

Oracle Health Insurance Back Office

Agent Installation & Configuration Manual

Version 1.1

Part number: G49637-01

July 31, 2026

Copyright © 2026, Oracle and/or its affiliates. All rights reserved.

Public

Purpose statement

This document provides an overview of the architecture used for the Oracle Health Insurance Back Office AI Agents and installation and configuration instructions to integrate the AI Agents in your Oracle Health Insurance Back Office environments.

Disclaimer

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

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

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

U.S. GOVERNMENT RIGHTS

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

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

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

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

Where an Oracle offering includes third party content or software, we may be required to include related notices. For information on third party notices and the software and related documentation in connection with which they need to be included, please contact the attorney from the Development and Strategic Initiatives Legal Group that supports the development team for the Oracle offering. Contact information can be found on the Attorney Contact Chart.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Change History

Release	Version	Changes
10.26.5.0.0	1.0	Creation
10.26.6.0.0	1.1	Documented the use of an Autonomous Transaction Processing (ATP) database instead of a regular 26ai database for the RAG component. Added instructions for OCI privileges that are needed by the OCI principal account used for accessing an OCI Generative AI endpoint.

Table of contents

Purpose	5
Installing the OHI Online Help AI Assistant objects in an Oracle database 26ai PDB	5
Overview	5
Scope and assumptions	6
Release scope	7
DBA actions before running the installer	7
DBMS_CLOUD and Oracle AI packages	8
Network access to OCI endpoints	9
Vector memory	10
Database accounts and grants	10
Temporary tablespace	10
User tablespace	10
Database accounts	10
Quota	11
System privileges for OHI_RAG_OWNER	11
Runtime and model-owner privileges	11
OCI credential and AI profile	11
Input files: regular database or Autonomous AI Database	13
Wallet aliases	15
Preparing the SQL scripts	16
Running the installation	16
Installer activities	17
Post-installation validation	17
Trace retention	19
OHI Back Office integration	19
Installing a higher major release of the Online Help data	20
Cloning an OHI Back Office environment	20
Patching an OHI Back Office environment	20

Purpose

This document provides an overview of the architecture used for the Oracle Health Insurance Back Office AI Agents and it contains installation and configuration instructions to integrate the AI Agents in your Oracle Health Insurance Back Office (OHI BO) environments.

Currently, Oracle Health Insurance Back Office has only one AI Agent: the OHI Online Help AI Assistant.

When new AI agents become available, instructions for those agents will be added in this document.

Installing the OHI Online Help AI Assistant objects in an Oracle database 26ai PDB

This chapter describes the database preparation and installation procedure for the OHI Online Help AI Assistant database objects in a dedicated Oracle Database 26ai Pluggable Database (PDB). This can be a regular (non-autonomous) or an Autonomous Transaction Processing Database (ATP).

Currently both types of databases are supported. However, given the very frequent functional enhancements for an autonomous database, future AI agents for OHI BO may only be supported on an autonomous database, so OHI AI agents can use agentic functionality early. Agentic functionality is often released later for regular databases.

At this moment the Standard Edition of the 26ai database is sufficient for running the agentic functionality.

