8 DP_Server_Home_path/sas
    ```
11. Ensure that JDK 1.8.0_281 is installed on each DP Server machine.

> **Note:**
>
> If you need to set up the DP Server outside the firewall, make sure the computer outside the firewall can connect to the database server inside the firewall. To do this, change a firewall setting to allow external access to the TNS listener port on the database server.

Secure Distributed Processing Server Files

The DP Server log files in the log directory may contain information that is sensitive to your organization. Oracle recommends granting full access to this directory only to the Oracle database user running the DP Server process and any other external processing engine user.

Set NLS_LANG to UTF8

On each Server where you install the DP Server, set the computer's NLS_LANG environment variable to UTF8.

- Windows
- UNIX

Windows

Check and set your NLS_LANG environment variable:

1. Right-click the **My Computer** icon on your desktop, then click **Properties**.
2. Click the **Advanced** tab, then click **Environment Variables**.
3. In **User Variables** and **System Variables**, check if there is a variable named NLS_LANG.
4. If there is an NLS_LANG variable, highlight it and click **Edit**.
5. Set the variable value to UTF8; for example: `AMERICAN_AMERICA.UTF8`

If you do not have the NLS_LANG environment variable, change your registry settings:

1. Click **Start**, then **Run**.
2. In the Run window, enter `regedit` and click **OK**.
3. Locate one of the following registry key entries:
   - HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE
   - HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*x*

     where *x* is the unique number identifying the Oracle home
4. Add a new key named NLS_LANG with a value including UTF8; for example:

   ```
   NLS_LANG=AMERICAN_AMERICA.UTF8
   ```

UNIX

Do the following:

1. Check the environment variable NLS_LANG:

   ```
   echo $NLS_LANG
   ```
2. Set the environment variable NLS_LANG to UTF8; for example:

   ```
   % setenv NLS_LANG American_America.UTF8
   ```

Copy and Edit Files

This section contains the following topics:

- Copy DP Server Files
- Edit the DP Server Start Script
- Make Scripts Executable
- Copy RTF Template Files for XML Publisher

- Copy, Edit, and Grant Permissions to Execution Command Files for Processing Engines

Copy DP Server Files

1. Go to the DP Server home directory you created when you installed the DP Server code.
2. Copy the following files from $CDR_TOP/admin/template to the DP Server home directory:
   - **cdr_apps_dpserver.sh** (or cdr_apps_dpserver.cmd for running Windows services such as OBIEE)
   - **checkJSapps.sh**
   - **stopJSapps.sh**
   - **killproc.sh**

> **Note:**
>
> Do not copy **killproc.sh** if the DP server is running on a Windows computer, for example, running the Oracle Analytics Server server.

Edit the DP Server Start Script

You must add local values to **cdr_apps_dpserver.sh** (or **cdr_apps_dpserver.cmd** on Windows) before you can start the DP Server.

1. Log in as the owner of the DP Server Home Directory.
2. Go to the DP Server home directory.
3. Edit cdr_apps_dpserver.sh (or cdr_apps_dpserver.cmd). Enter the actual value for each of the following:
   - **$DPSERVER_HOME**: Enter the full path for the DP Server home on this computer.
   - **SVC**. Enter the Service Location Name (not a Service name) that you defined or will define in the Service Location subtab for the Service Location that corresponds to this computer. (For more details, see Define Service Locations and Services.) The name is case-sensitive. For example:

     ```
     SVC=SERVICE_LOCATION_NAME
     ```

     > **Note:**
     >
     > On Windows you must enter this value at runtime.

   - **JDK Location (JDK_LOC)**: Enter the full path to the JDK 1.8 executable. For example:

     ```
     JDK_LOC=$ORACLE_HOME/jdk/bin
     ```

   - **JVM Arguments**: Add the following line immediately after the Java command (`COMMANDLINE=java`), which follows the RAC flag setting:

     ```
     -DFile.encoding=UTF8 -Duser.language=en -Duser.country=US
     ```

   - **Message Receive Interval**: Enter the value for 'Message Receive Interval' as 1,200,000 milliseconds(20 minutes).

For example:

```
Q_MRI=${9-1200000}
```

- **Debug Level (DEBUG_LEVEL)**: Default value is "all" (which is same as pre 3.3 releases). The recommended value is "low" to generate the minimal set of log statements. Setting the value to "medium"/"all" generates verbose log statements and the size of the log files is expected to grow faster.
  For example:

  ```
  DEBUG_LEVEL=low
  ```

You can accept the default values for all other variables. Some values must be set at runtime. See Start the DP Server for details.

Make Scripts Executable

Make all the scripts executable with the following command:

```
chmod 755 *.sh
```

Copy RTF Template Files for XML Publisher

The following steps are required only on the computer where you are running XML Publisher:

1. In the DP Server home directory, create a directory called **cdrtemplates**.
2. Copy the following files from $CDR_TOP/patch/115/publisher/templates to the new **cdrtemplates** directory:
   - cdr_output_summ_cs.rtf
   - lsh-title-page.rtf
   - lsh-toc-template.rtf
   - lsh-pagenum.rtf
   - lsh-template.rtf
   - lsh-blank-page.pdf

Copy, Edit, and Grant Permissions to Execution Command Files for Processing Engines

Do the following on each computer where you have installed a processing engine:

1. From $CDR_TOP/admin/template, copy the sample execution command script for each processing engine installed on the computer. You can copy the scripts directly into the DP Server Home directory or create a subdirectory for them.

   > **Note:**
   >
   > Keep a record of the absolute location of these scripts. You will need it when you define a service location for the computer. See Define Service Locations and Services for details.

   The scripts include:

- **cdrzip.sh** and **cdrunzip.sh**for Text Data Marts
- **sasNormal.sh** for SAS Programs
- **oraexp.sh** for Oracle Export Data Marts
- **orareprunner.sh** for Oracle Reports Programs
- **txtNormal.sh** for Text Load Sets
- **xmlprunner.sh** for post-processing Report Sets
- **xmlpreprunner.sh** for the Oracle LSH system reports and for cover sheets for outputs
- **obieeinstall.cmd** for Oracle Business Intelligence Business Areas—required only on the BI Server
- **obieedeploy.cmd** for Oracle Business Intelligence Business Areas— required only on the BI Server

2. Edit each script with information specific to the computer, for example:
   - Oracle SID
   - Location of the technology server
   - Location of Oracle setup script `coraenv`
   - Paths

   Ensure that environment variables are accessible to the DP server. For example, if the script refers to the variable $ORACLE_HOME, either define the variable or provide the full path in the script.

   > **Note:**
   >
   > If you run SAS programs, add instructions to sasNormal.sh to start SAS in UTF8 mode. (See Start SAS in UTF8 Mode for details.) In addition, include the DP Server Home path in the environment variable as shown:
   >
   > ```
   > PATH=$ORACLE_HOME/bin:$ORACLE_HOME/lib32: DP_Server_Home_Path:$PATH export
   > PATH
   > ```

3. Make all the scripts executable on the UNIX system with the following command:

   ```
   chmod 755 *.sh
   ```

Define Service Locations and Services

You must define Service Locations and Services in the Oracle LSH user interface for each computer where the Oracle LSH Distributed Processing (DP) Server will run. You define one service location for each computer, and at least one service for each engine or development environment that you want to run on that computer.

To define service locations and services you must have a user account with the Oracle LSH System Admin role assigned to it.

To log into Oracle LSH, do the following:

1. Open your web browser.
2. Enter the eBusiness Suite SSWA (Self-Service Web Application) URL as follows:

http://<host name>.<domain name>:<HTTP port>/oa_servlets/AppsLogin

For example:

http://*appshost*.*your_company*.com:8000/oa_servlets/AppsLogin

The Applications Login screen appears.

3. Enter the username and password associated with the LSH System Admin responsibility and click **Login**.
4. Under Navigator, click **Life Sciences Data Hub**. The system displays the list of Oracle LSH user interface locations to which you have access.
5. Click **Service Location**. The Service Location screen opens.

To define service locations and services in the Oracle LSH user interface, follow the instructions in "Setting Up Services" in the *Oracle Life Sciences Data Hub System Administrator's Guide*.

> **Note:**
>
> For Oracle DMW, you need one or two Service Locations. You need two services; **Text for SQL*Loader** and **SAS**. The two services can be on the same Service Location if it has access to both SQL*Loader and the SAS processing engine as well as the folders you will create to put data files into for loading into Oracle DMW. You can put text and SAS files in different locations.

Start the DP Server

To start the DP Server, do the following:

1. Log on as the owner of the DP Server Home Directory.
2. Run the script by entering the following command for UNIX. Information on the parameters is given below.

   ```
   ./cdr_apps_dpserver.sh ORACLE_SID DB_HOST DB_PORT RAC_TNS RAC_FLAG FW_ENABLED
   FW_FREQ FW_POLL
   ```

   or for Windows:

   ```
   c:> cdr_apps_dpserver.cmd ORACLE_SID DB_HOST DB_PORT RAC_TNS RAC_FLAG
   FW_ENABLED FW_FREQ FW_POLL
   ```

   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.

  > **Note:**
  >
  > At the time of publication, Release 3.0 is not certified with RAC.

  - 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.
- *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 LSHDB adxxxsdb.example.com 20502 NA NO-RAC NO 0 0
```

where:

- `LSHDB` is the Oracle SID
- `adxxxsdb.example.com` is the host
- `20502` is the port
- You may enter NA (Not Applicable) or any other value for RAC_TNS.
- NO-RAC is the setting for RAC_FLAG
- NO indicates that File Watcher is not enabled; Oracle DMW is not being used.
- 0 FileWatcher Refresh Frequency, since File Watcher is not enabled
- 0 FileWatcher Polling Frequency, since File Watcher is not enabled

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=AP1RAC.example.com)
(PORT=1521))(ADDRESS=(PROTOCOL=tcp)(HOST=AP2RAC.example.com)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=CDRXXX)))'  RAC NO 0 0
```

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=AP1RAC.example.com)(PORT=1521))(ADDRESS=(PROTOCOL=tcp)(HOST=AP3RAC.example.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=CDRXXX)))` is the JDBC connection string of the database server
- RAC is the setting for RAC_FLAG
- NO indicates that File Watcher is not enabled; Oracle DMW is not being used.
- 0 FileWatcher Refresh Frequency, since File Watcher is not enabled
- 0 FileWatcher Polling Frequency, since File Watcher is not enabled

3. The script prompts you for a password. Enter the password for the cdr_dpserver user.
4. Check if the DP server is running:

```
./checkJSapps.sh SID
```

> **Note:**
>
> Do not change the value of DB_USER.

Start the Message Queue

1. Connect to PDB as apps user.
2. Make sure the queue is stopped. View the log:

```
select MESSAGE from cdr_msg_queues_log order by log_message_id;
```

If the most recent statement is the following, the queue is stopped.

```
End Procedure cdr_exe_msg_submission.process_queues()
```

If not, stop the queue:

```
begin cdr_exe_msg_queues_admin.stop_processing_queues; end; /
```

Wait until you see the "End Procedure" statement in the log.

3. After the queue is stopped, start and enable the queue:

```
begin
cdr_exe_msg_queues_admin.start_processing_queues;
cdr_exe_msg_queues_admin.enable_job_processing_queue;
end;
/
```

4. Check that the queue is started and enabled:

```
select MESSAGE from cdr_msg_queues_log order by log_message_id;
```

The output should contain the following statements (there may be Submission statements after these):

```
Begin Procedure cdr_exe_msg_submission.process_queues()
dequeued from control Q: _MSGCONTROL_ENABLE
```

Restart and Enable the Job Queue

Still logged in as apps:

1. Stop and disable the Job Queue:

```
begin
cdr_exe_job_queues.stop_processing_queues;
end;
/
```

2. Start and enable the job queue:

```
begin
cdr_exe_job_queues.start_jobq_process_enabled;
end;
/
```

Set Up Client Computers

There are two types of client setups depending on the role of the Oracle LSH user:

- Consumers and Administrators
- Application Developers

For information on supported operating systems and browsers for the client, see *Oracle Life Sciences Applications Supported Technology Stacks* (Document ID 180430.1) on My Oracle Support.

- Consumers and Administrators
- Application Developers

Consumers and Administrators

Oracle LSH Consumers, who retrieve information in the form of reports and visualizations, and Oracle LSH Administrators, who perform administrative tasks within Oracle LSH, require the following on their personal computers:

- A Web browser (one of those supported by Oracle Applications)
- Java Virtual Machine (JVM)

Consumers require JVM if they are using Oracle Discoverer Plus to create data visualizations.

Administrators require either JInitiator or JVM to use any of the Oracle Forms screens related to security, to run the post-installation jobs or to set up user accounts or functional roles.

The first time a user opens one of the Oracle Forms screens, the user is prompted to download and install JVM if it is not already installed.

> **Note:**
>
> If you are currently using JInitiator, you can continue to do so.

Application Developers

An Oracle LSH Application Developer (also called a Definer) writes source code on his or her personal computer in an integrated development environment (IDE) and then uploads the source code file to Oracle LSH.

An Oracle LSH Definer client requires:

- A Web browser (one of those supported by Oracle Applications)
- Java Virtual Machine (JVM)
- A Zip utility: 7-Zip or WinZip Pro 11.2 SR-1, WinZip 8.1, or any other WinZip version that includes the WZUNZIP.exe
- Oracle LSH client plug-in (see Install the Client Plug-In)
- NLS_LANG environment variable set to UTF8 (see Set NLS_LANG to UTF8)
- One or more IDE clients (see Set Up Development Environments)

This section contains the following topics:

Install the Client Plug-In

Oracle LSH files contained in **CdrClientInstall.zip** handle the integration of the IDEs with Oracle LSH.

System Administrator Tasks

To prepare these files for installation by developers, do the following:

1. Download **CdrClientInstall.zip** from $CDR_TOP/plugin/sas.
2. Unzip **CdrClientInstall.zip**. One of the files unzipped is **cdrconfig.xml**.
3. (Optional) Edit **cdrconfig.xml** to add a directory path for each IDE developers may use, including the IDEs mentioned in Integrating Other Systems. and any other IDEs that your company is using with Oracle LSH, either by buying a third party adapter or developing your own adapter.

