Oracle® Healthcare Translational Research

Administrator's Guide

Release 3.4

F33198-01

February 2021

Oracle Healthcare Translational Research Administrator's Guide, Release 3.4

F33198-01

Contents

Preface

Welcome to your updated version of the Oracle Healthcare Translational Research Administrator's Guide! This document will guide you through your tasks as an administrator in OHTR.

- Documentation Accessibility
- Find more information
- Related Documents

Documentation Accessibility

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

Access to Oracle Support

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

Find more information

Oracle Help Center

The latest user documentation for Oracle Health Sciences products is available at `https://docs.oracle.com/en/industries/health-sciences/`.

My Oracle Support

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

Related Documents

Oracle Business Intelligence Enterprise Edition Documentation

OBIEE documentation is available at `https://docs.oracle.com/middleware/12212/biee/index.html`.

Oracle Healthcare Foundation

OHF documentation is available at `https://docs.oracle.com/en/industries/health-sciences/oracle-healthcare-foundation/index.html`

1

Create and manage user accounts and user groups

In this chapter you will learn how to:

- Configure user accounts
- Set up the logging level for Cohort Explorer
- Manage roles and permissions
- Administrate user groups
- Set limits for exporting variants

Configure user accounts

User accounts can be created in three ways:

- Through Oracle Access Manager
- Through an Oracle WebLogic Server
- Through the Oracle Database

Through Oracle Access Manager

In Oracle Access Manager (OAM). If Oracle Access Manager is configured, the user can set his/ her credentials:

- The user can log into the Oracle Healthcare Translational Research application by using the single sign-on interface which is shared among multiple applications. For example, with an Oracle Business Intelligence Enterprise Edition (OBIEE) full license, the same credentials can be used for generating Oracle Business Intelligence Enterprise Edition reports.
- The user creates the password, which is not visible to an administrator.
- After a configurable number of unsuccessful login attempts, the user is locked out.
- After a configurable amount of inactive time, the login session times out.

> **Note:**
>
> Roles are automatically setup as described in Manage roles and permissions.

Through an Oracle WebLogic Server

If Oracle Access Manager is not configured, the identified roles must be manually set up in a WebLogic instance.

For more information, see Create Users and Add Users to Groups in the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help* available at

https://docs.oracle.com/middleware/1221/wls/WLACH/taskhelp/security/ManageUsersAndGroups.html

Through the Oracle Database

For users who want to access the backend database of Oracle Healthcare Translational Research, the **COHORTDATAMARTUSER** role is granted, giving data access only through views, not tables and preventing PII access except through explicit assignments.

Set up the logging level for Cohort Explorer

To set the logging level of TRC modules for tracking users and activities within the application:

1. Execute wlst.sh (or wlst.cmd) in `MW_HOME/oracle_common/common/bin`, where

   `MW_HOME` is the home directory of Oracle Middleware.

2. Connect to the Oracle WebLogic Server.

   ```
   connect('username','password','t3://hostname:port_number')
   ```

3. Run in domainRuntime tree, which is required for WLST logging commands.

   ```
   domainRuntime()
   ```

4. Set the logging level of oracle.hsgbu.trc with the following command:

   ```
   setLogLevel(target='SERVER_NAME', logger='oracle.hsgbu.trc',
   level='INFO', addLogger=1)
   ```

   Where SERVER_NAME is the name of the target server where TrcApp is deployed.

5. Run the following command to verify the logging level:

   ```
   getLogLevel(target='SERVER_NAME', logger='oracle.hsgbu.trc') The
   following should be displayed: NOTIFICATION:1
   ```

   You can update the level to the desired Oracle WebLogic Server logging severity levels mentioned here: https://docs.oracle.com/middleware/12213/wls/WLLOG/logging_services.htm#WLLOG116

This log will be available at the following location <MW_HOME>/user_projects/domains/oh_domain/servers/<managed_server_name>/logs

File name of the log file will be: <managed_server_name>-diagnostic.log

Manage roles and permissions

The Oracle Healthcare Translational Research user is assigned one or more of the following roles:

> **Note:**
>
> Users belonging only to the following groups have limited functionality based on the assigned roles:
>
> - trc-comics-limited-user-group
> - trc-limited-user-group
> - trc-basic-user-group