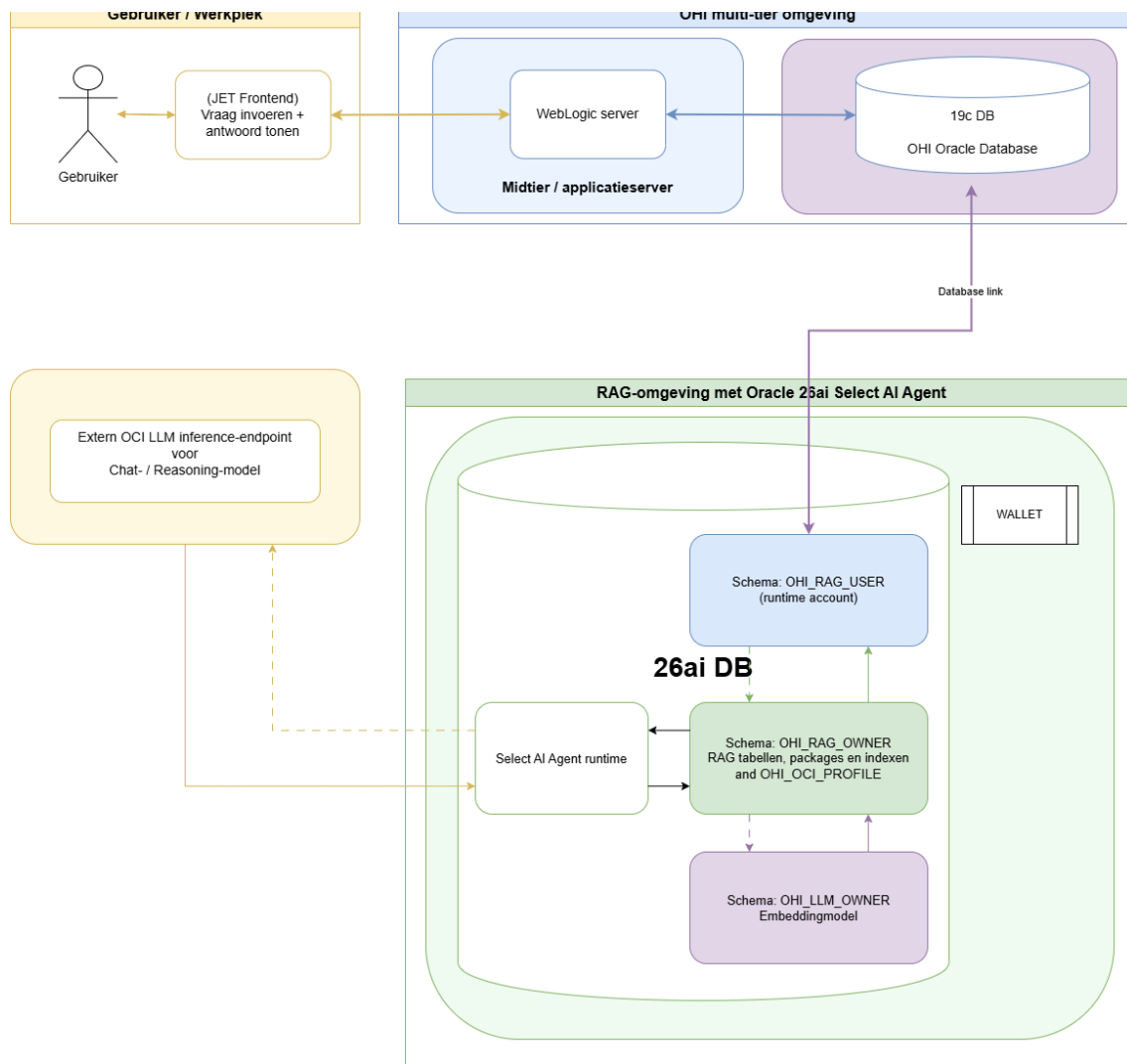
Overview

The database stores and indexes OHI Back Office Online Help content using Oracle Text and Oracle Vector Search. This enables a Retrieval-Augmented Generation (RAG) solution, where a Large Language Model (LLM) combines its reasoning capabilities with information retrieved from the OHI 'RAG database'. The reasoning model determines a search phrase, based on the question asked and the history in the chat conversation. That search phrase is passed to the search function in the RAG database. The search phrase is decomposed in a tokenized sentence to search by means of Oracle Text through the Online Help information and an LLM embedding model is used to generate a vector representation that is used to search in the vector representation of the Online Help information. The top N best scoring results are returned to the reasoning LLM enabling accurate and context-aware responses.

The agentic functionality itself is implemented through [Select AI Agent](#) functionality as this is offered by the Oracle 26ai database.

A database link between the OHI BO database and the 26ai database is used to call the 26ai agentic functionality, where the agent itself uses the described search functionality as a 'tool' to extend its general knowledge with OHI BO specific Online Help information. As such the OHI Online Help agent/assistant has no access to the OHI BO database and cannot answer any question related to the data in the OHI BO environment.

The following figure depicts at a high level how the different components interact.



The installation of the required schema objects in the 26ai database is performed using SQL*Plus or SQLcl and a set of SQL scripts delivered in a ZIP archive. The ZIP archive is located in the \$OZG_BASE/conf/generic directory of the OHI Back Office environment.