   If you are using 7-Zip instead of WinZip, follow instructions in Section 5.17.2.1.3, "Enabling Using 7-Zip".
4. Write the unzipped contents of **CdrClientInstall.zip** to a CD-ROM.

5. Give the CD-ROM to each Oracle LSH Definer who will use an IDE, with instructions for where to install the IDE software so that the directory paths you entered in **cdrconfig.xml** are correct.

> **Note:**
>
> Alternatively, do not edit **cdrconfig.xml** before writing it to the CD-ROM and tell each Definer to edit his or her own copy.

Definer Task

On each Definer's personal computer, load the CD-ROM that contains the unzipped files. InstallShield automatically runs **setup.exe**, which loads **cdrconfig.xml** and **cdrclient.exe** to a location the Definer specifies on his or her local computer. The default location is *ProgramFilesDir*\oracle\cdr where *ProgramFilesDir* is the registry entry for the value name ProgramFilesDir. If the specified location does not exist, the InstallShield creates it.

In addition, it sets the location for the CDR Work directory. By default this location is %USERPROFILE%/Application Data/CDR. Oracle recommends using this setting if you are installing the client IDE on a server for access by multiple users. It creates a separate work space for each user so they do not overwrite each other's files.

Enabling Using 7-Zip

If you want to use 7-Zip instead of WinZip, do one of the following:

- Edit cdrconfig.xml to Use 7-Zip If WinZip Is Not Installed
- Create a .bat File and Edit cdrconfig.xml
- Edit cdrconfig.xml When 7-Zip is the Only Option

Edit cdrconfig.xml to Use 7-Zip If WinZip Is Not Installed

Add the following code:

```
REM ------------------------------------------
REM wzunzip
REM ------------------------------------------
if  exist "%ProgramFiles%\WinZip\wzunzip.exe"
"%ProgramFiles%\WinZip\wzunzip.exe" -yo %1 %2 %3
if  not exist "%ProgramFiles%\WinZip\wzunzip.exe"  goto 7zip
goto end

:7zip
REM ------------------------------------------
REM 7-zip
REM ------------------------------------------
if  exist "%ProgramFiles%\7-Zip\7z.exe"  "%ProgramFiles%\7-Zip\7z.exe" x -y
%3
if  not exist "%ProgramFiles%\7-Zip\7z.exe" goto errormsg
goto end

:errormsg
msg "%username%" Either wzunzip nor 7z could be found. No Launch possible.

goto end

:end
```

Create a .bat File and Edit cdrconfig.xml

1. Create a file called wzunzip.bat with the following contents:

   ```
   "C:\Program Files\7-Zip\7z" x %3
   ```

2. Add the path to the wzunzip.bat to the CDRCONFIG.xml file.

Edit cdrconfig.xml When 7-Zip is the Only Option

Change this line in CDRCONFIG file:

```
<USER_DEFINE NAME="WINZIPEXE" VALUE="&quot;C:\Program Files\
WinZip\wzunzip.exe&quot;"/>
```

to:

```
<USER_DEFINE NAME="WINZIPEXE" VALUE="&quot;C:\Program Files\ 7-
Zip\7z.exe&quot;"/>
```

Set NLS_LANG to UTF8

Check and set your NLS_LANG environment variable:

1. Right-click the **My Computer** icon on your desktop, then click **Properties**.
2. Click the **Advanced** tab, then click **Environment Variables**.
3. In **User Variables** and **System Variables**, check if there is a variable named NLS_LANG.
4. If there is an NLS_LANG variable, highlight it and click **Edit**.
5. Set the variable value to UTF8; for example: `AMERICAN_AMERICA.UTF8`

If you do not have the NLS_LANG environment variable, change your registry settings:

1. Click **Start**, then **Run**.
2. In the Run window, enter `regedit` and click **OK**.
3. Locate one of the following registry key entries:
   - HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE
   - HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*x*

     where *x* is the unique number identifying the Oracle home
4. Add a new key named NLS_LANG with a value including UTF8; for example:

   ```
   NLS_LANG=AMERICAN_AMERICA.UTF8
   ```

Set Up Development Environments

Oracle LSH supports several integrated development environments (IDEs). For information on configuring these for use with Oracle LSH, see:

- Set Up SAS as an Integrated Development Environment
- Integrating Oracle SQL Developer or Oracle SQL*Plus with the Oracle Life Sciences Data Hub

For information on setting up the Oracle BI Administration Tool, see the *Oracle Life Sciences Data Hub System Administrator's Guide*.

Schedule Jobs

Schedule the various jobs.

1. Log in as PDB sys user and schedule the jobs as mentioned below.
2. Schedule the Study Health Monitor job.

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
   job_name           =>  'STUDY_HEALTH_REFRESH',
   job_type           =>  'STORED_PROCEDURE',
   job_action         =>  'APPS.DME_STUDY_HEALTH_GEN.REFRESH_ALL_STUDY_LC',
   repeat_interval    =>  'FREQ=MINUTELY;INTERVAL=15',
   auto_drop          =>   FALSE,
   job_class          =>  'DEFAULT_JOB_CLASS',
   comments           =>  'DMW study health StudyHealth data refresh job');
END;
/
exec dbms_scheduler.enable('APPS.STUDY_HEALTH_REFRESH');
```

3. Schedule the Discrepancy Dashboard job.

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
   job_name           =>  'DASHBOARD_REFRESH',
   job_type           =>  'STORED_PROCEDURE',
   job_action         =>  'APPS.DME_DASHBOARD_DISC.REFRESH_ALL_STUDY_LC',
   repeat_interval    =>  'FREQ=MINUTELY;INTERVAL=15',
   auto_drop          =>   FALSE,
   job_class          =>  'DEFAULT_JOB_CLASS',
   comments           =>  'DMW Dashboard data refresh job');
END;
/
exec dbms_scheduler.enable('APPS.DASHBOARD_REFRESH');
```

4. Schedule the Data Cleanup job.

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
   job_name           =>  'DATA_CLEANUP',
   job_type           =>  'STORED_PROCEDURE',
   job_action         =>  'APPS.dme_data_cleanup.CleanData',
   repeat_interval    =>  'FREQ=DAILY;INTERVAL=1',
   auto_drop          =>   FALSE,
   job_class          =>  'DEFAULT_JOB_CLASS',
   comments           =>  'DMW Data Cleanup job');
END;
/
exec dbms_scheduler.enable('APPS.DATA_CLEANUP');
```

5. Schedule the Study Data Load Dashboard job.

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
   job_name           =>  'SYNC_LOAD_SCHED_CONFIG',
   job_type           =>  'STORED_PROCEDURE',
   job_action         =>
'APPS.DME_DF_LOAD_SCHED_CONFIG.syncLoadSchedConfig',
   repeat_interval    =>  'FREQ=DAILY;INTERVAL=1',
   auto_drop          =>   FALSE,
   job_class          =>  'DEFAULT_JOB_CLASS',
   comments           =>  'DMW study health StudyHealth data refresh job');
END;
/
exec dbms_scheduler.enable('APPS.SYNC_LOAD_SCHED_CONFIG');
```

Run the Health Check Scripts

Run the Health Check scripts for Oracle LSH and Oracle DMW as described in My Oracle Support Article 2733714.1 (https://support.oracle.com).

6

Integrating Other Systems

The Oracle Life Sciences Data Hub (Oracle LSH) supports integration with other systems as sources of data, as means of visualizing and reporting on Oracle LSH data, and as processing engines for transforming Oracle LSH data. Oracle LSH handles integration with such systems with adapters custom made for each external system. Adapters for the systems included in this chapter are included with Oracle LSH.

You must install the adapters and grant them security privileges. These tasks are covered in Load the Adapter Files and Grant Security Rights to Seeded Adapters.

Each external system you choose to integrate with Oracle LSH requires installation and some additional setup, described here. The exception is Oracle Reports, which is installed as part of Oracle Applications and requires no further setup.

For information on supported versions of these products, see Integrated External Systems or for the most current information, see *Oracle Life Sciences Applications Supported Technology Stacks* (Document ID 180430.1) on My Oracle Support.

This section contains the following topics:

Integrating Oracle Clinical with the Oracle Life Sciences Data Hub

Oracle LSH 3.4.1 is compatible with Oracle Clinical 5.4.1 applied.

Oracle LSH includes a set of adapters custom-designed for the purpose of loading data and metadata from Oracle Clinical into Oracle LSH.

To load data and metadata from Oracle Clinical into Oracle LSH you must do the following:

1. In the Oracle LSH user interface, define a remote location and connection to each Oracle Clinical location from which you want to load data. See "Registering Locations and Connections" in the *Oracle Life Sciences Data Hub Administrator's Guide* for further information.
2. Define, check in, install, and run one or more Oracle Clinical Load Sets in Oracle LSH. See "Defining Load Sets" in the *Oracle Life Sciences Data Hub Application Developer's Guide* for further information.

   You can also use APIs to create Load Sets. These packages are documented in the *Oracle Life Sciences Data Hub Application Programming Interface Guide*.

Integrating SAS with the Oracle Life Sciences Data Hub

Oracle LSH is designed for close integration with SAS at several levels. You can load SAS data set files into Oracle LSH, use SAS as an integrated development environment to create Oracle LSH Programs with SAS source code, and use the SAS engine to run these Programs on Oracle LSH data.

Oracle LSH supports integration with SAS 9.4.

> **Note:**
>
> Oracle Life Sciences Data Hub does not support a SAS Windows server.

This section includes the following topics:

- SAS Compatibility
- Set Up SAS Job Execution
- Set Up Loading Data from SAS
- Set Up SAS as an Integrated Development Environment

SAS Compatibility

SAS with connectivity to Oracle LSH requires an Oracle 19 *c* library. Oracle recommends that the Oracle 19 *c* Client reside on the same server computer as the SAS installation.

To install Oracle LSH and SAS as a processing engine on the same computer:

1. Create a 19 *c* Oracle Home by installing the Oracle Client 19 *c*.
2. Install SAS.
3. Set the UNIX environment variable LD_LIBRARY_PATH to the 19*c* $*ORACLE_HOME*/lib.

Set Up SAS Job Execution

To enable running SAS jobs from Oracle LSH, you must integrate the SAS server with Oracle LSH and start the server in UTF8 mode. This section contains the following topics:

- Integrate the SAS Server with Oracle LSH
- Start SAS in UTF8 Mode

Integrate the SAS Server with Oracle LSH

Do each of the following:

1. Install SAS Access to Oracle on the SAS server.
2. Install the Oracle LSH Distributed Processing (DP) Server on the computer where the SAS server is installed. See Set Up the Distributed Processing Server.
3. Define a service location in Oracle LSH for the computer where the SAS server is installed. See "Defining Service Locations" in the *Oracle Life Sciences Data Hub System Administrator's Guide* for instructions.

4. Define one or more services for the service location. See "Defining Service Locations" in the *Oracle Life Sciences Data Hub System Administrator's Guide* for instructions.

Start SAS in UTF8 Mode

To help ensure that Oracle LSH stores and displays special characters in your data correctly, start SAS in UTF8 mode by editing the DP Server execution command file sasNormal.sh, which you copied and edited in Copy, Edit, and Grant Permissions to Execution Command Files for Processing Engines.

If you are using SAS 9.4, add lines to sasNormal.sh as in the following examples:

- In UNIX:

  ```
  sas -encoding UTF8
  ```

- In Windows, specify the version of SAS you are using. For example, for SAS 9.4:

  ```
  C:\Program Files\SAS\SASFoundation\9.4\sas.exe" -CONFIG C:\Program
  Files\SAS\SASFoundation\9.4\nls\u8\SASV9.CFG"
  ```

> **Note:**
>
> Do not include line breaks in your command. (The page width forces the line to break in the example shown.)

Set Up Loading Data from SAS

Oracle LSH includes an adapter custom-designed for the purpose of loading data from SAS database into Oracle LSH.

To enable loading data from SAS into Oracle LSH you must do the following:

1. Complete all steps listed in Set Up SAS Job Execution .
2. Make sure that the LOB Loader Oracle LSH post-installation job has been run; this job creates the SAS adapter. This is a required step in the installation of Oracle LSH; see Load the Adapter Files.
3. Make sure that you have followed instructions in Grant Security Rights to Seeded Adapters.
4. Assign at least one user group to the SAS adapter. See "Setting Up Adapters to External Systems" in the *Oracle Life Sciences Data Hub System Administrator's Guide*.

Set Up SAS as an Integrated Development Environment

To use SAS as an integrated development environment (IDE) each Definer must do the following on his or her local PC:

- Install SAS in the location specified by the system administrator.
- Install the Oracle LSH client plug-in by inserting the CD-ROM supplied by the system administrator.

  InstallShield runs **cdrclient.exe**, which loads **cdrconfig.xml** and **cdrclient.exe** either to the default location or to a location the Definer specifies on his or her local computer.
- Ensure that **cdrconfig.xml** has the correct directory path for the SAS executable.

- Set the user preference for the SAS connection mode (details below). Instructions are in the "SAS Connection Type" section of the Getting Started chapter of the *Oracle Life Sciences Data Hub User's Guide*.
- Install any software required to support the preferred connection mode (details below).
- Set the NLS_LANG environment variable or registry settings to support UTF8 character encoding; see Set the NLS_LANG Environment Variable to UTF8.

SAS Connection Modes: SAS can work as an integrated development environment (IDE) in different ways. Each user must set a preference for the way he or she wants to work. Oracle LSH supports the following connection modes:

- **Connected Mode**. The Definer has the SAS client installed on his or her personal computer. When he or she launches SAS as an IDE from an Oracle LSH Program, Oracle LSH downloads views based on the source Table Descriptors defined in the Program. The Definer works locally on the SAS client, using the views to read current data in Oracle LSH. The Definer's SAS program can write to local SAS data sets. When the SAS program is ready, the Definer goes into the Oracle LSH Program and uploads the SAS source code as an Oracle LSH Source Code file.

  The client must use the SAS Access to Oracle tool to connect to Oracle LSH.
- **SAS Connected Mode with Work Area Data**. This mode is the same as Connected mode except that it connects to the Work Area schema in the database. From SAS, the user can browse views of current data in all Table instances in the Work Area, not just the Table instances linked to Table Descriptors of the Program.