Role	Permissions	Accessible Screens
trc-bioinformatician-group	• Can download files that have a link stored in Oracle Healthcare Omics (OHO, formerly known as Oracle Data Banks) and can be located in the middle tier in an accessible location. • Cannot build new reports or modify the content of existing reports.	• Cohort Query Tab • Cohort Query Tab: Genomic Data (tab in accordion) • Query Tab: Relative Time Events: gene variant (radio button) • Cohort Viewer (top tab) • Cohort Viewer: Cohort List • Cohort Viewer: Cohort Timeline • Cohort Reports • Cohort Viewer: Genomic Data Export • Single Patient Viewer: View Record • Single Patient Viewer: View Record: Genomic Data Collected • Circular Genomic Viewer (Visquick) • Genomic Query • Genomic Query: Columns after searching for gene/variant coming from CDM (Patient Count, Specimen Anatomical Site columns) • My Workspace: My Recent Queries • My Cohort Lists • Queries or Lists shared with me • Queries or Lists shared with All • My Queries • Gene Sets • Manage Gene Sets • Jobs
trc-cohort-group	• Can query any data from CDM but cannot query OHO directly. • Can export the clinical data in a supported format and view the Dashboard. • Cannot build new reports or modify the content of existing reports.	• Cohort Query Tab • Cohort Query Tab: Relative Time Events: gene variant (radio button) • Cohort Viewer (top tab) • Cohort Viewer: Cohort List • Cohort Viewer: Cohort Timeline • Single Patient Viewer: View Record • My Workspace: My Recent Queries • Short Cuts • My Cohort Lists • Queries or Lists shared with me • Queries or Lists shared with All • My Queries
trc-limited-user-group	• Can view the patient count on the Query Patients page • Cannot build new reports or modify the content of existing reports.	• Cohort Query Tab (Save Query button is not accessible) • My Workspace: My Recent Queries

Role	Permissions	Accessible Screens
trc-omics-group	• Can query and read data from OHO • Can export omics data in file formats readable by genome viewers. For example, VCF, SEG, RES formats for IGV. • Cannot build new reports or modify the content of existing reports.	• Genomic Query • My Workspace: My Recent Queries • Short Cuts • Gene Sets • Jobs
trc-comics-limited-user-group	• Can query data from CDM and OHO using Query Patients interface • Can only access Patient Genomic Data export within the Patient Viewer interface • Can create Gene Sets • Can view queries and gene sets	• Cohort Query Tab • Cohort Query Tab: Genomic Data (tab in accordion) • Query Tab: Relative Time Events: gene variant (radio button) • Cohort Viewer (top tab) • Cohort Viewer: Genomic Data Export • My Workspace: My Recent Queries • Gene Sets • Jobs

Role	Permissions	Accessible Screens
trc-pi-user-group	• Has specific privileges allowing access to identifiable information on patients or subjects • In Subject context, can view merged version of Patient and Subject clinical data all in one View Record page.	• View personally identifiable (PI) information on the Cohort List page. • View PI data in Single Patient or Subject Viewer, View Record page. • Cohort Query Tab • Cohort Query Tab: Genomic Data (tab in accordion) • Query Tab: Relative Time Events: gene variant (radio button) • Cohort Viewer (top tab) • Cohort Viewer: Cohort List • Cohort List (PI attributes) • Cohort Viewer: Cohort Timeline • Cohort Reports • Cohort Viewer: Genomic Data Export • Single Patient Viewer: View Record • Single Patient Viewer: View Record (PI attributes) • Single Subject Viewer: View Record (PI attributes) • Single Subject Viewer: View Record (show patient data together) • Single Patient Viewer: View Record: Genomic Data Collected • Circular Genomic Viewer (Visquick) • Genomic Query • Genomic Query: Columns after searching for gene/variant coming from CDM (Patient Count, Specimen Anatomical Site columns) • My Workspace: My Recent Queries • Short Cuts • My Cohort Lists • Queries or Lists shared with me • Queries or Lists shared with All • My Queries • Gene Sets • Saved Queries • Jobs
trc-admin-group	• Can create user accounts and user groups for sharing cohort queries or lists • Can add or remove users from groups • Can manage user roles	Manage User Group

Administrate user groups

Administrating user groups can be done directly from the Oracle Healthcare Translational Research UI.

To simplify sharing, create user groups and assign users to them. Instead of sharing queries or lists with each user individually, set up a list of users to share multiple items to multiple users at the same time.

> **Note:**
>
> For more details on how to create, edit or delete user groups in the UI, check the *Oracle Healthcare Translational Research User's Guide*, on the .

Set limits for exporting variants

In Oracle Healthcare Translational Research, in the Single Patient Viewer screen, users can only export variants separately, instead of doing it for all phenotypes at once.

If the number of variants that they want to export is less or equal than the pre-configured limit of 50000, then all variants for the selected node are exported. If the number of variants that they want to export exceeds that limit, variants are then exported only for the selected chromosome within the node.

In some cases, that limit can be modified from the database. To change the limit:

1. Log in the Oracle Healthcare Foundation / Oracle Healthcare Translational Research database using the Enterprise (ENT) schema user or as an admin user.
2. Find the current value of the Variant Export limit using this query:

   ```
   Select value from W_EHA_CONFIG_PARAMETER where name
   ='variant_export';
   ```

3. Update the variant export limit with a different value, using this query:

   ```
   Update W_EHA_CONFIG_PARAMETER set value=<new_variant_export_limit>
   where name='variant_export';
   ```

   Instead of <new_variant_export_limit> add the new limit for exporting variants.
4. Check with the user who wants to export variants to see if everything works.