As described before, The OHI Online Help AI Assistant leverages the Oracle Database 26ai "Select AI Agent" and Vector Search capabilities.

The procedures in this chapter are intended for database administrators responsible for:

- Preparing the target 26ai PDB.
- Creating and configuring the required database schemas.
- Configuring access to Oracle Cloud Infrastructure (OCI) Generative AI services.
- Executing the database installation scripts and validating the installation.

Scope and assumptions

This installation procedure assumes that an Oracle Database 26ai container database (CDB) and target PDB already exist and that the target PDB is open in 'read write' mode.

This can be a regular CDB and PDB as well as an Autonomous Transaction Processing PDB (running as a service in Oracle Cloud Infrastructure and managed by OCI). Creation of the CDB and/or PDB itself is outside the scope of this chapter. See below for storage requirements.

For the OHI Online Help AI Assistant, the target PDB should have a minimum SGA size of 5 GB available, including at least 1 GB allocated to the shared pool and 1 GB allocated to the database buffer cache. A minimum of 4 OCPUs is recommended. These values are based on the expected workload for the delivered OHI Online Help AI assistant.

The installation creates and configures three database accounts, installs the OHI RAG database objects, configures access to Oracle AI functionality, loads the documents and terms as needed for the RAG solution, and builds the hybrid vector index.

- OHI_RAG_OWNER owns the RAG tables, packages, views, Oracle Text configuration, scheduler job, and hybrid vector index.
- OHI_RAG_USER is the runtime user that calls the RAG and agent packages.
- OHI_LLM_OWNER owns or exposes the embedding model that is used to store vector representation in the vector index for the Online Help textual information.

The installation is executed with Secure External Password Store (SEPS) wallet aliases for the three accounts.

During the installation three files need to be loaded.

One file contains a so called 'embedding' model in ONNX format, which is used to determine the vector values representing the meaning of the text that is embedded. It is a relatively small LLM model that will be loaded into the database.

The Online Help information itself also needs to be loaded and is stored in two *.ndjson files.

Release scope

The OHI RAG installation is tied to a specific OHI release level. Each Database 26ai PDB may contain the RAG installation for one OHI release only. The installer validates the release level of the existing installation and prevents incompatible deployments.

Multiple OHI Back Office environments may share the same RAG PDB, provided that all environments use the same OHI major release and therefore the same Online Help content and RAG metadata.

If support for multiple OHI major releases is required (for example, a production environment running one release and a test environment running a newer release), a separate Database 26ai PDB must be used for each release. This allows each release to maintain its own release-specific (meta)data.

DBA actions before running the installer

Carry out the following actions in the target database, typically as SYS or from a SYSDBA session in a regular database, or using ADMIN in an autonomous database, unless explicitly stated otherwise.

Determine the passwords, default tablespace, quotas, and the input-file location appropriate for the target database type before starting the installation.

- Password for OHI_RAG_OWNER.
- Password for OHI_RAG_USER.
- Password for OHI_LLM_OWNER.
- Default tablespace, for example USERS. Make sure this can grow with 15Gb.
- Tablespace quota for OHI_RAG_OWNER, 10G will be sufficient.
- Tablespace quota for OHI_LLM_OWNER, 5G will be sufficient.
- For a regular database: database host path for the ONNX model and NDJSON files that need to be loaded. For Autonomous AI Database: OCI Object Storage bucket URI prefix for these files.

For a regular database only:

Create a database directory object named OHI_RAG_DIR that points to the operating system path containing the mentioned ONNX model file and the NDJSON data files, for example:

```
create or replace directory OHI_RAG_DIR as '/share01/ai_dir';
```

For a regular database, the operating system path may be chosen according to local standards and must satisfy the following requirements:

- The path must be accessible to the Oracle Database software owner.
- The database must have read access to all files through the OHI_RAG_DIR directory object.
- The ONNX model file and NDJSON files must be present in the directory before the installation is started.
- In Oracle RAC, Exadata, and other clustered regular-database environments, the path must be available on all database nodes. Use shared storage, for example ACFS, to ensure consistent access from all instances. If that is not possible, make sure the sessions used during the installation process described in this document run on the instance that has access to the specified file system location.