  The client must use the SAS Access to Oracle tool to connect to Oracle LSH.
- **Disconnected Mode**. The Definer has the SAS client installed on his or her personal computer. When the Definer launches SAS as an IDE, Oracle LSH uses the Distributed Processing Server to download the current data in the Table instances mapped to source Table Descriptors in the Program into the Definer's local SAS environment, creating data sets with the same structure as the Oracle LSH Table Descriptors. The Definer's SAS program can read from and write to local SAS data sets. When the SAS program is ready, the Definer goes into the Oracle LSH Program and uploads the SAS source code as an Oracle LSH Source Code file.

  No connection other than a network is required to the Oracle LSH Distributed Processing Server.

Integrating Oracle SQL Developer or Oracle SQL*Plus with the Oracle Life Sciences Data Hub

To use Oracle SQL Developer or SQL*Plus as an IDE for Oracle LSH PL/SQL Programs, each Definer must do the following on his or her local PC:

- Install Oracle SQL Developer or SQL*Plus in the location specified by the system administrator.
- Install the Oracle LSH client plug-in by inserting the CD-ROM supplied by the system administrator.
- Ensure that **cdrconfig.xml** has the correct directory path for the Oracle SQL Developer (or SQL*Plus) executable.

> **Note:**
>
> If the Definer has both Oracle SQL Developer and SQL*Plus installed on the local computer, he or she can switch between the two IDEs by changing the executable directory path in **cdrconfig.xml**.

- Start Oracle SQL Developer or SQL*Plus and create a connection to the Oracle LSH database. The username and password for this connection must be those of an Oracle LSH database user account.
- Follow the steps in Set the NLS_LANG Environment Variable to UTF8.
- Install Winzip Pro 11.2 SR-1, Winzip 8.1, or any other Winzip that includes the WZUNZIP.exe

This section contains the following topic:

Set the NLS_LANG Environment Variable to UTF8

To set an Oracle client application like SQL*Plus to use the right encoding, you must set the environment variables on the client machine to UTF8. The required settings vary, depending on the operating system.

Windows

Check and set your NLS_LANG environment variable:

1. Right-click the **My Computer** icon on your desktop, then click **Properties**.
2. Click the **Advanced** tab, then click **Environment Variables**.
3. In **User Variables** and **System Variables**, check if there is a variable named NLS_LANG.
4. If there is an NLS_LANG variable, highlight it and click **Edit**.
5. Set the variable value to UTF8; for example: `AMERICAN_AMERICA.UTF8`

If you do not have the NLS_LANG environment variable, change your registry settings:

1. Click **Start**, then **Run**.
2. In the Run window, enter `regedit` and click **OK**.
3. Locate one of the following registry key entries:
   - HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE
   - HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*x*

     where *x* is the unique number identifying the Oracle home
4. Add a new key named NLS_LANG with a value including UTF8; for example:

   ```
   NLS_LANG=AMERICAN_AMERICA.UTF8
   ```

UNIX

Do the following:

1. Check the environment variable NLS_LANG:

   ```
   echo $NLS_LANG
   ```

2. Set the environment variable NLS_LANG to UTF8; for example:

   ```
   % setenv NLS_LANG American_America.UTF8
   ```

Integrating Oracle Business Intelligence Enterprise Editions (OBIEE) for Visualizations

You can create Oracle LSH Business Areas of type Oracle Analytics Server to make data available to visualizations in OBIEE Answers. Definers can install the Administrator's Tool on their PC to develop more complex OBIEE Repository (.rpd) files to support more complex data visualizations. Users can then launch the Oracle Business Intelligence Dashboard through Oracle LSH or through a URL to see data visualizations.

> **Note:**
>
> Additional configuration is required. See the chapter on Oracle Business Intelligence visualizations in the *Oracle Life Sciences Data Hub System Administrator's Guide*.

Oracle LSH supports OBIEE 12.2.1.4 for visualizations. You must install the Distributed Processing (DP) Server on each machine where the Oracle BI server is installed.

- Install Oracle Analytics Server 12.2.1.4 for Visualizations

Install Oracle Analytics Server 12.2.1.4 for Visualizations

Install Oracle Analytics Server to support visualizations as described in the following topics:

- Install Oracle Analytics Server
- Disable Oracle Analytics Server 12c Enterprise Manager SSO authentication
- Start the WebLogic Server
- Install the Oracle LSH DP Server
- Deploy Repository File
- Set Up Oracle Analytics Server Visualizations

Install Oracle Analytics Server

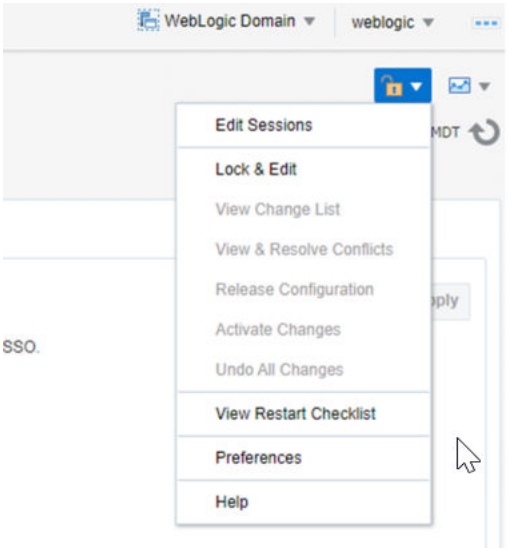
Install Oracle Analytics Server 12.2.1.4 using *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence 12c Release 12.2.1.3 (E83388-01)*.

> **Note:**
>
> As noted in *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence 12c Release 12.2.1.3*, you need to install Oracle Fusion Middleware Repository Creation Utility 12c (12.2.1.3) before installing Oracle Analytics Server 12c.

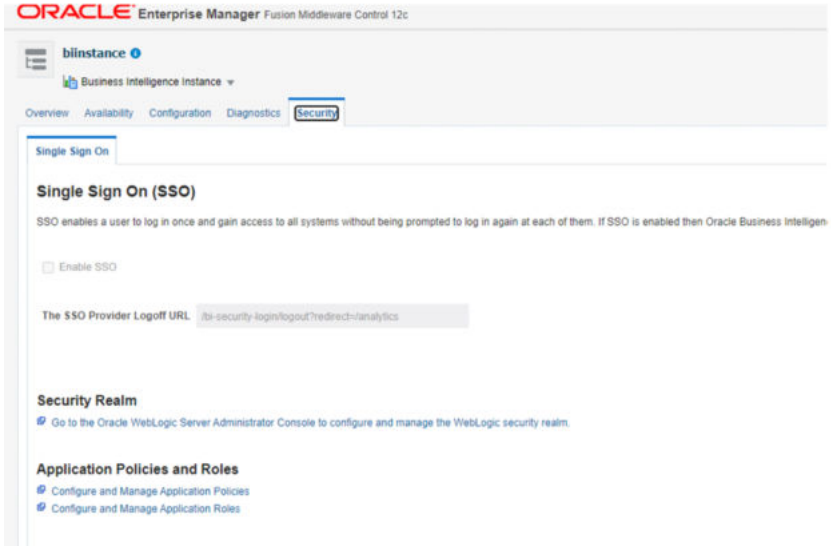
Disable Oracle Analytics Server 12c Enterprise Manager SSO authentication

With Oracle Analytics Server 12c, SSO authentication is enabled by default. For Oracle LSH users to access Analytics, disable the SSO authentication as described in the following procedure.

1. Log in to Oracle Analytics Server 12c Enterprise Manager using as a WebLogic user.
2. From the Target navigation menu on left side of the screen, click **biinstance** under the Business intelligence folder.
3. Click the **Security** tab of Business Intelligence instance. Then click **Lock & Edit** from the **Active Edit Session** menu on the right side of the screen as shown:



4. Disable SSO by clearing the **Enable SSO** checkbox as shown. Then click **Activate Changes** from the **Active Edit Session** drop-down menu to apply the changes in Business Intelligence server.



5. Restart the WebLogic server to activate the changes as described in Start the WebLogic Server.

Start the WebLogic Server

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

1. On the Business Intelligence server computer, traverse to `<bi_home>\user_projects\domains\bi\bitools\bin` directory.
2. Do one of the following:
   - To start the WebLogic server and Business Intelligence server, execute the **start.cmd** script.
   - To stop the WebLogic server and Business Intelligence server, execute the **stop.cmd** script. If you need to restart it, execute the **start.cmd** once the stop command completes successfully.

Install the Oracle LSH DP Server

You must install the Oracle LSH DP Server on the same machine; see Set Up the Distributed Processing Server.

- Copy and Edit Execution Command Scripts

Copy and Edit Execution Command Scripts

The obieedeploy.cmd and obieeinstall.cmd command script files should be modified for Oracle Analytics Server 12c integration. The templates for these command scripts are available under $cdr/admin/template. Copy the Oracle LSH versions of obieedeploy.cmd and obieeinstall.cmd from $cdr/admin/templates to the Oracle LSH Distributed Processing Server location and referred to in the Execution Command under the Oracle Analytics Server Install and Deploy service configured in Oracle LSH under the Oracle Analytics Server Service Location.

Edit **obieeinstall.cmd** by configuring the following environment variables based on their values applicable for the local Oracle Analytics Server environment.

- `set PATH=%PATH%;C:\bi_home\bi\bifoundation\server\bin`
- `set ORACLE_BI_APPLICATION=coreapplication`
- `set ORACLE_BI_INSTANCE=C:\bi_home`
- `set COMPONENT_NAME=coreapplication_obis1`
- `set COMPONENT_TYPE=OracleServerComponent`
- `set ORACLE_INSTANCE=C:\bi_home`
- `set DOMAIN_HOME=C:\bi_home\user_projects\domains\bi`

Edit **obieedeploy.cmd** by setting the RPD_DIR environment variable that indicates the Local Folder path where the Oracle LSH RPD will be copied during the Oracle Analytics Server BA install from Oracle LSH. Similar to Oracle Analytics Server 11g, in Oracle Analytics Server 12c integration, this folder should point to any local folder path on the Business Intelligence server machine's file system.

- `set RPD_DIR=C:\RPD`

Deploy Repository File

In Oracle LSH 3.0 (similar to Oracle Analytics Server 11g), the RPD corresponding to the Oracle Analytics Server BA is just copied on to a local folder on the Oracle Business Intelligence server machine. The RPD folder path is defined in the obieedeploy.cmd. The RPD copied in the RPD folder is required to be deployed manually. Unlike Oracle Analytics Server 11g, the Business Intelligence server services need not be restarted manually through Oracle Enterprise Manager. In Oracle Analytics Server 12c, RPD need to be uploaded through command line tool.

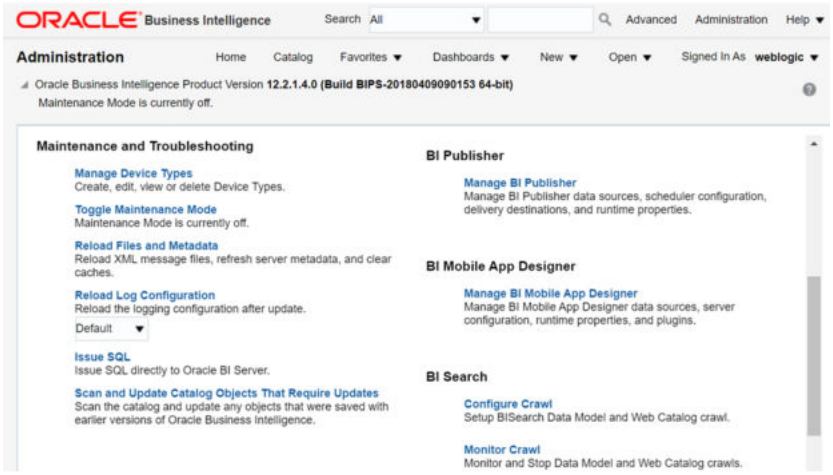
1. Run this command to upload the Oracle LSH RPD to the Oracle Analytics Server 12c Business Intelligence server:

   ```
   %ORACLE_BI_INSTANCE%\user_projects\domains\bi\bitools\bin\datamodel.cmd
   uploadrpd -I <RPD filename> -W <RPD_Password> -SI <Service Instance> -U
   <Weblogic User> -P <Weblogic User Password>
   ```

   For example:

   ```
   datamodel.cmd uploadrpd -I
   C:\LSH_DP_SERVER\RPD\dmw30ut12c_win2016.rpd -W Password1 -SI ssi -U
   weblogic -P Password2
   ```

2. After you upload the RPD file to the Oracle Business Intelligence server, log in in to the Business Intelligence server Analytics page as a WebLogic user.
3. Click **Administration** from the right side of the screen. In Administration page, click **Reload Files and Metadata** to refresh the uploaded RPD file in the Oracle Business Intelligence server for further analytics use.



Set Up Oracle Analytics Server Visualizations

See Chapter 11, "Setting Up Oracle Business Intelligence Visualizations" in the *Oracle Life Sciences Data Hub System Administrator's Guide*, for instructions.

7

Supported Upgrade Paths

For upgrading to Oracle LSH release 3.4.1, the following upgrade paths are supported:

- If you are upgrading from a release prior to Oracle LSH release 3.3.x:
  1. Upgrade to Oracle LSH release 3.3.x. Follow instructions in Oracle Life Sciences Data Hub Installation Guide.
  2. Upgrade to Oracle LSH release 3.4.1. Follow instructions in Upgrading to Oracle LSH Release 3.4.1.
- If you are upgrading from Oracle LSH release 3.3.x, follow instructions in Upgrading to Oracle LSH Release 3.4.1.

8

Upgrading the Thesaurus Management System (TMS) Database to Release 5.4.1

Upgrade the Oracle TMS database components required to support the Oracle LSH classification system. To upgrade to TMS 5.4.1:

1. Upgrade the Oracle TMS Database Server Code on a Windows computer as described in Oracle Thesaurus Management System Installation Guide for release 5.4.1.
2. Upgrade the Oracle TMS Database to 5.4.1 as described in Oracle Thesaurus Management System Installation Guide for release 5.4.1.