2 Manage access to patient or subject records and PII

In this chapter you will learn to:

Use data access policies

Use data access policies to control users' access to patient and subject data in several ways:

- **Control access to Personally Identifiable Information (PII) attributes**. Create data access policies that specify which Personally Identifiable Information (PII) attributes are visible for subjects in a particular study or patients in a particular patient group, to users assigned to the configuration. PII attributes that are not visible are obfuscated (masking values are displayed).
  This functionality is always enabled so that anyone needing access to PII data must be assigned to a data access policy that grants the required access.
- **Control access to non-PII subject or patient information** by enabling row-level filtering. This setting applies across all studies and patient groups.
  - If disabled (the default state), users can see all non-PII subject or patient data in any study or patient group.
  - If enabled, only users assigned to a data access policy that allows access to a study or patient group can see any non-PII subject or patient data in the study or patient group.
- **Allow access to all data** by assigning a user to a global configuration that allows access to all subject and patient data, including PII data, in all studies and patient groups.
- **Control access to omics data stored in the Omics Data Bank Schema.** Create data access policies containing attributes at the patient level that control whether the all patient omics results can be seen by users or not. The same model applies at a subject level. Data access policies can be created through patient groups for patients and through studies for subjects.
  Moreover, an expiration date is introduced for the omics access attribute. Once the expiration date has passed, all access restrictions are removed automatically.

If a user has access to the same study or patient group through multiple data access policies, if any of the data access policies permits access to a particular subject or patient's data, it is visible to the user.

> **Note:**
>
> In previous versions of Oracle Healthcare Translational Research, data access policies were called VPD configurations.

For more informatin, see:

- Create a data access policy
- Assign a user to a data access policy
- Deactivate a data access policy
- Limit access to non-PII data in patient and subject records
- Grant access to all subject and patient data
- Attribute groups
- Import subject and patient data to the Cohort Data Model
- Import omics reference and result data to the Omics Data Bank
- Read some use cases for more context

Create a data access policy

Specify a study or patient group and a set of personally identifiable information (PII) that should be visible to certain users. All users assigned to the data access policy have access to the specified PII data for the subjects or patients in the specified study or patient group. If some users should have access to a different set of PII data for the same study or patient group, create a different data access policy for them.

> **Note:**
>
> The same steps apply when creating a data access policy for the Omics Data Bank.

To create a data access policy:

1. On the database server, log in to SQL*Plus as CDM.
2. Run VPD_UTIL.ADD_VPD_CONFIG, entering values as follows:
   - A unique name for the data access policy.
   - The name of one study or patient group whose data will be accessible through the data access policy.
   - A description of the data access policy.
   - A value for every attribute group parameter containing either `subj` or `pt`, depending on whether you specified a study or a patient group.
     A value of `Y` allows access. `N` prevents access. For a description of each attribute group, see:

     Table 2-1

     Table 2-2

     Table 2-3
   - **Optional!** An expiration date for the data access policy in the format `YYYY-MM-DD`. On this date the data access policy will be automatically deactivated and any users assigned to it will no longer be able to access subject/patient data through it.

> **Note:**
>
> For data access policies for Omics Data Bank, the date applies only when I_PT_ODB or I_SUBJ_ODB are set to N (No). This means that the data access policy restricts access to omics data for users. In these cases, the date represents the last day of the restriction.

Example 1 - Data access policy with some PII access to 'Study B' subjects

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG    (I_CONFIG_NAME  =>
'STUDY_B_SUBJECTS_FULL_PII' ,
I_SUBJ_ADD     =>    'N'                                    ,
    I_SUBJ_CONSENT =>    'Y'                                    ,
    I_SUBJ_DATE    =>    'Y'                                    ,
    I_SUBJ_DX      =>    'Y'                                    ,
    I_SUBJ_ENC     =>    'Y'                                    ,
    I_SUBJ_HIST    =>    'Y'                                    ,
    I_SUBJ_ID      =>    'Y'                                    ,
    I_SUBJ_MED     =>    'Y'                                    ,
    I_SUBJ_NAME    =>    'N'                                    ,
    I_SUBJ_OBSV    =>    'Y'                                    ,
    I_SUBJ_PROC    =>    'Y'                                    ,
    I_SUBJ_SPEC       =>    'Y'                                    ,
I_DESCRIPTION  =>    'data access policy with access to all Study B
subjects with all PII attributes except subject name and address',
I_SUBJECT_STUDY=>'Study B'
    );
END;
/
```

Example 2 - Data access policy with full PII access to GROUP1 patients

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG    (I_CONFIG_NAME  =>
'GROUP1_PATIENTS_ALL_PII' ,
    I_PT_ADD     =>    'Y'                                    ,
    I_PT_CONSENT =>    'Y'                                    ,
    I_PT_DATE    =>    'Y'                                    ,
    I_PT_DX      =>    'Y'                                    ,
    I_PT_ENC     =>    'Y'                                    ,
    I_PT_HIST    =>    'Y'                                    ,
    I_PT_ID      =>    'Y'                                    ,
    I_PT_MED     =>    'Y'                                    ,
    I_PT_NAME    =>    'Y'                                    ,
    I_PT_OBSV    =>    'Y'                                    ,
    I_PT_PROC    =>    'Y'                                    ,
    I_PT_SPEC     =>    'Y'                                    ,
I_DESCRIPTION  =>    'data access policy with access to GROUP1 patients
for all PII attributes',
    I_PATIENT_GROUP  =>'GROUP1',
I_EXPIRATION_DATE       => date '2025-12-31'
    );
```

```
END;
/
```