DBMS_CLOUD and Oracle AI packages

The OHI Online Help AI Assistant relies on Oracle Database components that provide access to Oracle Cloud Infrastructure (OCI) services and AI functionality:

- DBMS_CLOUD provides database access to OCI resources and credentials management.
- DBMS_CLOUD_AI provides integration with Oracle AI services, including the Select AI Agent functionality used by the OHI Online Help AI Assistant solution.

If on a regular database DBMS_CLOUD, DBMS_CLOUD_AI and DBMS_CLOUD_AI_AGENT are not installed, execute the Oracle-supplied installation scripts shown below. The installation consists of two steps:

- `catclouduser.sql`: Creates and configures the Oracle-maintained schema and supporting objects required by the DBMS_CLOUD package.
- `dbms_cloud_install.sql`: Installs the DBMS_CLOUD and DBMS_CLOUD_AI packages and their supporting database objects.

Use `catcon.pl` so that the scripts are executed in the CDB and all PDBs where the functionality is needed, you can add `-c 'my_pdb_1 my_pdb_2'` to install this for specific PDB's `my_pdb_1` and `my_pdb_2`.

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl \
-u sys/<SYS_PASSWORD> \
-force_pdb_mode 'READ WRITE' \
-b dbms_cloud_install \
-d $ORACLE_HOME/rdbms/admin/ \
-l /tmp \
catclouduser.sql
```

```
$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl \
-u sys/<SYS_PASSWORD> \
-force_pdb_mode 'READ WRITE' \
-b dbms_cloud_install \
-d $ORACLE_HOME/rdbms/admin/ \
-l /tmp \
dbms_cloud_install.sql
```

Network access to OCI endpoints

The instructions below are not needed in an autonomous database.

As SYS in the root container (CDB\$ROOT), grant network access to the OCI Generative AI endpoints used by DBMS_CLOUD and DBMS_CLOUD_AI.

The value supplied for the host parameter should correspond to the OCI Generative AI endpoint hostname for the region configured in the AI profile.

The OCI Generative AI endpoint can be obtained from the OCI Console by navigating to Analytics & AI → Generative AI and selecting the region in which the service is deployed.

The hostname can be derived from the Generative AI endpoint URL by removing the protocol prefix (https://) and any trailing path information.

For example, when the AI profile uses the OCI region eu-frankfurt-1, the Generative AI endpoint URL may be:

```
https://inference.generativeai.eu-frankfurt-1.oci.oraclecloud.com
```

In that case, the hostname to be used in the ACL is:

```
inference.generativeai.eu-frankfurt-1.oci.oraclecloud.com
```

Alternatively, the ACL can be configured with a broader hostname pattern such as *.oraclecloud.com, although using the specific Generative AI endpoint hostname is recommended from a security perspective.

begin

```
DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
  host          => '*.oraclecloud.com',
  lower_port    => 443,
  upper_port    => 443,
  ace           => xs$ace_type(
                    privilege_list => xs$name_list('http', 'http_proxy'),
                    principal_name => upper('C##CLOUD$SERVICE'),
                    principal_type => xs_acl.ptype_db
                  )
);
```

end;

/

Vector memory

Oracle Vector Search uses a Hierarchical Navigable Small World (HNSW) index to efficiently perform approximate nearest-neighbour searches on vector embeddings. Creating and maintaining HNSW indexes requires vector memory, therefore VECTOR_MEMORY_SIZE must be configured before running the installation.

For HNSW index creation, VECTOR_MEMORY_SIZE in CDB\$ROOT must be at least 1 GB. This is a CDB-level setting and must be configured with SCOPE=SPFILE, followed by a database restart. In an autonomous database this is already arranged.

Check the current setting:

```
SHOW PARAMETER vector_memory_size;
```

If the value is less than 1 GB, update it as follows:

```
ALTER SYSTEM SET vector_memory_size = 1G SCOPE=SPFILE;
```

Restart the database and verify that the new value is active before continuing with the installation.

Database accounts and grants

The following actions must be performed in the target pluggable database (PDB).

Temporary tablespace

The OHI RAG accounts use the database default temporary tablespace. In most environments it is not necessary to explicitly specify a temporary tablespace during user creation, because Oracle automatically assigns the database default.

The current default temporary tablespace can be verified as follows:

```
select property_value as default_temp_tablespace
from   database_properties
where  property_name = 'DEFAULT_TEMP_TABLESPACE';
```

User tablespace

Specify a suitable permanent tablespace for the OHI RAG schemas.

Replace <DATA_TABLESPACE> with the tablespace name appropriate for the target environment. This may be the USERS tablespace or a dedicated application tablespace, depending on local database administration standards and storage policies.

Database accounts

Create the three accounts in the target PDB. Replace the password placeholders and tablespace name with values appropriate for the local environment.

```
create user OHI_RAG_OWNER identified by "<OHI_RAG_OWNER_PASSWORD>"
  default tablespace <DATA_TABLESPACE>
  account unlock;
create user OHI_RAG_USER identified by "<OHI_RAG_USER_PASSWORD>"
  default tablespace <DATA_TABLESPACE>
  account unlock;
create user OHI_LLM_OWNER identified by "<OHI_LLM_OWNER_PASSWORD>"
  default tablespace <DATA_TABLESPACE>
  account unlock;
```

Make sure the passwords do not expire, or add an action to your calendar to change them before the expiration date. Do not forget to update the wallet entries too when the passwords are changed (see below).

Quota

Assign quota to the owner schemas. The values shown below are the recommended minimum quotas for the current implementation and may be increased if additional content is expected in the future.

```
alter user OHI_RAG_OWNER quota 10G on <DATA_TABLESPACE>;
alter user OHI_LLM_OWNER quota 5G on <DATA_TABLESPACE>;
```

System privileges for OHI_RAG_OWNER

Grant the privileges required to create the RAG schema objects, text index, scheduler job, public synonym, and Oracle AI integration objects.

```
grant create session to OHI_RAG_OWNER;
grant alter session to OHI_RAG_OWNER;
grant create table to OHI_RAG_OWNER;
grant create view to OHI_RAG_OWNER;
grant create procedure to OHI_RAG_OWNER;
grant create public synonym to OHI_RAG_OWNER;
grant create job to OHI_RAG_OWNER;
grant select any dictionary to OHI_RAG_OWNER;
grant ctxapp to OHI_RAG_OWNER;
grant execute on ctxsys.ctx_ddl to OHI_RAG_OWNER;
grant execute on ctxsys.dbms_vector_chain to OHI_RAG_OWNER;
grant execute on dbms_cloud to OHI_RAG_OWNER;
grant execute on dbms_cloud_ai to OHI_RAG_OWNER;
grant execute on dbms_cloud_ai_agent to OHI_RAG_OWNER;
grant create credential to OHI_RAG_OWNER;
grant create any context to OHI_RAG_OWNER;
```

Runtime and model-owner privileges

Grant the required system privileges to OHI_RAG_USER.

```
grant create session to OHI_RAG_USER;
grant alter session to OHI_RAG_USER;
```

Grant the required system privileges to OHI_LLM_OWNER.