9

Upgrading to Oracle LSH Release 3.4.1

This section includes the following topics:

Run the Start Maintenance Script

Before you upgrade any of the software, follow the steps in this procedure to start the maintenance activity.

To run the Start Maintenace script:

1. Download or copy the maintenance start script file from the Oracle LSH application server (EBS Middle Tier server) `$CDR_TOP/patch/115/sql/cdrmaintstart.sql` to the database server's ORACLE_HOME location or any other preferred location.
2. From the database server, log in to SQL*Plus (not SQL Developer) as the APPS database user.
3. Enter this command to stop the study health monitor scheduler:

   ```
   SQL> EXECUTE DBMS_SCHEDULER.DISABLE('STUDY_HEALTH_REFRESH', FORCE
   => TRUE);
   ```

4. Enter this command to execute the script:

   ```
   SQL> @cdrmaintstart.sql
   ```

5. Check the log file.

   The log file validates the success of the Start Maintenance process and provides a maintenance ID. For example, it lists messages to show what it found up and running and what it stopped (such as the message queue and job queue).

Stop Server Processes

This step is required for all upgrade paths.

Before you begin the upgrade, stop the following servers:

- Oracle LSH Distributed Processing (DP) Server
- Application Server

Back Up the Oracle E-Business Suite Database

Oracle recommends that you make a cold backup of the Oracle E-Business Suite database in case you encounter problems during the upgrade process. You can use the backup to restore the database (if necessary).

> **Note:**
>
> Shut down the database using the NORMAL option to ensure you can use the backup to restore the database. Do not use the IMMEDIATE or ABORT option to shut down the database.

Database Tier Preinstallation Steps

This section contains the following topics:

Abort the Long-Running Jobs

1. Log in to SQL*Plus as **apps**.
2. Run the following query to find the long running jobs:

```
select * from cdr_jobs where job_status_rc like '%EXECUTING%';
```

3. Log in to the Oracle LSH application.
4. Navigate to **Life Sciences Data Hub** > **Job Execution**.
5. Search for the job using the job ID and cancel it.

Cancel the Sessions Holding Locks on Application Objects

1. Run the following query:

```
SELECT 'alter system kill session'||''''||sid||','||serial#||''''||'
immediate;'from v$session where sid in ( select session_id FROM
```

```
sys.dba_ddl_locks
WHERE (name like 'CDR_%' or name like 'DME_%') );
```

2. Run the alter statement to cancel the active sessions.

Disconnect Blocking Sessions

Before you upgrade, check for and stop any current database sessions.

1. If WebLogic Server is running, stop it. See *Oracle® Fusion Middleware Administering Server Startup and Shutdown for Oracle WebLogic Server* (12.2.1.4) at https://docs.oracle.com/en/middleware/fusion-middleware/12.2.1.4/asadm/starting-and-stopping.html#GUID-B57BE53D-F90C-42FB-9B73-27A06AE3768B.

   Log files for the AdminServer and the DMWServer are located in:

   ```
   middleware_home/user_projects/domains/DMWDomain/servers/AdminServer/logs
   ```

   and

   ```
   middleware_home/user_projects/domains/DMWDomain/servers/DMWServer/logs
   ```

2. Log in to SQL*Plus as **apps**.
3. Run the following query to find current sessions:

   ```
   SELECT 'USER: '||s.username||' SID: '||s.sid||' SERIAL #: '||S.SERIAL#
   "USER
   HOLDING LOCK", s.inst_id
   FROM gv$lock l
   ,dba_objects o
   ,gv$session s
   WHERE l.id1 = o.object_id
   AND s.sid = l.sid
   AND o.owner = 'CDR'
   AND o.object_name = 'DME_DISC_WORKTABLIST';
   ```

   This query returns the user, SID, serial number, and instance ID of each current session.
4. Disconnect each current session, passing in the SID and serial number:

   ```
   alter system disconnect session 'SID, serial_number' IMMEDIATE
   ```

Upgrade to Oracle LSH 3.4.1

You must upgrade Oracle LSH 3.4.1 as a patch to Oracle Applications. The patch **36272413** is on the media pack.

> **Note:**
>
> The information on how to install the patch and set up the cleanup job are covered under document IDs 3019034.1 and 2925664.1 respectively on My Oracle Support. Contact Life Sciences Support to get these documents.

1. Locate p36272413_R12_GENERIC.zip in the staging area.
2. Unzip p36272413_R12_GENERIC.zip to $NE_BASE/EBSapps/patch.
3. Apply the patch. For information on how to apply the patch, see *How To Apply Patches for LSH/DMW in Downtime Mode* (Document ID 3019034.1) on My Oracle Support.
4. Set up the cleanup job. For information on how to set up the cleanup job, see document ID 2925664.1 My Oracle Support.
5. Log in to the database as APPS and execute the following script:
   `$CDR_TOP/patch/115/sql/dmeErrLogChanges.txt`

   The script alters the name of the internal table “dme_err_log” and creates a public view on the table. The script creates a trigger on the view to prevent the unauthorized users from inserting or updating records from the script.

Install the CdrRuntime.jar File

1. On the Oracle LSH server, navigate to the $CDR_TOP/jar folder. It contains the CdrRuntime.zip file.
2. Copy the CdrRuntime.zip file to the Oracle database server in any temporary location. Then enter this command to unzip the file and extract CdrRuntime.jar:

   ```
   unzip CdrRuntime.zip
   ```

3. Run the following commands as an Oracle home owner on the Oracle database server to load the java files to the database.

   First, execute the following command:

   ```
   dropjava -force -thin -user apps/<password>
   @(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<DB hostname>)(PORT=<DB port
   number>))(CONNECT_DATA=(SERVICE_NAME=<DB service name>))) CdrRuntime.jar
   ```

   Then, execute this command:

   ```
   loadjava -force -thin -user apps/
   <password>@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<DB host name>)(PORT=<DB
   port number>))(CONNECT_DATA=(SERVICE_NAME=<DB service name>))) CdrRuntime.jar
   ```

4. Run this query to return a set of alter java commands that should be run to compile invalid classes:

   ```
   SELECT
   'alter java class "'
   || object_name
   || '" compile;'
   FROM
   dba_objects
   WHERE
   object_type = 'JAVA CLASS'
   and object_name LIKE '%cdr%'
   AND status = 'INVALID';
   ```

5. Connect to PDB as APPS user and run the statements returned by the above query.
6. Rerun the query mentioned in step 4 and confirm that it does not return any rows.

> **Note:**
>
> If the query returns rows, execute the statements returned again. Repeat this process until the SQL does not return any rows.

Run the Post-Installation Programs

You must run the Oracle LSH LOB Loader and post-installation concurrent programs.

> **Note:**
>
> If you are using RAC, shut down all but one database node before running the post-installation programs. If you leave more than one node up, the jobs may run successfully but you may get errors.

Log On to Oracle Applications

To run the jobs, you must log on to Oracle Applications as an Oracle LSH user with the following roles:

- LSH Setup Admin
- LSH Adapter Security Admin

1. Open your Web browser.
2. Enter the eBusiness Suite SSWA (Self-Service Web Application) URL as follows:

   http://<host name>.<domain name>:<HTTP port>/oa_servlets/AppsLogin

   For example:

   http://*appshost*.*your_company*.com:8000/oa_servlets/AppsLogin

   The application's Login screen appears.
3. Log in as a user with LSH Setup Admin and LSH Adapter Security Admin privileges. The Oracle Applications Home page appears.
4. Select the **LSH Setup Admin** Responsibility by clicking on it. Lookups appears in the second column.
5. Click **Lookups**. If necessary, click **Grant This Session** or **Grant Always**.
6. From the View menu, select **Requests**.

Load the Adapter Files

Oracle LSH includes predefined adapters that control the interaction between Oracle LSH and other systems. The Oracle LSH LOB Loader concurrent program finds all the adapter SQL files and loads them into a table in Oracle LSH.

To run the Oracle LSH LOB Loader:

1. Follow steps in Log On to Oracle Applications.
2. Click **Submit a New Request**. The Submit a New Request window opens.
3. Select **Single Request** and click **OK**.
4. Click the gray LOV button on the right of the **Name** field.
5. Select **LSH LOB Loader Concurrent Program** and click **OK**.
6. Click **Submit**.
7. Click **OK**.
8. Click **No**.

To monitor the concurrent program's progress:

1. Click **Find**.
2. Click **Refresh Data** periodically to update the execution phase and status displayed on screen.

When the status is Complete, you can view the log file by clicking the **View Log** button.

> **Note:**
>
> - Always check the log file because the phase may be Complete and the status Normal, but the program may not have successfully completed all its tasks.
> - If you see this message in the log file: "ORA-20001: APP-FND-02901: You do not have access privilege to any operating unit. Please check if your profile option MO: Security Profile includes any operating unit or the profile option MO: Operating Unit is set. has been detected in MO_GLOBAL.INIT," see *ORA-20001: APP-FND-02901 Errors Running Collections or Plan in 12.2 OR in Trying to View Request Log from Legacy Collections Self Service In R12.0/12.1* (Document ID 981828.1) on My Oracle Support.
> - If you see this message in the log file: "ERROR: LOBLoaderCP.runProgram() Exception String index out of range: -1", see *Problems Encountered During the Installation of LSH 2.2, Upgrade to LSH 2.2, and Execution of its Verification Tests* (Document ID 1327829.1) on My Oracle Support.

Run the Post-Installation Concurrent Program

Run the Job: To run the Oracle LSH post-installation concurrent process:

1. Click **Submit a New Request** under one of the following circumstances:
   - In the Requests window you used to monitor the Oracle LSH LOB Loader concurrent process.

- After following the steps in Log On to Oracle Applications.

2. Select **Single Request** and click **OK**.
3. Click the gray LOV button on the right of the **Name** field.
4. Select **LSH: Post Installation Program** and click **OK**.
5. Enter values for the following parameters:

> **Note:**
>
> Carefully set the following parameters. You cannot change the values for some of the parameters after you run the job.

- **Company ID**. The company ID serves as part of the primary key for all the Oracle LSH objects you define in this instance of Oracle LSH.
- **Owning Location**. Enter the name of your Oracle application instance.
- **Object Sequence Start Value**. Enter a single digit. The system will end all object IDs with this digit to further distinguish objects created in this Oracle LSH instance.
- **Object Sequence Start Value**. Leave blank. The system will end all object IDs with the number 1.
- **Database Host Name**. Enter the machine name of the database server instance.
- **Database Port Number.** Enter the port number of the database server instance.

6. Click **Submit**.
7. Note the job ID and click **No**.

Monitor the Process: To monitor the concurrent process's progress:

1. Click **Find**. Use the job ID to search for the process.
2. Click **Refresh Data** periodically to update the execution phase and status displayed on screen.
3. When the phase is Complete, click **View Log**.

> **Note:**
>
> Always check the log file, because the phase may be Complete and the status Normal and yet the process may not have successfully completed all its tasks. In such case, contact Life Sciences Support.

Check the log file to make sure it did the following:

- Set the company ID.
- Set the owning location.
- Recreated the cdr_object_id_seq with the start value you provided.
- Inserted one record each in the cdr_namings and cdr_naming_versions tables for the instance domain.

> **Note:**
>
> The job does the above only the first time it runs.

- Set the profile to check if the post-installation has been run for this site.

Grant Security Rights to Seeded Adapters

In order to ensure that seeded adapters have the security rights they need to call APIs, do the following:

1. Log in to the application server.
2. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

   ```
   source /<BASE_LOCATION>/EBSapps.env RUN
   ```

3. Connect to PDB as apps user.
4. Run $CDR_TOP/patch/115/sql/cdradaptergrants.sql

Remove EBS Audit and Create New Auditing Process

This section is applicable only if you are upgrading from a release prior to 3.4. It is not applicable if you are upgrading from release 3.4 to 3.4.1.

> **Note:**
>
> You can execute the cdr34audpostprocess.sql script only once in the environment.

1. Log in to the application server.
2. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

   ```
   source /<BASE_LOCATION>/EBSapps.env RUN
   ```

3. Connect to PDB as apps user.
4. Run $CDR_TOP/patch/115/sql/cdr34audpostprocess.sql.

Post-Upgrade Database Tasks

Perform the following tasks in Oracle Database 19c:

- Apply patch 32940955, if not already applied.
- Set the following values:
  - SGA_MAX_SIZE = 126 GB (after checking free huge pages)
  - SHARED_POOL_SIZE = 26 GB (at the CDB level)
  - Reset the SHARED_POOL_SIZE to 0 (at the PDB level) by executing the following command **from PDB**:
    ```
    alter system reset shared_pool_size scope=both;
    ```

- Execute the following commands:

```
alter session set container="CDB$ROOT"
alter system set "_gc_persistent_read_mostly"=false scope=spfile;
```

- Log in as apps account and execute the following script:

```
SET serveroutput ON
declare
  status1 boolean;
begin
  status1 := cdr_profiles_pkg.SAVE
('DMW:Set Based Processing Enabled','$YESNO$NO','SITE');
  if status1 then
   dbms_output.put_line('Success');
  else
   dbms_output.put_line('Failure');
  end if;
end;
/
commit;
```

> **Note:**
>
> Make sure that executing the following query returns the result as $YESNO$NO:
>
> ```
> select cdr_profiles_pkg.value
> ('DMW:Set Based Processing Enabled') from dual;
> ```
>
> If the result of the query is not $YESNO$NO, contact Life Sciences Support.

Set Up the Distributed Processing Server

The Distributed Processing (DP) Server is the mechanism Oracle LSH uses to communicate with the external processing engines that run some Oracle LSH jobs.

Install the DP Server on each computer where you have installed an external processing engine (such as SAS) and where you have installed XML Publisher. If you install multiple external processing engines on the same computer, you can install the DP Server once on that computer.

For information about the DP Server, see "Setting Up Services" in the *Oracle Life Sciences Data Hub System Administrator's Guide*. For information on integrating particular external systems with Oracle LSH, see Integrating Other Systems.

> **Note:**
>
> For Oracle DMW, the DP Server is required for File Watcher, for loading SAS and text data files. SAS files require the SAS processing engine and text files require the SQL*Loader, which is installed with Oracle Database.