Example 3 - Data access policy with full PII access to Omics Data Bank

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG    (I_CONFIG_NAME  =>
'GROUP1_PATIENTS_ALL_PII' ,
    I_PT_ADD     =>    'Y'                                    ,
    I_PT_CONSENT =>    'Y'                                    ,
    I_PT_DATE    =>    'Y'                                    ,
    I_PT_DX      =>    'Y'                                    ,
    I_PT_ENC     =>    'Y'                                    ,
    I_PT_HIST    =>    'Y'                                    ,
    I_PT_ID      =>    'Y'                                    ,
    I_PT_MED     =>    'Y'                                    ,
    I_PT_NAME    =>    'Y'                                    ,
    I_PT_OBSV    =>    'Y'                                    ,
    I_PT_PROC    =>    'Y'                                    ,
    I_PT_SPEC        =>
'Y'                                    ,
I_PT_ODB    =>    'Y'                                    ,
I_DESCRIPTION  =>    'data access policy with access to GROUP1 patients
for all PII attributes and Omics Data Bank',
    I_PATIENT_GROUP  =>'GROUP1',
END;
/
```

Assign a user to a data access policy

Assign users to a data access policy to give them permission to see the specified PII for the specified subjects or patients. You can assign either Oracle WebLogic Server user accounts or database user accounts, one account at a time.

1. On the database server, log in to SQL*Plus as CDM.
2. Run VPD_UTIL.ADD_CONFIG_USER, entering values as follows:
   - The data access policy name.
   - The user's user name.
   - (Optional) An expiration date for the user assignment in the format `YYYY-MM-DD`. On this date the user will be automatically deassigned and will no longer be able to access subject/patient data through the data access policy.

   For example:

   ```
   BEGIN
   VPD_UTIL.ADD_CONFIG_USER
      (I_EXISTING_CONFIG_NAME  =>    'PATIENT_GROUP_1_ALL_ATTRIBUTES' ,
      I_USER_NAME  => 'TESTER' ,
      I_EXPIRATION_DATE       => DATE '2025-12-31'   );
    END;/
   ```

Deactivate a data access policy

Deactivating a data access policy removes the specified data access from all the users assigned to it, though users may have access through a different data access policy.

1. On the database server, log in to SQL*Plus as CDM.
2. Run stored procedure VPD_UTIL.INACTIVATE_CONFIG, entering the data access policy name for I_EXISTING_CONFIG_NAME:

   ```
   exec VPD_UTIL.INACTIVATE_CONFIG;
   ```

Limit access to non-PII data in patient and subject records

By default, any application user is permitted to access non-PII records for all patients and subjects. If required, the row-level filtering mode can be turned on at the system level, which limits user access to only a subset of patients or subjects. If this optional mode is turned on, only users explicitly assigned to a data access policy are granted access to the patients or subjects associated with the policy. A user can be assigned to any number of policies.

1. On the database server, log in to SQL*Plus as CDM.
2. Run stored procedure VPD_UTIL.ENABLE_ROW_FILTER_POLICIES:
   - To require that users must be assigned to a data access policy to see any subject or patient data for a particular study or patient group, enter a value of `1`:

     ```
     exec vpd_util.enable_row_filter_policies(1)
     ```

   - To enable all users to see non-PII data for any subject or patient, enter a value of `0`. This is the default value.

     ```
     exec vpd_util.enable_row_filter_policies(0)
     ```

Grant access to all subject and patient data

A global data access policy permits access to all patients and subjects and all their PII attribute values. Users who are assigned to this data access policy do not need to be assigned to any other data access policy, even if row filtering is on. Its ID value is `1`.

1. On the database server, log in to SQL*Plus as CDM.
2. Run stored procedure VPD_UTILADD_CONFIG_USER entering values as follows:
   - The configuration ID set to `1`.
   - The user's user name.
   - (Optional) An expiration date for the user assignment in the format `YYYY-MM-DD`. On this date the user will be automatically deassigned and will no longer be able to access subject/patient data through the data access policy.

For example:

```
BEGIN
VPD_UTIL.ADD_CONFIG_USER
  (I_EXISTING_CONFIG_ID  =>    1 ,
   I_USER_NAME  => 'JSMITH' ,
   I_EXPIRATION_DATE       => DATE '2025-12-31'   );
 END;
/
```

Attribute groups

Some data access policies grant data access to groups of attributes, while others can allow access only for certain attributes.

PII attributes are combined into PII attribute groups. For more information on attribute groups, see:

- Table 2-2
- Table 2-3

For more information on individual Omics Attributes, see: Table 2-1

Table 2-1   Omics Data Attributes

Subject and Patient Attributes	Description	Table	API Procedure Input Parameter
SUBJ_ODB	Defines whether patient omics data is granted (Y) or denied (N) in a configuration	W_EHA_VPD_CONFIG	I_SUBJ_ODB
SUBJ_ODB_EXP_DT	Defines expiration date for N value in SUBJ_ODB	W_EHA_VPD_CONFIG	I_SUBJ_ODB_EXP_DT
PT_ODB	Defines whether patient omics data is granted (Y) or denied (N) in a given configuration	W_EHA_VPD_CONFIG	I_PT_ODB
PT_ODB_EXP_DT	Defines expiration date for N value in PT_ODB	W_EHA_VPD_CONFIG	I_PT_ODB_EXP_DT

Table 2-2   Personally Identifiable Attribute Groups for Subjects

Subject Attribute Groups	Description	Table	Column(s)	API Procedure Input Parameter
SUBJ_ADD	Subject Address	W_EHA_SUBJECT_D	CITY, POSTAL_CODE, STREET_ADDRESS_1, STREET_ADDRESS_2, STREET_ADDRESS_3	I_SUBJ_ADD

Table 2-2 (Cont.) Personally Identifiable Attribute Groups for Subjects

Subject Attribute Groups	Description	Table	Column(s)	API Procedure Input Parameter
SUBJ_CONSENT	Subject Consent Dates	W_EHA_ENC_PATIENT_H	CONSENT_START_DT, CONSENT_END_DT	I_SUBJ_CONSENT
SUBJ_DATE	Subject Lifecycle Dates	W_EHA_SUBJECT_D	DOB, DECEASED_DT	I_SUBJ_DATE
SUBJ_DX	Subject Diagnosis Dates	W_EHA_DX_SUBJECT_H	DIAGNOSIS_ONSET_DT, DIAGNOSIS_REPORTED_DT, DIAGNOSIS_END_DT, AGE_AT_FIRST_ONSET	I_SUBJ_DX
SUBJ_ENC	Subject Encounter Dates	W_EHA_ENC_SUBJECT_H	ENCOUNTER_START_DT, ENCOUNTER_END_DT	I_SUBJ_ENC
SUBJ_HIST	Subject History	W_EHA_SBJ_HISTORY_SBJ_H	SUBJECT_HISTORY_START_DT, SUBJECT_HISTORY_END_DT	I_SUBJ_HIST
SUBJ_ID	Subject Identifier	W_EHA_SUBJECT_D	SUBJECT_IDENTIFIER	I_SUBJ_ID
SUBJ_MED	Subject Medication Dates	W_EHA_SUBADMN_SUBJECT_H	SUBADMN_START_DT, SUBADMN_END_DT	I_SUBJ_MED
SUBJ_NAME	Subject Name	W_EHA_SUBJECT_D	FIRST_NAME, MIDDLE_NAME, LAST_NAME	I_SUBJ_NAME
SUBJ_OBSV	Subject Observation Dates	W_EHA_OBSV_SUBJECT_H	OBSV_DT	I_SUBJ_OBSV
SUBJ_PROC	Subject Procedure Dates	W_EHA_PROC_SUBJECT_H	PROCEDURE_START_DT, PROCEDURE_END_DT	I_SUBJ_PROC
SUBJ_SPEC	Subject Specimen Identifier and Collection Date	W_EHA_SPECIMEN_SUBJECT_H	SPECIMEN_COLLECTION_DT, SPECIMEN_NUMBER	I_SUBJ_SPEC

Table 2-3 Personally Identifiable Attribute Groups for Patients

Patient Attribute Groups	Description	Table	Column(s)	API Procedure Input Parameter
PT_ADD	Patient Address	W_EHA_RESEARCH_PATIENT_D	CITY, POSTAL_CODE, STREET_ADDRESS_1, STREET_ADDRESS_2, STREET_ADDRESS_3	I_PT_ADD
PT_CONSENT	Patient Consent Dates	W_EHA_CONSENT_PATIENT_H	CONSENT_START_DT, CONSENT_END_DT	I_PT_CONSENT

Table 2-3 (Cont.) Personally Identifiable Attribute Groups for Patients

Patient Attribute Groups	Description	Table	Column(s)	API Procedure Input Parameter
PT_DATE	Patient Lifecycle Dates	W_EHA_RESEARCH_PATIENT_D	DOB, DECEASED_DT	I_PT_DATE
PT_DX	Patient Diagnosis Dates	W_EHA_DX_PATIENT_H	DIAGNOSIS_ONSET_DT, DIAGNOSIS_REPORTED_DT, DIAGNOSIS_END_DT, AGE_AT_FIRST_ONSET	I_PT_DX
PT_ENC	Patient Encounter Dates	W_EHA_ENC_PATIENT_H	ENCOUNTER_START_DT, ENCOUNTER_END_DT	I_PT_ENC
PT_HIST	Patient History	W_EHA_PT_HISTORY_PT_H	PATIENT_HISTORY_START_DT, PATIENT_HISTORY_END_DT	I_PT_HIST
PT_ID	Patient Identifier	W_EHA_RESEARCH_PATIENT_D	PATIENT_IDENTIFIER	I_PT_ID
PT_MED	Patient Medication Dates	W_EHA_SUBADMN_PATIENT_H	SUBADMN_START_DT, SUBADMN_END_DT	I_PT_MED
