Oracle Life Sciences Data Hub

Installation Guide

Release 3.5
G49644-01
February 2026

ORACLE

Oracle Life Sciences Data Hub Installation Guide, Release 3.5

G49644-01

Contents

4 Upgrade to Oracle LSH 3.5

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

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 KB867359 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 KB161006 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 3.5** 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 3.5.0.0.0** and click **Add to Cart**.
4. Click **Checkout**. You see a list of the selected software:
   - Oracle Life Sciences Data Management Workbench 3.5.0.0.0 (Oracle Standard Terms and Conditions)
   - Oracle Life Sciences Data Hub 3.5.0.0.0
   - Oracle Life Sciences Data Management Workbench 3.5.0.0.0
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*> 3.5 release and Oracle Life Sciences Data Management Workbench <*your operating system*> 3.5 release:
   - Oracle Life Sciences Data Hub 3.5
   - Oracle Life Sciences Data Management Workbench 3.5
8. Leave the list of zipped files selected to download the package of Oracle Life Sciences Data Management Workbench 3.5.0.0.0 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 (**p37944396_35000_Generic.zip**) and Oracle LSH (**p37944438_R12_GENERIC.zip**).
12. See #unique_23 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.

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 3.5	Media Pack	37944438
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 Document ID KA1239 on My Oracle Support	My Oracle Support	KB858280 and KA1237

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 14.1.2 and Coherence for Linux x86	Media Pack	p30188255_122140_Generic.zip
ADF patch	My Oracle Support	38015961
Oracle Life Sciences Data Management Workbench 3.5	Media Pack	37944396

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

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

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

- *Oracle Life Sciences Applications Supported Technology Stacks*, Document ID KB161006

Upgrading to Oracle LSH 3.5

Upgrade to Oracle LSH 3.5 references the following My Oracle Support articles:

- *Steps For Configuration And Monitoring CLEANUP_NON_EXIST_OBJ Jobs*, Document ID KB464238
- *How To Apply Patches for LSH/DMW in Downtime Mode*, Document ID KB382785

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 KB604911)

2

System Requirements and Technology Stack

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 KB161006 on My Oracle Support.

This section includes the following 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 KB161006 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 KB161006 on My Oracle Support.

Linux x86-64 (64-Bit):

- Oracle Enterprise Linux 7 or 8
- Red Hat Enterprise 7.x or 8.x

Clients

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

- Google Chrome: Tested using version 143.0.7499.193 (Official Build) (64-bit).

  Version 109.0.5414.120 (Official Build) (64-bit) and later are supported.
- Mozilla Firefox: Tested using version 146.0.1 (64-bit).

  Version 109.0.1 (64-bit) and later are supported.
- Microsoft Edge Chromium: Tested using version 143.0.3650.139 (Official Build) (64-bit).

  Version 111.0.5500.0 (Official Build) (64-bit) and later are supported.

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 KB161006 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_461 or later 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_461 or later 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

Upgrade Oracle TMS Database to Release 5.4.1

Oracle LSH uses Oracle Thesaurus Management System (Oracle 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 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.

If you are using an Oracle TMS release prior to 5.4.1, you must upgrade Oracle TMS database components required to support the Oracle LSH classification system to release 5.4.1.

To upgrade to Oracle TMS release 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.

4

Upgrade to Oracle LSH 3.5

To upgrade from Oracle LSH release 3.4.x to Oracle LSH release 3.5, follow instructions in 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 maintenance script:

1. Download or copy the start maintenance 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* 14c (14.1.2.0.0) at https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-server/14.1.2/start/overview.html.

   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 3.5

You must upgrade to Oracle LSH 3.5 as a patch to Oracle Applications. The patch **37944438** 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 KB382785 and KB464238 respectively on My Oracle Support. Contact Life Sciences Support to get these documents.

1. Locate p37944438_R12_GENERIC.zip in the staging area.
2. Unzip p37944438_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 KB382785 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/dme35ddlscript.sql
   ```

5. Log in to the database as APPS and execute the following script:

   > **Note**
   >
   > Execute the following script only if you are upgrading from release 3.4.0.1, 3.4.1, or 3.4.2 to release 3.5.

   ```
   $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.
6. Set up the cleanup job. For information on how to set up the cleanup job, see *Steps For Configuration And Monitoring CLEANUP_NON_EXIST_OBJ Jobs*, Document ID KB464238 on My Oracle Support.

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
- Load the Adapter Files
- Run the Post-Installation Concurrent Program
- Grant Security Rights to Seeded Adapters
- Remove EBS Audit and Create New Auditing Process
- 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 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 KB251419 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 KB724389 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.x to 3.5.

> **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.