Setting up the DP Server includes the following steps. You must do them in the following order:

- Create the Distributed Processing Server User Account
- Install the Distributed Processing Server
- Secure Distributed Processing Server Files
- Set NLS_LANG to UTF8
- Copy and Edit Files
- Define Service Locations and Services
- Start the Distributed Processing Server
- Start the Message Queue
- Restart and Enable the Job Queue

Create the Distributed Processing Server User Account

You must run a script to create the Distributed Processing (DP) Server database account `cdr_dpserver` and set its password. Use this account to start the DP Server.

> **Note:**
>
> When you start the DP Server on each service location, you need this password. You should change the default password for use within your company.
>
> To change a password:
>
> 1. Log in to SQL*Plus.
> 2. Enter the following:
>
>   `alter user old_password identified by new_password`

To run the script:

1. Log in to the application server.
2. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

   `source /<BASE_LOCATION>/EBSapps.env RUN`
3. Go to $CDR_TOP/patch/115/sql.
4. Log in to SQL*Plus as apps
5. Run the script:

   `cdrcreatedpserveruser.sql`

   At the prompt, enter the password you want to use for the cdr_dpserver account.
6. Exit from SQL*Plus.

Install the Distributed Processing Server

On each computer where you have installed one or more processing engines for use with Oracle LSH, do the following to install the Oracle LSH Distributed Processing (DP) Server:

1. Create a home directory for the DP Server. It can be located anywhere on the computer where the DP Server resides. Oracle recommends naming it `DPServer_Home`.
2. In the DP Server Home directory, create two subdirectories: **lib** and **log**.

   The lib directory will hold the jar files the DP Server uses. The log directory will hold 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.
3. Change to the lib directory.
4. Source the RUN file system environment file from the Oracle LSH application server by using the following command:

   ```
   source /<BASE_LOCATION>/EBSapps.env RUN
   ```
5. Copy **DPServer.zip** from $CDR_TOP/jar to the `DPServer_Home/lib` directory.
6. Using GNU zip or another utility, unzip **DPServer.zip** into the lib directory. The DPServer.zip file contains the following files:
   - DPServer.jar
   - fileWatcherServer.jar
   - xmlparserv2.jar
   - aqapi.jar
   - jmscommon.jar
   - jta.jar
   - ojdbc8.jar
   - orai18n-mapping.jar
   - ucp.jar
7. Change directories to the DPServer_Home directory.
8. Create a working directory with a meaningful name for each service that will run on this machine. For example, if you will run SAS jobs on this computer, create a directory such as SASWORK. If you will also run Oracle Reports jobs on this computer, create another directory with a name like REPWORK.

   Each time one of these engines runs a job, the DP Server creates a directory containing the files required for the job and gives the directory the job ID as a name. When you define services in the Oracle LSH user interface, specify that you want the DP Server to create these job directories in the working directories you have created. For more details, see Define Service Locations and Services.
9. Set the TNS alias in the tnsnames.ora file to the global_name of the database server. This is required because the DP server runs jobs, such as SAS programs, that connect to the database server using the global_name.
10. On the DP Server machine, create a symbolic link from the location where SAS is installed to user home:

    ```
    ln -s  SAS_executalble_path/sas_u8 DP_Server_Home_path/sas
    ```
11. Ensure that JDK 1.8.0_281 is installed on each DP Server machine.

> **Note:**
>
> If you need to set up the DP Server outside the firewall, make sure the computer outside the firewall can connect to the database server inside the firewall. To do this, change a firewall setting to allow external access to the TNS listener port on the database server.

Secure Distributed Processing Server Files

The DP Server log files in the log directory may contain information that is sensitive to your organization. Oracle recommends granting full access to this directory only to the Oracle database user running the DP Server process and any other external processing engine user.

Set NLS_LANG to UTF8

On each Server where you install the DP Server, set the computer's NLS_LANG environment variable to UTF8.

- Windows
- UNIX

Windows

Check and set your NLS_LANG environment variable:

1. Right-click the **My Computer** icon on your desktop, then click **Properties**.
2. Click the **Advanced** tab, then click **Environment Variables**.
3. In **User Variables** and **System Variables**, check if there is a variable named NLS_LANG.
4. If there is an NLS_LANG variable, highlight it and click **Edit**.
5. Set the variable value to UTF8; for example: `AMERICAN_AMERICA.UTF8`

If you do not have the NLS_LANG environment variable, change your registry settings:

1. Click **Start**, then **Run**.
2. In the Run window, enter `regedit` and click **OK**.
3. Locate one of the following registry key entries:
   - HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE
   - HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*x*

     where *x* is the unique number identifying the Oracle home
4. Add a new key named NLS_LANG with a value including UTF8; for example:

   ```
   NLS_LANG=AMERICAN_AMERICA.UTF8
   ```

UNIX

Do the following:

1. Check the environment variable NLS_LANG:

```
echo $NLS_LANG
```

2. Set the environment variable NLS_LANG to UTF8; for example:

```
% setenv NLS_LANG American_America.UTF8
```

Copy and Edit Files

This section contains the following topics:

- Copy DP Server Files
- Edit the DP Server Start Script
- Make Scripts Executable
- Copy RTF Template Files for XML Publisher
- Copy, Edit, and Grant Permissions to Execution Command Files for Processing Engines

Copy DP Server Files

1. Go to the DP Server home directory you created when you installed the DP Server code.
2. Copy the following files from $CDR_TOP/admin/template to the DP Server home directory:
   - **cdr_apps_dpserver.sh** (or cdr_apps_dpserver.cmd for running Windows services such as OBIEE)
   - **checkJSapps.sh**
   - **stopJSapps.sh**
   - **killproc.sh**

> **Note:**
>
> Do not copy **killproc.sh** if the DP server is running on a Windows computer, for example, running the Oracle Analytics Server server.

Edit the DP Server Start Script

You must add local values to **cdr_apps_dpserver.sh** (or **cdr_apps_dpserver.cmd** on Windows) before you can start the DP Server.

1. Log in as the owner of the DP Server Home Directory.
2. Go to the DP Server home directory.
3. Edit cdr_apps_dpserver.sh (or cdr_apps_dpserver.cmd). Enter the actual value for each of the following:
   - **$DPSERVER_HOME**: Enter the full path for the DP Server home on this computer.
   - **SVC**. Enter the Service Location Name (not a Service name) that you defined or will define in the Service Location subtab for the Service Location that corresponds to this computer. (For more details, see Define Service Locations and Services.) The name is case-sensitive. For example:

     ```
     SVC=SERVICE_LOCATION_NAME
     ```

> **Note:**
>
> On Windows you must enter this value at runtime.

- **JDK Location (JDK_LOC)**: Enter the full path to the JDK 1.8 executable.
  For example:

  ```
  JDK_LOC=$ORACLE_HOME/jdk/bin
  ```

- **JVM Arguments**: Add the following line immediately after the Java command (`COMMANDLINE=java`), which follows the RAC flag setting:

  ```
  -DFile.encoding=UTF8 -Duser.language=en -Duser.country=US
  ```

- **Message Receive Interval**: Enter the value for 'Message Receive Interval' as 1,200,000 milliseconds(20 minutes).
  For example:

  ```
  Q_MRI=${9-1200000}
  ```

- **Debug Level (DEBUG_LEVEL)**: Default value is "all" (which is same as pre 3.3 releases). The recommended value is "low" to generate the minimal set of log statements. Setting the value to "medium"/"all" generates verbose log statements and the size of the log files is expected to grow faster.
  For example:

  ```
  DEBUG_LEVEL=low
  ```

You can accept the default values for all other variables. Some values must be set at runtime. See Start the DP Server for details.

Make Scripts Executable

Make all the scripts executable with the following command:

```
chmod 755 *.sh
```

Copy RTF Template Files for XML Publisher

The following steps are required only on the computer where you are running XML Publisher:

1. In the DP Server home directory, create a directory called **cdrtemplates**.
2. Copy the following files from $CDR_TOP/patch/115/publisher/templates to the new **cdrtemplates** directory:
   - cdr_output_summ_cs.rtf
   - lsh-title-page.rtf
   - lsh-toc-template.rtf
   - lsh-pagenum.rtf
   - lsh-template.rtf
   - lsh-blank-page.pdf

Copy, Edit, and Grant Permissions to Execution Command Files for Processing Engines

Do the following on each computer where you have installed a processing engine:

1. From $CDR_TOP/admin/template, copy the sample execution command script for each processing engine installed on the computer. You can copy the scripts directly into the DP Server Home directory or create a subdirectory for them.

   > **Note:**
   >
   > Keep a record of the absolute location of these scripts. You will need it when you define a service location for the computer. See Define Service Locations and Services for details.

   The scripts include:

   - **cdrzip.sh** and **cdrunzip.sh**for Text Data Marts
   - **sasNormal.sh** for SAS Programs
   - **oraexp.sh** for Oracle Export Data Marts
   - **orareprunner.sh** for Oracle Reports Programs
   - **txtNormal.sh** for Text Load Sets
   - **xmlprunner.sh** for post-processing Report Sets
   - **xmlpreprunner.sh** for the Oracle LSH system reports and for cover sheets for outputs
   - **obieeinstall.cmd** for Oracle Business Intelligence Business Areas—required only on the BI Server
   - **obieedeploy.cmd** for Oracle Business Intelligence Business Areas— required only on the BI Server

2. Edit each script with information specific to the computer, for example:
   - Oracle SID
   - Location of the technology server
   - Location of Oracle setup script `coraenv`
   - Paths

   Ensure that environment variables are accessible to the DP server. For example, if the script refers to the variable $ORACLE_HOME, either define the variable or provide the full path in the script.

> **Note:**
>
> If you run SAS programs, add instructions to sasNormal.sh to start SAS in UTF8 mode. (See Start SAS in UTF8 Mode for details.) In addition, include the DP Server Home path in the environment variable as shown:
>
> ```
> PATH=$ORACLE_HOME/bin:$ORACLE_HOME/lib32: DP_Server_Home_Path:$PATH export
> PATH
> ```

3. Make all the scripts executable on the UNIX system with the following command:

   ```
   chmod 755 *.sh
   ```

Define Service Locations and Services

You must define Service Locations and Services in the Oracle LSH user interface for each computer where the Oracle LSH Distributed Processing (DP) Server will run. You define one service location for each computer, and at least one service for each engine or development environment that you want to run on that computer.

To define service locations and services you must have a user account with the Oracle LSH System Admin role assigned to it.

To log into Oracle LSH, do the following:

1. Open your web browser.
2. Enter the eBusiness Suite SSWA (Self-Service Web Application) URL as follows:

   http://<host name>.<domain name>:<HTTP port>/oa_servlets/AppsLogin

   For example:

   http://*appshost*.*your_company*.com:8000/oa_servlets/AppsLogin

   The Applications Login screen appears.
3. Enter the username and password associated with the LSH System Admin responsibility and click **Login**.
4. Under Navigator, click **Life Sciences Data Hub**. The system displays the list of Oracle LSH user interface locations to which you have access.
5. Click **Service Location**. The Service Location screen opens.

To define service locations and services in the Oracle LSH user interface, follow the instructions in "Setting Up Services" in the *Oracle Life Sciences Data Hub System Administrator's Guide*.

> **Note:**
>
> For Oracle DMW, you need one or two Service Locations. You need two services; **Text for SQL*Loader** and **SAS**. The two services can be on the same Service Location if it has access to both SQL*Loader and the SAS processing engine as well as the folders you will create to put data files into for loading into Oracle DMW. You can put text and SAS files in different locations.

Start the Distributed Processing Server

To start the Distributed Proecssing Server, do the following:

1. Log on as the owner of the Distributed Proecssing Server Home Directory.
2. Run the script by entering the following command for UNIX. Information on the parameters is given below.

   ```
   ./cdr_apps_dpserver.sh ORACLE_SID DB_HOST DB_PORT RAC_TNS RAC_FLAG FW_ENABLED
   FW_FREQ FW_POLL
   ```

   or for Windows:

   ```
   c:> cdr_apps_dpserver.cmd ORACLE_SID DB_HOST DB_PORT RAC_TNS RAC_FLAG
   FW_ENABLED FW_FREQ FW_POLL
   ```

   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 Distributed Proecssing 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.

   - *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 LSHDB adxxxsdb.example.com 20502 NA NO-RAC NO 0 0
```

where:

- `LSHDB` is the Oracle SID
- `adxxxsdb.example.com` is the host
- `20502` is the port
- You may enter NA (Not Applicable) or any other value for RAC_TNS.
- NO-RAC is the setting for RAC_FLAG
- NO indicates that File Watcher is not enabled; Oracle DMW is not being used.
- 0 FileWatcher Refresh Frequency, since File Watcher is not enabled
- 0 FileWatcher Polling Frequency, since File Watcher is not enabled

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=AP1RAC.example.com)
(PORT=1521))(ADDRESS=(PROTOCOL=tcp)(HOST=AP2RAC.example.com)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=CDRXXX)))'  RAC NO 0 0
```

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=AP1RAC.example.com)(PORT=1521))(ADDRESS=(PROTOCOL=tcp)(HOST=AP3RAC.example.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=CDRXXX)))` is the JDBC connection string of the database server
- RAC is the setting for RAC_FLAG
- NO indicates that File Watcher is not enabled; Oracle DMW is not being used.
- 0 FileWatcher Refresh Frequency, since File Watcher is not enabled
- 0 FileWatcher Polling Frequency, since File Watcher is not enabled

3. The script prompts you for a password. Enter the password for the cdr_dpserver user.
4. Check if the DP server is running:

```
./checkJSapps.sh SID
```

> **Note:**
>
> Do not change the value of DB_USER.

Start the Message Queue

1. Connect to PDB as apps user.

2. Make sure the queue is stopped. View the log:

```
select MESSAGE from cdr_msg_queues_log order by log_message_id;
```

If the most recent statement is the following, the queue is stopped.

```
End Procedure cdr_exe_msg_submission.process_queues()
```

If not, stop the queue:

```
begin cdr_exe_msg_queues_admin.stop_processing_queues; end; /
```

Wait until you see the "End Procedure" statement in the log.