PT_NAME	Patient Name	W_EHA_RESEARCH_PATIENT_D	FIRST_NAME, MIDDLE_NAME, LAST_NAME	I_PT_NAME
PT_OBSV	Patient Observation Dates	W_EHA_OBSV_PATIENT_H	OBSV_DT	I_PT_OBSV
PT_PROC	Patient Procedure Dates	W_EHA_PROC_PATIENT_H	PROCEDURE_START_DT, PROCEDURE_END_DT	I_PT_PROC
PT_SPEC	Patient Specimen Identifier and Collection Date	W_EHA_SPECIMEN_PATIENT_H	SPECIMEN_COLLECTION_DT, SPECIMEN_NUMBER	I_PT_SPEC

Import subject and patient data to the Cohort Data Model

To make subject and patient data available in Oracle Healthcare Translational Research (OHTR):

1. Load the data into Oracle Healthcare Data Warehouse (HDM) using the Healthcare Data Interface tables (HDI).

2. Propagate the data to the Cohort Data Model (CDM) using either Informatica or Oracle Data Integrator (ODI) ETLs.

Instructions for loading data can be found in the *Oracle Healthcare Foundation Administrator's Guide* for Informatica and for Oracle Data Integrator. The guides are available in password-protected patch 22640545. To get the password, log a Service Request (SR) on My Oracle Support.

> **Note:**
>
> In the Single Patient Viewer screen, the consent records are displayed only if a valid Data Source Number is populated in the W_EHA_CONSENT_PATIENT_H.DATASOURCE_NUM_ID column. Make sure that the value populated in this column exists in the W_EHA_DATASOURCE_CDM.ROW_WID column, as well.

Import omics reference and result data to the Omics Data Bank

Instructions for loading data can be found in the *Oracle Healthcare Foundation Administrator's Guide* for Informatica and for Oracle Data Integrator. The guides are available in password-protected patch 22640545. To get the password, log a Service Request (SR) on My Oracle Support.

Read some use cases for more context

Scenario A

Dr. Smith needs to view all patient and subject data with de-identified PII. She works in an environment where row-level filtering is disabled, meaning assignment to a data access policy is not required.

There is no need to explicitly assign a data access policy to Dr. Smith. In her environment, any user has access to non-PII data for all patients and subjects.

Scenario B

Dr. Chen needs to see patient data in Patient Group 1, including all PII values except patient name and address. She works in an environment where access to patient and subject records is controlled. To give her access:

1. Create a data access policy for the patient group that Dr. Chen has access to view. Set all patient attribute groups in the policy except Patient name and address to Y.

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG    (I_CONFIG_NAME  =>
'GROUP1_PATIENTS_ALL_PII_EXCEPT_NAME_ADDRESS' ,
    I_PT_ADD     =>    'N'                                    ,
    I_PT_CONSENT =>    'Y'                                    ,
    I_PT_DATE    =>    'Y'                                    ,
    I_PT_DX      =>    'Y'                                    ,
    I_PT_ENC     =>    'Y'                                    ,
    I_PT_HIST    =>    'Y'                                    ,
    I_PT_ID      =>    'Y'                                    ,
    I_PT_MED     =>    'Y'                                    ,
    I_PT_NAME    =>    'N'                                    ,
    I_PT_OBSV    =>    'Y'                                    ,
    I_PT_PROC    =>    'Y'                                    ,
    I_PT_SPEC     =>    'Y'                                    ,
I_DESCRIPTION  =>    'Configuration with access to GROUP1 patients with PII
attributes except Name and Address',
    I_PATIENT_GROUP  =>'GROUP1',
I_EXPIRATION_DATE       => date '2025-12-31'
    );
END;
/
```