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;
/
```

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.

- Grant DME_ERR_LOG to the workarea schemas:
  1. Copy the file $CDR_TOP/patch/115/sql/dmeCrDmeErrTabLSH.sql to a download directory.
  2. Log in to SQL*Plus as the APPS database user.
  3. Execute the script from the download directory:

     ```
     <download_directory>/dmeCrDmeErrTabLSH.sql
     ```

     The script creates the local DME_ERR_LOG table in all the Oracle LSH work areas that are not connected to any Oracle DMW study.

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*.

> **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.

> **Note**
>
> You would have already setup the DP Server during your fresh installation of the 3.3.x or 3.4.x release. However, ensure that the following tasks had been completed.

- 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_461 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

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.

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 Distributed Processing 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 the following lines to sasNormal.sh to start SAS in UTF8 mode:
>
> - 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.)
>
> 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 Processing Server, do the following:

1. Log on as the owner of the Distributed Processing 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 Processing 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 maintenance script:

1. Download or copy the stop maintenance 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 and as described in *Configuration Health Check Scripts for LSH and DMW 3.x Installations*, Document ID KB653762 on My Oracle Support.

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_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.

5
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:

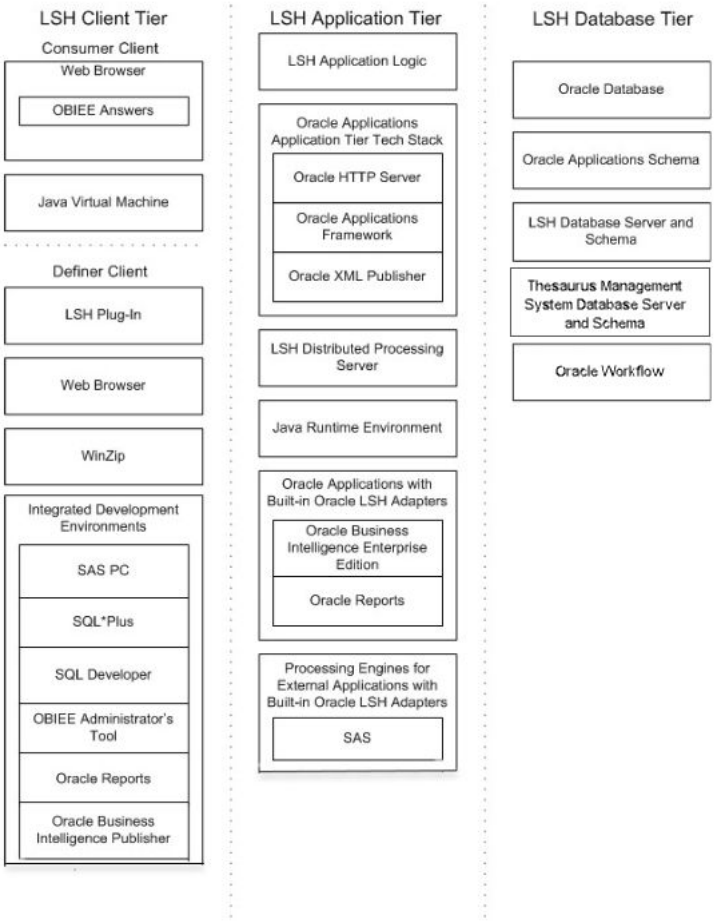
> **Note**
>
> Oracle DMW users can skip this section and proceed to the *Oracle Life Sciences Data Management Workbench Installation Guide*.

- Define Oracle LSH Distributed Processing (DP) Server service locations and services to integrate Oracle 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*, Document ID KB457314 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.

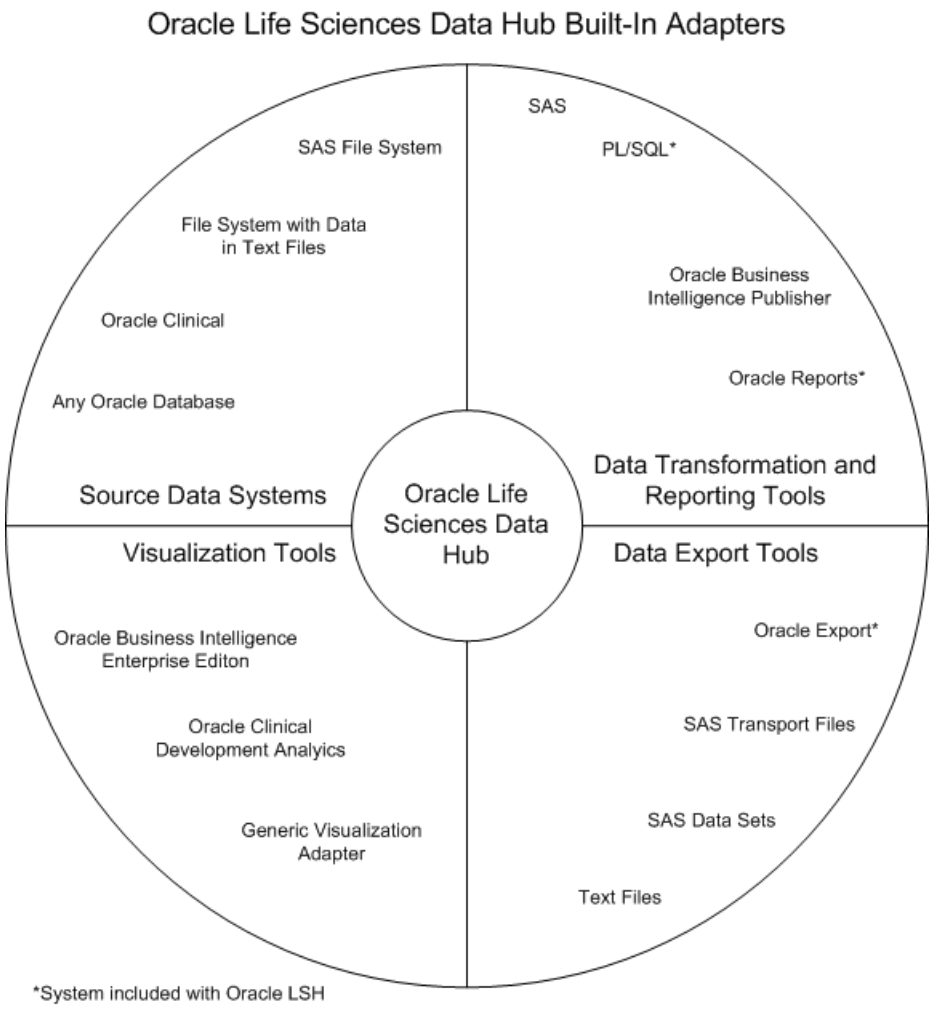
Oracle Thesaurus Management System (TMS)

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.

#unique_96/unique_96_Connect_42_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