3. After the queue is stopped, start and enable the queue:

```
begin
cdr_exe_msg_queues_admin.start_processing_queues;
cdr_exe_msg_queues_admin.enable_job_processing_queue;
end;
/
```

4. Check that the queue is started and enabled:

```
select MESSAGE from cdr_msg_queues_log order by log_message_id;
```

The output should contain the following statements (there may be Submission statements after these):

```
Begin Procedure cdr_exe_msg_submission.process_queues()
dequeued from control Q: _MSGCONTROL_ENABLE
```

Restart and Enable the Job Queue

Still logged in as apps:

1. Stop and disable the Job Queue:

```
begin
cdr_exe_job_queues.stop_processing_queues;
end;
/
```

2. Start and enable the job queue:

```
begin
cdr_exe_job_queues.start_jobq_process_enabled;
end;
/
```

Run the Stop Maintenance Script

After you upgrade the software, follow the steps in this procedure to stop the maintenance activity.

To run the Stop Maintenace script:

1. Download or copy the maintenance start script file from the Oracle LSH application server (EBS Middle Tier server) `$CDR_TOP/patch/115/sql/cdrmaintstop.sql` to the database server's ORACLE_HOME location or any other preferred location.
2. From the database server, log in to SQL*Plus (not SQL Developer) as the APPS database user.
3. Enter this command to execute the script:

   ```
   SQL> @cdrmaintstop.sql
   ```

4. Check the log file.

   The log file validates the success of the Stop Maintenance process and provides a maintenance ID. For example, it lists the job report (ID, user name, job duration, job type), Distributed Processing (DP) server report (location, description, and machine), and service details. If you notice that any errors with stopping the maintenance script, contact Oracle Support.

Start Server Processes

This step is required for all upgrade paths.

Start the following servers:

- Application Server
- Oracle LSH Distributed Processing (DP) Server

  If the following issue occurs during the DP server start process, add an entry to listener.ora and reload the listener (instructions are mentioned below):

  ```
  [CdrExeJSExe] Exception thrown: Error in creating JDBC Connections for
  ConnectionPool.
  Related SQL Exception: Unable to start the Universal Connection Pool:
  oracle.ucp.UniversalConnectionPoolException: Cannot get Connection from
  Datasource: java.sql.SQLRecoverableException: ORA-01034: ORACLE not
  available
  ORA-27101: shared memory realm does not exist
  Linux-x86_64 Error: 2: No such file or directory
  Additional information: 4460
  Additional information: -1725901513
  oracle.apps.cdr.dpserver.exec.server.CdrExeJSException: Error in creating
  JDBC Connections for ConnectionPool.
  at
  oracle.apps.cdr.dpserver.exec.server.CdrExeJSDataSource.getConnection(CdrEx
  eJSDataSource.java:191)
  at
  oracle.apps.cdr.dpserver.exec.server.CdrExeJSInitializer.<init>(CdrExeJSIni
  tializer.java:92)
  at
  ```

```
oracle.apps.cdr.dpserver.exec.server.CdrExeJSExe.startService(CdrExeJSExe.j
ava:263)
at
oracle.apps.cdr.dpserver.exec.server.CdrExeJSExe.main(CdrExeJSExe.java:562)
```

1. Add the following entry to listener.ora:

   ```
   USE_SID_AS_SERVICE_<Listener_Name> = ON
   ```

   where, <Listener_Name> is the listener name that is in use.
2. Reload the listener:

   ```
   lsnrctl reload
   ```

Run the Health Check Scripts

Run the Health Check scripts for Oracle LSH and Oracle DMW as described in My Oracle Support Article 2733714.1 (https://support.oracle.com).

Migrate Secure File

If you have already migrated the secure file previously, **do not** perform the instructions in this section. To check, execute the following and check the value of the SECUREFILE column. If the value is YES, that means the Secure file migration is already completed for the table.

```
select OWNER, TABLE_NAME, COLUMN_NAME,SECUREFILE
from dba_lobs
where table_name in
('CDR_INSTALLATION_LOG','CDR_OUTPUT_BLOBS','CDR_OUTPUT_CLOBS','CDR_INSTALL_SCR
IPTS','DME_DISC_CSV_FILES');
```

This section includes the following topics:

- Initiate the Secure File Migration of CDR_INSTALLATION_LOG
- Monitor the Secure File Migration of CDR_INSTALLATION_LOG
- Post Secure File Migration Task for CDR_INSTALLATION_LOG
- Initiate the Secure File Migration of CDR_OUTPUT_BLOBS
- Monitor the Secure File Migration of CDR_OUTPUT_BLOBS
- Post Secure File Migration Task for CDR_OUTPUT_BLOBS
- Initiate the Secure File Migration of CDR_OUTPUT_CLOBS
- Monitor the Secure File Migration of CDR_OUTPUT_CLOBS
- Post Secure File Migration Task for CDR_OUTPUT_CLOBS
- Initiate the Secure File Migration of CDR_INSTALL_SCRIPTS
- Monitor the Secure File Migration of CDR_INSTALL_SCRIPTS
- Post Secure File Migration Task for CDR_INSTALL_SCRIPTS
- Migrate Secure File for DME_DISC_CSV_FILES

Initiate the Secure File Migration of CDR_INSTALLATION_LOG

1. Log on to the application tier.
2. Source the environment file.
3. Navigate to the `$CDR_TOP/patch/115/sql` directory.
4. Log in to SQL*Plus as the APPS user.
5. Execute the script `cdrsfpostinstscript.sql`.

   A prompt to enter the number of threads appears.
6. Enter `4`.

   A prompt to enter the logfile pathname appears.
7. Press **Enter** to select the default logfile pathname or enter a name of your choice.
8. After the script execution is complete, check for any errors. In case of an error, contact Life Sciences Support.

The migration process starts.

Monitor the Secure File Migration of CDR_INSTALLATION_LOG

1. Make sure **at least two DBMS SCHEDULER JOBS are scheduled and running**. These jobs name starts with MIGRATE_BASICFILE_TO_SECUREFILE. Execute the following command to confirm:

```
select
    owner,
    job_name,
    JOB_ACTION,
    START_DATE,
    ENABLED,
    STATE
    from dba_SCHEDULER_JOBS
    where job_name like 'MIGRATE_BASICFILE_TO_SECUREFILE%';
```

2. Wait for the BASIC FILE to SECUREFILE migration to complete. You can monitor the migration progress by executing the following command:

```
select
    thread_id,
    Number_of_batch_to_processe,
    no_of_processed_batch,
    case when THREAD_ID in (1,2) then 'DBMS JOB SHOULD BE RUNNING'
    when Number_of_batch_to_processe > no_of_processed_batch then 'DBMS
JOB SHOULD BE RUNNING'
    else 'DBMS JOB SHOULD NOT BE RUNNING' end as status
    from
    (
    select thread_id,
    count(distinct batchid) Number_of_batch_to_processe,
    (
    select count(1)
    from CDR_INSTALLATION_LOG_SFM_TMP
```

```
where THREAD_ID=a.THREAD_ID
and status in ('COMPLETE','FAILED')
)no_of_processed_batch
from CDR_INSTALLATION_LOG_SFM_TMP  a
group by thread_id
);
```

Post Secure File Migration Task for CDR_INSTALLATION_LOG

To perform the next set of tasks, wait for at least 12 hours after the secure file migration completes. A complete application downtime is required.

1. Stop all application tier services and job queue.

   > **Note:**
   >
   > DO NOT stop the database.

   a. Stop the listener and database services.

   b. Disable all the enabled DBMS SCHEDULER JOBS.

   c. Make sure that no scheduler job is in the RUNNING state.

   d. Disable the Logon Trigger.

   e. Make sure that no application-related sessions are there in gv$session.

2. If there is any cronjob related to Oracle LSH or Oracle DMW, suspend them. Disable any custom scheduler or DBMS jobs related to Oracle LSH or Oracle DMW.

3. Note the count of the invalid objects of APPS, APPLSYS, and CDR schema by executing the following command:

   ```
   select owner,
       status,
       count(1)
       from dba_objects
       where status <> 'VALID'
       and owner in ('APPS','APPLSYS','CDR')
       AND object_name <> 'CDR_SECUREFILE_MIGRATION'
       AND object_name NOT LIKE 'CDR%SECFILE%'
       AND object_name NOT LIKE 'CDR%SFM%'
       group by owner,
       status;
   ```

4. Once all the application tier services are stopped, execute the cutover script by performing the following steps:

   a. Log in to the application tier.

   b. Source the environment file.

   c. Navigate to the `$CDR_TOP/patch/115/sql` directory.

   d. Log in to SQL*Plus as the APPS user.

   e. Execute the script `cdrsfinstallcutoff.sql`.

      A prompt to press enter to start the process appears.

f. Press **Enter**.

A prompt to enter the logfile pathname appears.

g. Press **Enter** to select the default logfile pathname or enter a name of your choice.

h. After the script execution is complete, check for any errors. In case of an error, contact Life Sciences Support.

It will take some time for the script to execute.

5. After the script `cdrsfinstallcutoff.sql` executes, make sure no DBMS SCHEDULER JOBS are running related to secure file migration.

6. Make sure the new CDR_INSTALLATION_LOG table's LOB column LOG_MESSAGE is of type SECUREFILE.

To do so, execute the following SQL command. The output of the SECUREFILE column should be YES corresponding to the CDR_INSTALLATION_LOG table.

```
select OWNER,
        TABLE_NAME,
        COLUMN_NAME,
        SEGMENT_NAME,
        TABLESPACE_NAME,
        SECUREFILE
    from dba_lobs
    where table_name like 'CDR_INSTALLATION_LOG%'
    and column_name ='LOG_MESSAGE'
    and OWNER='CDR';
```

7. If there are new INVALID objects in the APPS, APPLSYS, or CDR schema, compile those invalid objects.

```
select owner,
    status,
    count(1)
    from dba_objects
    where status <> 'VALID'
    and owner in ('APPS','APPLSYS','CDR')
    AND object_name <> 'CDR_SECUREFILE_MIGRATION'
    AND object_name NOT LIKE 'CDR%SECFILE%'
    AND object_name NOT LIKE 'CDR%SFM%'
    group by owner,
    status;
```

a. Start the UTLRP and wait for its completion.

b. Enable the Logon Trigger.

c. Start the listener/database services.

d. Enable all the DBMS SCHEDULER JOBS which were disabled at step 1.b.

8. After all validations are successful, start all the application tier services and job queue.

Initiate the Secure File Migration of CDR_OUTPUT_BLOBS

1. Log on to the application tier.

2. Source the environment file.

3. Navigate to the `$CDR_TOP/patch/115/sql` directory.
4. Log in to SQL*Plus as the APPS user.
5. Execute the script `cdrsecfilepostinstblob.sql`.

   A prompt to enter the number of threads appears.
6. Enter `4`.

   A prompt to enter the logfile pathname appears.
7. Press **Enter** to select the default logfile pathname or enter a name of your choice.
8. After the script execution is complete, check for any errors. In case of an error, contact Life Sciences Support.

The migration process starts.

Monitor the Secure File Migration of CDR_OUTPUT_BLOBS

1. Make sure **at least two DBMS SCHEDULER JOBS are scheduled and running**. These jobs name starts with MIGRATE_BASICFILE_TO_SECUREFILE. Execute the following command to confirm:

```
select
    owner,
    job_name,
    JOB_ACTION,
    START_DATE,
    ENABLED,
    STATE
    from dba_SCHEDULER_JOBS
    where job_name like 'MIGRATE_BASICFILE_TO_SECUREFILE%';
```

2. Wait for the BASIC FILE to SECUREFILE migration to complete. You can monitor the migration progress by executing the following command:

```
select
    thread_id,
    Number_of_batch_to_processe,
    no_of_processed_batch,
    case when THREAD_ID in (1,2) then 'DBMS JOB SHOULD BE RUNNING'
    when Number_of_batch_to_processe > no_of_processed_batch then 'DBMS
JOB SHOULD BE RUNNING'
    else 'DBMS JOB SHOULD NOT BE RUNNING' end as status
    from
    (
    select thread_id,
    count(distinct batchid) Number_of_batch_to_processe,
    (
    select count(1)
    from CDR_OUTPUT_BLOBS_SFM_TMP
    where THREAD_ID=a.THREAD_ID
    and status in ('COMPLETE','FAILED')
    )no_of_processed_batch
    from CDR_OUTPUT_BLOBS_SFM_TMP   a
    group by thread_id
    );
```

Post Secure File Migration Task for CDR_OUTPUT_BLOBS

To perform the next set of tasks, wait for at least 12 hours after the secure file migration completes. A complete application downtime is required.

1. Stop all application tier services and job queue.

   > **Note:**
   >
   > DO NOT stop the database.

   a. Stop the listener and database services.
   b. Disable all the enabled DBMS SCHEDULER JOBS.
   c. Make sure that no scheduler job is in the RUNNING state.
   d. Disable the Logon Trigger.
   e. Make sure that no application-related sessions are there in gv$session.

2. If there is any cronjob related to Oracle LSH or Oracle DMW, suspend them. Disable any custom scheduler or DBMS jobs related to Oracle LSH or Oracle DMW.
3. Note the count of the invalid objects of APPS, APPLSYS, and CDR schema by executing the following command:

```
select owner,
    status,
    count(1)
    from dba_objects
    where status <> 'VALID'
    and owner in ('APPS','APPLSYS','CDR')
    AND object_name <> 'CDR_SECUREFILE_MIGRATION'
    AND object_name NOT LIKE 'CDR%SECFILE%'
    AND object_name NOT LIKE 'CDR%SFM%'
    group by owner,
    status;
```

4. Once all the application tier services are stopped, execute the cutover script by performing the following steps:

   a. Log in to the application tier.
   b. Source the environment file.
   c. Navigate to the `$CDR_TOP/patch/115/sql` directory.
   d. Log in to SQL*Plus as the APPS user.
   e. Execute the script `cdrsecfileblobcutoff.sql`.

      A prompt to press enter to start the process appears.

   f. Press **Enter**.

      A prompt to enter the logfile pathname appears.

   g. Press **Enter** to select the default logfile pathname or enter a name of your choice.

h. After the script execution is complete, check for any errors. In case of an error, contact Life Sciences Support.

It will take some time for the script to execute.