2. Assign Dr. Chen (jchen12) to the above data access policy.

```
BEGIN
VPD_UTIL.ADD_CONFIG_USER
  (I_EXISTING_CONFIG_NAME   =>
'GROUP1_PATIENTS_ALL_PII_EXCEPT_NAME_ADDRESS'   ,
   I_USER_NAME             => 'jchen12',
   I_EXPIRATION_DATE       => date '2025-12-31');
END;
/
```

Scenario C

Dr. Gupta is authorized to see all patient and subject data, including identifiable data.

```
BEGIN
VPD_UTIL.ADD_CONFIG_USER
  (I_EXISTING_CONFIG_ID    => 1,
   I_USER_NAME             => 'kgupta',
   I_EXPIRATION_DATE       => date '2025-12-31');
END;
/
```

Scenario D

Dr. Black needs to see de-identified data in STUDY A and identified data in STUDY B.

1. Create a data access policy that grants access to de-identified PII data on subjects from STUDY A.

> **Note:**
>
> This step is optional when row-level filtering is disabled.

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG     (I_CONFIG_NAME  =>
'STUDY_A_SUBJECTS_NO_PII' ,
  I_SUBJ_ADD     =>    'N'                                    ,
    I_SUBJ_CONSENT =>    'N'                                    ,
    I_SUBJ_DATE    =>    'N'                                    ,
    I_SUBJ_DX      =>    'N'                                    ,
    I_SUBJ_ENC     =>    'N'                                    ,
    I_SUBJ_HIST    =>    'N'                                    ,
    I_SUBJ_ID      =>    'N'                                    ,
    I_SUBJ_MED     =>    'N'                                    ,
    I_SUBJ_NAME    =>    'N'                                    ,
    I_SUBJ_OBSV    =>    'N'                                    ,
    I_SUBJ_PROC    =>    'N'                                    ,
    I_SUBJ_SPEC       =>    'N'                                    ,
  I_DESCRIPTION  =>    'Configuration with access to Study A 
subjects with no PII attribute values',
  I_SUBJECT_STUDY=>'Study A'
    );
END;
/
```

2. Create a data access policy that grants access to subjects from STUDY B and their identifiable attribute values:

```
BEGIN
VPD_UTIL.ADD_VPD_CONFIG    (I_CONFIG_NAME  => 
'STUDY_B_SUBJECTS_FULL_PII' ,
I_SUBJ_ADD     =>    'Y'                                    ,
    I_SUBJ_CONSENT =>    'Y'                                    ,
    I_SUBJ_DATE    =>    'Y'                                    ,
    I_SUBJ_DX      =>    'Y'                                    ,
    I_SUBJ_ENC     =>    'Y'                                    ,
    I_SUBJ_HIST    =>    'Y'                                    ,
    I_SUBJ_ID      =>    'Y'                                    ,
    I_SUBJ_MED     =>    'Y'                                    ,
    I_SUBJ_NAME    =>    'Y'                                    ,
    I_SUBJ_OBSV    =>    'Y'                                    ,
    I_SUBJ_PROC    =>    'Y'                                    ,
    I_SUBJ_SPEC       =>    'Y'                                    ,
I_DESCRIPTION  =>    'Configuration with access to Study B subjects 
with all PII attribute values',
I_SUBJECT_STUDY=>'Study B'
    );
END;
/
```

3. Assign Dr. Black to these data access policies:

> **Note:**
>
> Assigning `STUDY_A_SUBJECTS_NO_PII` configuration is optional if row-level filtering is disabled.

```
BEGIN
VPD_UTIL.ADD_CONFIG_USER
  (I_EXISTING_CONFIG_NAME  => 'STUDY_A_SUBJECTS_NO_PII',
   I_USER_NAME             => 'sblack',
   I_EXPIRATION_DATE       => date '2025-12-31');
END;
/
```

```
BEGIN
VPD_UTIL.ADD_CONFIG_USER
  (I_EXISTING_CONFIG_NAME  => 'STUDY_B_SUBJECTS_FULL_PII',
   I_USER_NAME             => 'sblack',
   I_EXPIRATION_DATE       => date '2025-12-31');
END;
/
```

3

Manage access to specimen aliases

In this chapter you will learn:

About specimen aliases and permissions

A laboratory that receives a specimen for processing may assign a barcode and use it for specimen identification purposes. In the Cohort Data Model (CDM) schema, these barcode identifiers are tracked as specimen aliases in addition to the primary lab specimen identifier (SPECIMEN_NUMBER/SPECIMEN_VENDOR_NUMBER).

The permissions to see specimen aliases are calculated based on the service provider (lab) that issued the alias and study or patient group that includes the specimen donor (patient or subject).

You will also learn to:

- Authorize specimen alias access
- Revoke specimen alias access for a user

Authorize specimen alias access

For each user who needs access to specimen aliases, specify:

- Either a specific service provider (such as a lab) or all providers
- A specific study or patient group
- All studies or all patient groups

If a user needs access to specimen aliases from more than one provider but not all, or to specimen aliases used in more than one study or patient group but not all, run this procedure once for each combination required:

1. On the database server, log in to SQL*Plus as CDM.
2. Run stored procedure VPD_UTIL.GRANT_SVCPRV_USER, entering values as follows:
   - The user's user name.
   - Service provider scope. Set one of the following parameters:
     – `I_ISSNG_SVCPRV_ID`. To limit access to aliases used by a single service provider, enter the service provider's ID.
     – `I_ANY_ISSNG_SVCPRVS`. To allow access to aliases created by any service provides, set this parameter s to `1`.
   - Subject and/or patient scope. Set one of the following parameters, or set one study parameter and one patient group parameter:
     – `I_STUDY_NAME`. To allow access to aliases used in a single study, enter the name of the study.

  - `I_ANY_STUDY`. To allow access to aliases used in all studies, set this parameter to `1`.
  - `I_PT_GROUP_NAME`. To allow access to aliases used for a single patient group, enter the name of the patient group.
  - `I_ANY_PT_GROUP`. To allow access to aliases used in all patient groups, set this parameter to `1`.
- **(Optional)** An expiration date for the user assignment. On this date the privileges given to the user will be automatically revoked. The date can be formatted in any valid date-type expression. For example:

```
 sysdate+x_days
date '2020-12-31'
trunc(sysdate +1217
to_date('2020-12-31','YYYY-MM-DD')
```

Example 1: Access to a single provider's aliases for a single patient group

Authorize a user to access aliases for a specific service provider in the context of a single patient group.

```
begin
 vpd_util.grant_svcprv_user (
  i_user_name =>' LABUSER1',
  i_pt_group_name =>'GROUP_1',
  i_issng_svcprv_id =>'SVCPRV1',
  i_expiration_date =>sysdate+100
) ;
end;
/
```

Example 2: Access to a single provider's aliases for a single study

Authorize a user to access aliases for a specific service provider in the context of a single study.

```
begin
 vpd_util.grant_svcprv_user (
  i_user_name =>' LABUSER2',
  i_study_name =>'STUDY3',
  i_issng_svcprv_id =>'18_SVCPRV_NBR',
  i_expiration_date =>sysdate+100
) ;
end;
/
```

Example 3: Access to any provider's aliases for any study or patient group

Authorize a user to access any specimen alias in the context of any patient group or study.

```
begin
 vpd_util.grant_svcprv_user (
  i_user_name =>' LABSUPEVISOR1',
  i_any_study =>1,
  i_any_pt_group =>1,
  i_any_issng_svcprvs =>1,
  i_expiration_date =>sysdate+100
) ;
end;
/
```

Revoke specimen alias access for a user

> **Note:**
>
> To revoke specimen alias access from a user, use numeric identifiers (row_wids) of the specific study or patient group.

1. On the database server, log in to SQL*Plus as CDM.
2. Run stored procedure VPD_UTILrevoke_svcprv_user, entering values as follows:
   - The user's user name.
   - Service provider scope. Set one of the following parameters:
     - `I_ISSNG_SVCPRV_ID`. To revoke access from aliases used by a single service provider, enter the service provider's ID.
     - `I_ANY_ISSNG_SVCPRVS`. To revoke access from aliases created by any service provides, set this parameter s to `1`.
   - Subject and/or patient scope. Set one of the following parameters, or set one study parameter and one patient group parameter:
     - `I_STUDY_NAME`. To revoke access from aliases used in a single study, enter the name of the study.
     - `I_ANY_STUDY`. To revoke access from aliases used in all studies, set this parameter to `1`.
     - `I_PT_GROUP_NAME`. To revoke access from aliases used for a single patient group, enter the name of the patient group.
     - `I_ANY_PT_GROUP`. To revoke access from aliases used in all patient groups, set this parameter to `1`.

     > **Note:**
     >
     > For the purpose of revoking access, the parameters for *any study*, *any pt group*, and *any provider* do not include any separately granted named study, patient, or provider scope. You must revoke these separately if needed.

Example 1: Revoke access to one study/provider combination

Revoke access to study_wid=99 and Service Provider (row_wid=105) from user TSTUSER1.

```
begin
vpd_util.revoke_svcprv_user (
    i_user_name =>'TSTUSER1',
  i_study_wid =>99,
  i_issng_svcprv_wid =>105
) ;
end;
/
```

Example 2: Revoke access to one patient group/provider combination

Revoke access to patient_group_wid=10 and Service Provider (row_wid=105) from user TSTUSER2.

```
begin
vpd_util.revoke_svcprv_user (
    i_user_name =>'TSTUSER2',
i_pt_group_wid =>10,
  i_issng_svcprv_wid =>105
) ;
end;
/
```

Example 3: Revoke access to any study, patient group, and provider granted using parameter I_ANY_STUDY, I_ANY_PT_GROUP, and I_ANY_ISSNG_SVCPRVS

Revoke access from a user to any study or patient group or service provider that was granted through an “any" parameter.

```
begin
vpd_util.revoke_svcprv_user (
    i_user_name =>'TSTUSER3',
  i_any_study =>1,
  i_any_pt_group =>1,
  i_any_issng_svcprvs =>1
) ;
end;
/
```

4

Optimize query engine performance

To optimize the performance of the query engine, refresh statistics by running stored procedure REFRESH_STATS_TAB.

Execute this procedure:

- After the initial CDM data load.
- After any significant changes in data volumes or distribution in the CDM schema.
- Periodically.

To execute the procedure:

1. On the database server, log in to SQL*Plus as CDM.
2. Execute the following commands:

   ```
   set serveroutput on
   set echo on
   spool REFRESH_STATS_TAB
   execute REFRESH_STATS_TAB
   spool off
   exit
   ```