5. After the script `cdrsecfileblobcutoff.sql` executes, make sure no DBMS SCHEDULER JOBS are running related to secure file migration.
6. Make sure the new CDR_OUTPUT_BLOBS table's LOB column FILE_BLOB is of type SECUREFILE.

   To do so, execute the following SQL command. The output of the SECUREFILE column should be YES corresponding to the CDR_OUTPUT_BLOBS table.

```
select OWNER,
        TABLE_NAME,
        COLUMN_NAME,
        SEGMENT_NAME,
        TABLESPACE_NAME,
        SECUREFILE
    from dba_lobs
    where table_name like 'CDR_OUTPUT_BLOBS%'
    and column_name ='FILE_BLOB'
    and OWNER='CDR';
```

7. If there are new INVALID objects in the APPS, APPLSYS, or CDR schema, compile those invalid objects.

```
select owner,
    status,
    count(1)
    from dba_objects
    where status <> 'VALID'
    and owner in ('APPS','APPLSYS','CDR')
    AND object_name <> 'CDR_SECUREFILE_MIGRATION'
    AND object_name NOT LIKE 'CDR%SECFILE%'
    AND object_name NOT LIKE 'CDR%SFM%'
    group by owner,
    status;
```

   a. Start the UTLRP and wait for its completion.
   b. Enable the Logon Trigger.
   c. Start the listener/database services.
   d. Enable all the DBMS SCHEDULER JOBS which were disabled at step 1.b.
8. After all validations are successful, start all the application tier services and job queue.

Initiate the Secure File Migration of CDR_OUTPUT_CLOBS

1. Log on to the application tier.
2. Source the environment file.
3. Navigate to the `$CDR_TOP/patch/115/sql` directory.
4. Log in to SQL*Plus as the APPS user.
5. Execute the script `cdrsecfilepostinstclob.sql`.

A prompt to enter the number of threads appears.

6. Enter `4`.

   A prompt to enter the logfile pathname appears.

7. Press **Enter** to select the default logfile pathname or enter a name of your choice.

8. After the script execution is complete, check for any errors. In case of an error, contact Life Sciences Support.

The migration process starts.

Monitor the Secure File Migration of CDR_OUTPUT_CLOBS

1. Make sure **at least two DBMS SCHEDULER JOBS are scheduled and running**. These jobs name starts with MIGRATE_BASICFILE_TO_SECUREFILE_CLOB. Execute the following command to confirm:

```
select
owner,
job_name,
JOB_ACTION,
START_DATE,
ENABLED,
STATE
from dba_SCHEDULER_JOBS
where job_name like 'MIGRATE_BASICFILE_TO_SECUREFILE_CLOB%';
```

2. Wait for the BASIC FILE to SECUREFILE migration to complete. You can monitor the migration progress by executing the following command:

```
select
        thread_id,
        Number_of_batch_to_processe,
        no_of_processed_batch,
        case when THREAD_ID in (1,2) then 'DBMS JOB SHOULD BE RUNNING'
        when Number_of_batch_to_processe > no_of_processed_batch then
'DBMS        JOB SHOULD BE RUNNING'
        else 'DBMS JOB SHOULD NOT BE RUNNING' end as status
        from
        (
        select thread_id,
        count(distinct batchid) Number_of_batch_to_processe,
        (
        select count(1)
        from CDR_OUTPUT_CLOBS_SFM_TMP
        where THREAD_ID=a.THREAD_ID
        and status in ('COMPLETE','FAILED')
        )no_of_processed_batch
        from CDR_OUTPUT_CLOBS_SFM_TMP   a
        group by thread_id
        );
```

Post Secure File Migration Task for CDR_OUTPUT_CLOBS

To perform the next set of tasks, wait for at least 12 hours after the secure file migration completes. A complete application downtime is required.

1. Stop all application tier services and job queue.

   > **Note:**
   >
   > DO NOT stop the database.

   a. Stop the listener and database services.
   b. Disable all the enabled DBMS SCHEDULER JOBS.
   c. Make sure that no scheduler job is in the RUNNING state.
   d. Disable the Logon Trigger.
   e. Make sure that no application-related sessions are there in gv$session.

2. If there is any cronjob related to Oracle LSH or Oracle DMW, suspend them. Disable any custom scheduler or DBMS jobs related to Oracle LSH or Oracle DMW.
3. Note the count of the invalid objects of APPS, APPLSYS, and CDR schema by executing the following command:

   ```
   select owner,
       status,
       count(1)
       from dba_objects
       where status <> 'VALID'
       and owner in ('APPS','APPLSYS','CDR')
       AND object_name <> 'CDR_SECUREFILE_MIGRATION'
       AND object_name NOT LIKE 'CDR%SECFILE%'
       AND object_name NOT LIKE 'CDR%SFM%'
       group by owner,
       status;
   ```

   Or, create a backup table with the list of all INVALID OBJECTS by executing the following command:

   ```
   CREATE TABLE <TABLE_NAME> AS SELECT * FROM DBA_OBJECTS WHERE STATUS
   <>'VALID';
   ```

4. Once all the application tier services are stopped, execute the cutover script by performing the following steps:

   a. Log in to the application tier.
   b. Source the environment file.
   c. Navigate to the `$CDR_TOP/patch/115/sql` directory.
   d. Log in to SQL*Plus as the APPS user.
   e. Execute the script `cdrsecfileclobcutoff.sql`.

      A prompt to press enter to start the process appears.

f. Press **Enter**.

A prompt to enter the logfile pathname appears.

g. Press **Enter** to select the default logfile pathname or enter a name of your choice.

h. After the script execution is complete, check for any errors. In case of an error, contact Life Sciences Support.

It will take some time for the script to execute.

5. After the script `cdrsecfileclobcutoff.sql` executes, make sure no DBMS SCHEDULER JOBS are running related to secure file migration.

6. Make sure the new CDR_OUTPUT_CLOBS table's LOB column FILE_CLOB is of type SECUREFILE.

To do so, execute the following SQL command. The output of the SECUREFILE column should be YES corresponding to the CDR_OUTPUT_CLOBS table.

```
select OWNER,
        TABLE_NAME,
        COLUMN_NAME,
        SEGMENT_NAME,
        TABLESPACE_NAME,
        SECUREFILE
    from dba_lobs
    where table_name like 'CDR_OUTPUT_CLOBS%'
    and column_name ='FILE_CLOB'
    and OWNER='CDR';
```

7. If there are new INVALID objects in the APPS, APPLSYS, or CDR schema, compile those invalid objects.

```
select owner,
    status,
    count(1)
    from dba_objects
    where status <> 'VALID'
    and owner in ('APPS','APPLSYS','CDR')
    AND object_name <> 'CDR_SECUREFILE_MIGRATION'
    AND object_name NOT LIKE 'CDR%SECFILE%'
    AND object_name NOT LIKE 'CDR%SFM%'
    group by owner,
    status;
```

a. Start the UTLRP and wait for its completion.

b. Enable the Logon Trigger.

c. Start the listener/database services.

d. Enable all the DBMS SCHEDULER JOBS which were disabled at step 1.b.

8. After all validations are successful, start all the application tier services and job queue.

Initiate the Secure File Migration of CDR_INSTALL_SCRIPTS

1. Log on to the application tier.

2. Source the environment file.

3. Navigate to the `$CDR_TOP/patch/115/sql` directory.
4. Log in to SQL*Plus as the APPS user.
5. Execute the script `cdrsecfilepostinstscrpt.sql`.

   A prompt to enter the number of threads appears.
6. Enter `4`.

   A prompt to enter the logfile pathname appears.
7. Press **Enter** to select the default logfile pathname or enter a name of your choice.
8. After the script execution is complete, check for any errors. In case of an error, contact Life Sciences Support.

Monitor the Secure File Migration of CDR_INSTALL_SCRIPTS

1. Make sure **at least two DBMS SCHEDULER JOBS are scheduled and running**. These jobs name starts with MIGRATE_BASICFILE_TO_SECUREFILE_SCRPT. Execute the following command to confirm:

```
select
    owner,
    job_name,
    JOB_ACTION,
    START_DATE,
    ENABLED,
    STATE
    from dba_SCHEDULER_JOBS
    where job_name like 'MIGRATE_BASICFILE_TO_SECUREFILE_SCRPT%';
```

2. Wait for the BASIC FILE to SECUREFILE migration to complete. You can monitor the migration progress by executing the following command:

```
select
        thread_id,
        Number_of_batch_to_processe,
        no_of_processed_batch,
        case when THREAD_ID in (1,2) then 'DBMS JOB SHOULD BE RUNNING'
        when Number_of_batch_to_processe > no_of_processed_batch then
'DBMS JOB SHOULD BE RUNNING'
        else 'DBMS JOB SHOULD NOT BE RUNNING' end as status
        from
        (
        select thread_id,
        count(distinct batchid) Number_of_batch_to_processe,
        (
        select count(1)
        from CDR_INSTALL_SCRIPTS_SFM_TMP
        where THREAD_ID=a.THREAD_ID
        and status in ('COMPLETE','FAILED')
        )no_of_processed_batch
        from CDR_INSTALL_SCRIPTS_SFM_TMP   a
        group by thread_id
        );
```

Post Secure File Migration Task for CDR_INSTALL_SCRIPTS

To perform the next set of tasks, wait for at least 12 hours after the secure file migration completes. A complete application downtime is required.

1. Stop all application tier services and job queue.

   > **Note:**
   >
   > DO NOT stop the database.

   a. Stop the listener and database services.
   b. Disable all the enabled DBMS SCHEDULER JOBS.
   c. Make sure that no scheduler job is in the RUNNING state.
   d. Disable the Logon Trigger.
   e. Make sure that no application-related sessions are there in gv$session.

2. If there is any cronjob related to Oracle LSH or Oracle DMW, suspend them. Disable any custom scheduler or DBMS jobs related to Oracle LSH or Oracle DMW.
3. Note the count of the invalid objects of APPS, APPLSYS, and CDR schema by executing the following command:

   ```
   select owner,
       status,
       count(1)
       from dba_objects
       where status <> 'VALID'
       and owner in ('APPS','APPLSYS','CDR')
       AND object_name <> 'CDR_SECUREFILE_MIGRATION'
       AND object_name NOT LIKE 'CDR%SECFILE%'
       AND object_name NOT LIKE 'CDR%SFM%'
       group by owner,
       status;
   ```

   Or, create a backup table with the list of all INVALID OBJECTS by executing the following command:

   ```
   CREATE TABLE <TABLE_NAME> AS SELECT * FROM DBA_OBJECTS WHERE STATUS
   <>'VALID';
   ```

4. Once all the application tier services are stopped, execute the cutover script by performing the following steps:

   a. Log in to the application tier.
   b. Source the environment file.
   c. Navigate to the `$CDR_TOP/patch/115/sql` directory.
   d. Log in to SQL*Plus as the APPS user.
   e. Execute the script `cdrsecfilescrptcutoff.sql`.

      A prompt to press enter to start the process appears.

f. Press **Enter**.

A prompt to enter the logfile pathname appears.

g. Press **Enter** to select the default logfile pathname or enter a name of your choice.

h. After the script execution is complete, check for any errors. In case of an error, contact Life Sciences Support.

It will take some time for the script to execute.

5. After the script `cdrsecfilescrptcutoff.sql` executes, make sure no DBMS SCHEDULER JOBS are running related to secure file migration.

6. Make sure the new CDR_INSTALL_SCRIPTS table's LOB column SCRIPT is of type SECUREFILE.

To do so, execute the following SQL command. The output of the SECUREFILE column should be YES corresponding to the CDR_INSTALL_SCRIPTS table.

```
select OWNER,
        TABLE_NAME,
        COLUMN_NAME,
        TABLESPACE_NAME,
        SECUREFILE
    from dba_lobs
    where table_name like 'CDR_INSTALL_SCRIPTS%'
    and column_name ='SCRIPT'
    and OWNER='CDR';
```

7. If there are new INVALID objects in the APPS, APPLSYS, or CDR schema, compile those invalid objects.

```
select owner,
    status,
    count(1)
    from dba_objects
    where status <> 'VALID'
    and owner in ('APPS','APPLSYS','CDR')
    AND object_name <> 'CDR_SECUREFILE_MIGRATION'
    AND object_name NOT LIKE 'CDR%SECFILE%'
    AND object_name NOT LIKE 'CDR%SFM%'
    group by owner,
    status;
```

a. Start the UTLRP and wait for its completion.

b. Enable the Logon Trigger.

c. Start the listener/database services.

d. Enable all the DBMS SCHEDULER JOBS which were disabled at step 1.b.

8. After all validations are successful, start all the application tier services and job queue.

Migrate Secure File for DME_DISC_CSV_FILES

To perform the next set of tasks, wait for at least 12 hours after the secure file migration completes. A complete application downtime is required.

1. Stop all application tier services and job queue.

> **Note:**
>
> DO NOT stop the database.

2. If there is any cronjob related to Oracle LSH or Oracle DMW, suspend them. Disable any custom scheduler or DBMS jobs related to Oracle LSH or Oracle DMW.
3. Note the count of the invalid objects of APPS, APPLSYS, and CDR schema by executing the following command:

```
select owner,
            status,
            count(1)
            from dba_objects
            where status <> 'VALID'
            and owner in ('APPS','APPLSYS','CDR')
            AND object_name <> 'CDR_SECUREFILE_MIGRATION'
            AND object_name NOT LIKE 'CDR%SECFILE%'
            AND object_name NOT LIKE 'CDR%SFM%'
            group by owner,
            status;
```

   Or, create a backup table with the list of all INVALID OBJECTS by executing the following command:

```
CREATE TABLE <TABLE_NAME> AS SELECT * FROM DBA_OBJECTS WHERE STATUS
<>'VALID';
```

4. Once all the application tier services are stopped, execute the cutover script by performing the following steps:
   a. Log in to the application tier.
   b. Source the environment file.
   c. Navigate to the `$CDR_TOP/patch/115/sql` directory.
   d. Log in to SQL*Plus as the APPS user.
   e. Execute the script `cdrsecfilecsvcutoff.sql`.

      A prompt to press enter to start the process appears.
   f. Press **Enter**.

      A prompt to enter the logfile pathname appears.
   g. Press **Enter** to select the default logfile pathname or enter a name of your choice.
   h. After the script execution is complete, check for any errors. In case of an error, contact Life Sciences Support.

   It will take some time for the script to execute.
5. After the script `cdrsecfilecsvcutoff.sql` executes, make sure no DBMS SCHEDULER JOBS are running related to secure file migration.
6. Make sure the new DME_DISC_CSV_FILES table's LOB column CONTENT is of type SECUREFILE.

To do so, execute the following SQL command. The output of the SECUREFILE column should be YES corresponding to the DME_DISC_CSV_FILES table.

```
select OWNER,
              TABLE_NAME,
              COLUMN_NAME,
              TABLESPACE_NAME,
              SECUREFILE
        from dba_lobs
        where table_name like 'DME_DISC_CSV_FILES%'
        and column_name =CONTENT'
        and OWNER='CDR';
```

7. If there are new INVALID objects in the APPS, APPLSYS, or CDR schema, compile those invalid objects.

```
select owner,
            status,
            count(1)
            from dba_objects
            where status <> 'VALID'
            and owner in ('APPS','APPLSYS','CDR')
            AND object_name <> 'CDR_SECUREFILE_MIGRATION'
            AND object_name NOT LIKE 'CDR%SECFILE%'
            AND object_name NOT LIKE 'CDR%SFM%'
            group by owner,
            status;
```

8. After all validations are successful, start all the application tier services and job queue.

10
What's Next

After you have finished all the installation tasks outlined in this book, you must do the following tasks before you can begin to use the Oracle LSH:

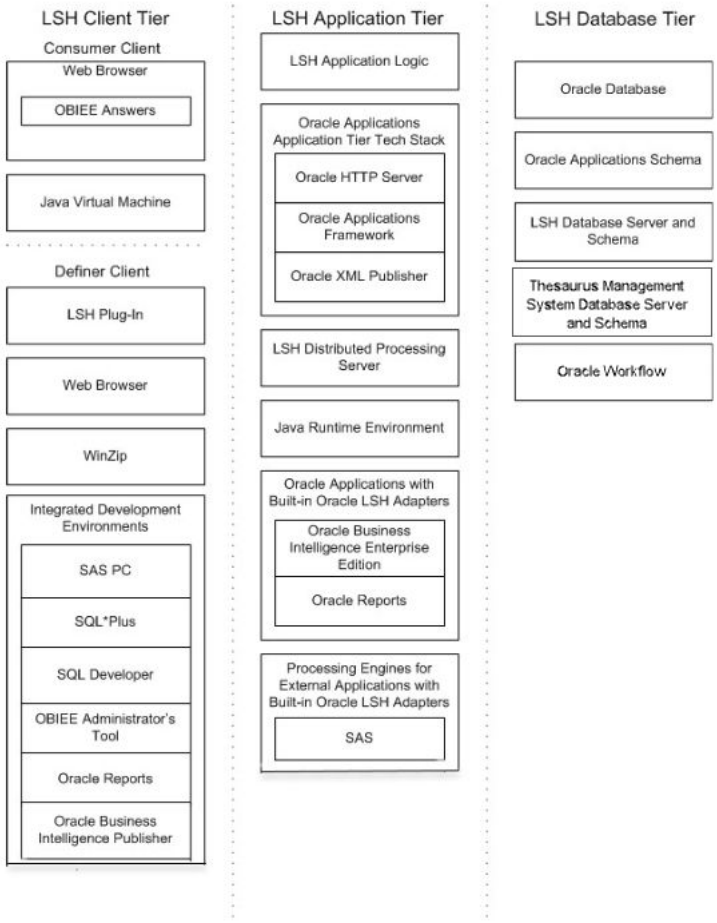
> **Note:**
>
> Oracle DMW users can skip this section and proceed to the *Oracle Life Sciences Data Management Workbench Installation Guide*.

- Define Oracle LSH Distributed Processing (DP) Server service locations and services to integrateOracle LSH with its processing engines and IDEs. See "Setting Up Services" in the *Oracle Life Sciences Data Hub Administrator's Guide*.
- Design an object security system, classification system, and organizational structure for your implementation of Oracle LSH. See "Designing a Security System," "Designing a Classification System," and "Designing an Organizational Structure" in the *Oracle Life Sciences Data Hub Implementation Guide*.
- Set up the security system, see "Setting Up the Security System" in the *Oracle Life Sciences Warehouse Security Guide*.
- Set up the classification system and organizational domains. See "Setting Up the Classification System" in the *Oracle Life Sciences Data Hub System Administrator's Guide* and "Applications User Interface" in *Oracle Life Sciences Data Hub Application Developer's Guide*.
- Define remote locations to integrate Oracle LSH with Oracle-based source data systems. See "Registering Locations and Connections" in the *Oracle Life Sciences Data Hub System Administrator's Guide*.
- If you plan to use OBIEE for visualizations or Oracle BIP for reports, do the additional setup required for each system. See the *Oracle Life Sciences Data Hub System Administrator's Guide* for information.
- If you plan to integrate with Oracle Identity Cloud Services (IDCS) for the single sign-on solution, follow instructions in *Oracle Life Sciences Data Hub and Oracle Life Sciences Data Management Workbench SSO Configuration with IDCS* (Doc ID 2862928.1) on My Oracle Support.

A

Architecture Overview

The following figure shows the logical Oracle Life Sciences Data Hub architecture described in this section.



This section contains the following topics:

- Client Tier
- Application Tier
- Database Tier
- Adapters to External Systems

Client Tier

There are two ways to set up a client, depending on the type of user:

Consumers and Administrators: Oracle Life Sciences Data Hub (Oracle LSH) Consumers, who retrieve information in the form of reports and visualizations, and Oracle LSH Administrators, who perform administrative tasks within Oracle LSH, require the following on their personal computers:

- A Web browser
- Java Virtual Machine (JVM)

Administrators require either JInitiator or JVM to use any of the Oracle Forms screens related to security, to run the post-installation jobs, and to set up user accounts and functional roles.

Definers: A full-service client for users who create (define) the programs that operate on Oracle LSH data and generate reports requires additional software:

- Oracle LSH plug-in for launching Integrated Development Environments (IDEs)
- WinZip
- Web browser
- In addition, Definers need one or more IDE clients. These may include: SAS PC, SQL*Plus, SQL Developer, Oracle BI Administration Tool, Oracle Reports, and Oracle Business Intelligence Publisher.

Application Tier

In addition to standard Oracle Applications components, the Oracle Life Sciences Data Hub application tier includes the following:

Oracle LSH Application Server: The Oracle LSH Application Server renders the user interface using the Oracle Applications Framework and handles the communication between the user interface and the database using the Java Runtime Environment.

Oracle LSH Distributed Processing (DP) Server: The Oracle LSH DP Server handles the communication between Oracle LSH and the external processing systems required to support the IDEs.

Processing Systems

XML Publisher is required for internal Oracle LSH processing. The other systems are required only if you are developing Oracle LSH Programs in those technologies:

- **Oracle XML Publisher** is used by Oracle LSH to run system reports. Oracle LSH also uses XML Publisher to create Report Sets as a single PDF output with a unified table of contents and custom templates.
- **Oracle Reports** executes user-developed Oracle Reports Programs.
- **Oracle Business Intelligence Publisher** executes user-developed Oracle BIP Programs.
- **SAS** executes user-developed SAS Programs.
- **Oracle Discoverer Plus**. Accessed by Consumer clients through a Web browser, this application generates data visualizations based on user-developed Oracle LSH Discoverer Business Areas.

Database Tier

The Oracle Life Sciences Data Hub database tier includes the following:

- **Oracle Enterprise Edition RDBMS**. All of the Oracle LSH database tier components use the Oracle Enterprise Edition database server and database.
- **Oracle Applications Schema**. The Oracle Applications Schema is the schema installed as part of the Oracle Applications installation.
- **Oracle Workflow**. Oracle LSH uses Oracle Workflow to allow users to create and execute Workflow programs. Oracle Workflow is installed with Oracle Applications.
- **Oracle Thesaurus Management System** (TMS). Oracle LSH uses the TMS database tier internally to run its classification system, which is a required part of Oracle LSH functionality.

Oracle Enterprise Edition RDBMS

All of the Oracle LSH database tier components use the Oracle Enterprise Edition database server and database.

Oracle Applications Schema

The Oracle Applications Schema is the schema installed as part of the Oracle Applications installation. It contains the Oracle LSH schema.

Oracle LSH Database Server and Schema

These include the Oracle LSH business logic in PL/SQL packages, internal Oracle LSH tables and views, security, adapters, and APIs; as well as Oracle LSH user-developed metadata tables and packages.

Oracle Workflow

Oracle LSH uses Oracle Workflow to allow users to create and execute Oracle LSH Workflow Programs. Oracle Workflow is installed with Oracle Applications.

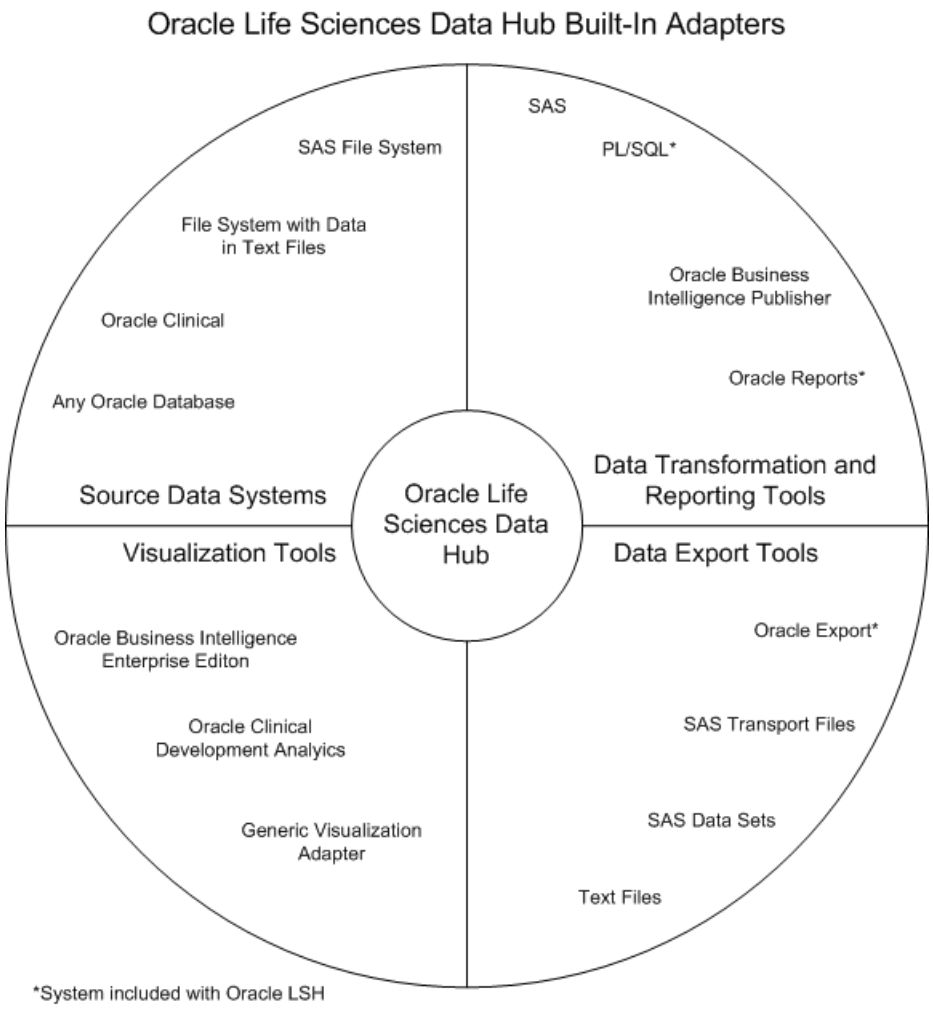
Oracle Thesaurus Management System (TMS)

The Oracle LSH classification system is based on TMS.

Adapters to External Systems

Oracle Life Sciences Data Hub (Oracle LSH) is intended for integration with other systems for a variety of purposes. Each external system integrated with Oracle LSH requires an adapter to handle whatever communication and exchange is required, depending on the purpose of the integration and the technical specifications of the external system. Oracle LSH is shipped with built-in adapters for loading data into Oracle LSH from other systems, for developing and executing programs operating on Oracle LSH data, and for exporting data out of Oracle LSH.

#GUID-F406D70F-BB51-42EE-B6DF-5AA0452F42C2/CBHGAICE shows the adapters that are included with Oracle LSH. Adapters to other systems may be available from third parties.



This section contains the following topics:

- Source Data Systems
- Data Transformation and Reporting Tools
- Visualization Tools
- Data Export Tools

Source Data Systems

Oracle Life Sciences Data Hub (Oracle LSH) includes adapters to external systems that you can use to load data into Oracle LSH:

- **SAS**. The SAS adapter allows you to load SAS data sets into Oracle LSH.
- **Text**. The Text adapter allows you to load text files from any system into Oracle LSH.
- **Oracle Databases**. The general Oracle Databases adapter allows you to load data from any Oracle database into Oracle LSH.
- **Oracle Clinical**. The Oracle Clinical adapter family includes eight specialized adapters for loading the following data and metadata from Oracle Clinical:
  - Data Extract SAS Views

  - Data Extract Oracle Views
  - Global Library
  - Labs
  - Study Data
  - Study Design and Definition
  - Stable Interface Tables
  - Randomization

Data Transformation and Reporting Tools

Oracle LSH includes adapters to set up the following systems as integrated development environments (IDEs) for developing and generating programs:

- SAS
- PL/SQL
- Oracle Reports
- Oracle Business Intelligence Publisher

Visualization Tools

Oracle LSH includes adapters to allow the following systems to display visualizations of Oracle LSH data:

- Oracle Discoverer
- Oracle Business Intelligence Enterprise Edition (OBIEE) Answers
- Oracle Clinical Development Analytics—to view visualizations in OBIEE Answers of Oracle Clinical data in Oracle LSH
- Generic Visualization Adapter—to integrate other visualization tools

Data Export Tools

Oracle LSH includes adapters to allow exporting Oracle LSH data:

- Oracle Export
- SAS—Transport Files and Data Sets
- Text Files