```
grant create session to OHI_LLM_OWNER;
grant alter session to OHI_LLM_OWNER;
grant create mining model to OHI_LLM_OWNER;
grant execute on dbms_cloud to OHI_LLM_OWNER;
grant create credential to OHI_LLM_OWNER;
```

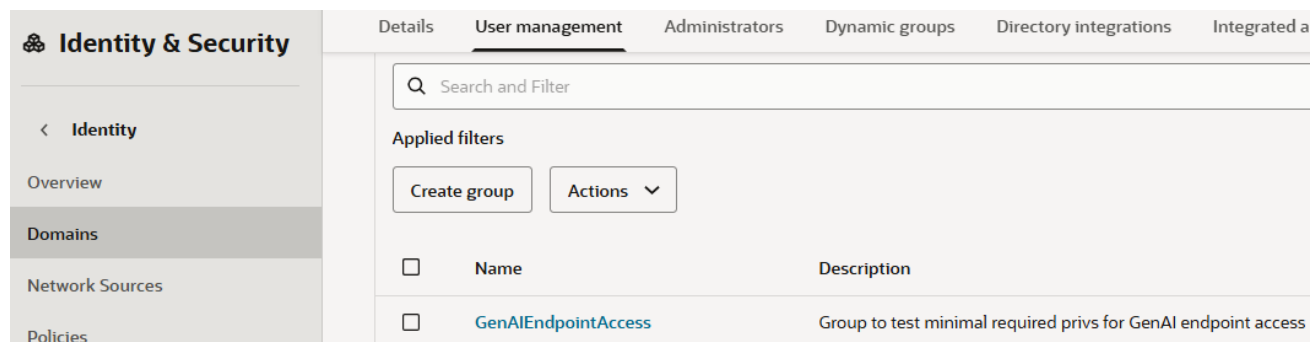
OCI credential and AI profile

Create the OCI credential for the Generative AI profile while connected as OHI_RAG_OWNER. Replace the anonymized example values with the values for the target tenancy, user/principal, key, and fingerprint.

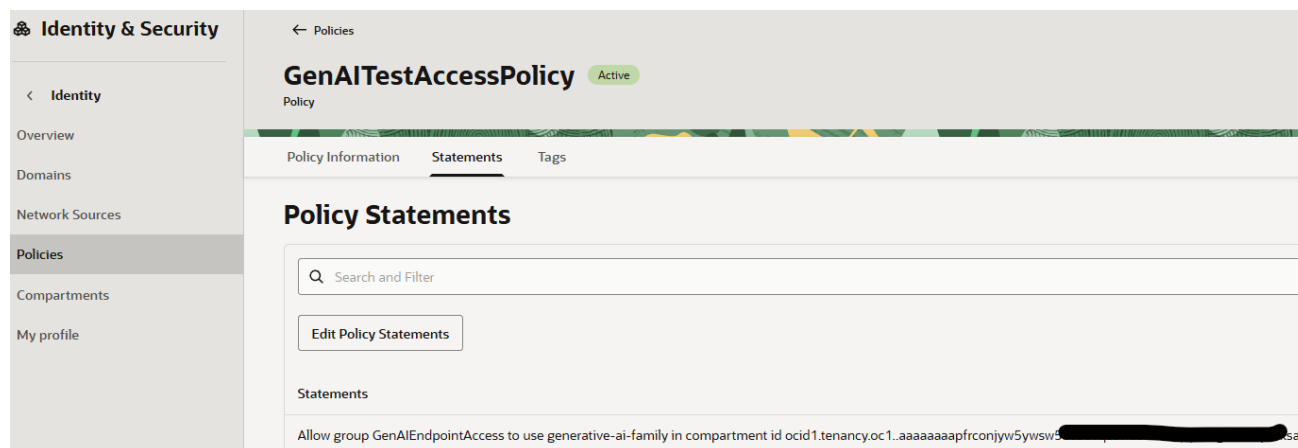
For instructions on obtaining the required OCI values (user OCID, tenancy OCID, API key fingerprint, and private key), refer to the [Oracle Cloud Infrastructure API signing key documentation](#).

Information on locating tenancy and user OCIDs is available in the [OCI documentation](#).

Make sure you use a principal that has received access to the Generative AI services. To provide this access you can create a group within your (default) domain under the User Management tab and grant this specific group access. Below an example group is shown. Assign the relevant users/principals to the group.



An example to provide the group access to the generative AI functionality is shown below by creating a policy with a statement stating the access in a generic way for the tenancy:



For the principal you have chosen to receive the generative AI access you create a credential in the RAG owner schema by the following command (adapted for your situation):

```
begin
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'OHI_OCI_CRED',
    user_ocid       => 'ocid1.user.oc1..example_anonymized_user_ocid',
    tenancy_ocid    => 'ocid1.tenancy.oc1..example_anonymized_tenancy_ocid',
    private_key     => 'EXAMPLE_ANONYMIZED_PRIVATE_KEY_LINE_1
EXAMPLE_ANONYMIZED_PRIVATE_KEY_LINE_2
EXAMPLE_ANONYMIZED_PRIVATE_KEY_LINE_3',
    fingerprint     => '00:00:00:00:00:00:00:00:00:00:00:00:00:00:00',
    comments        => 'OHI OCI Cloud credentials'
  );
end;
/
```

Create the AI profile after the credential is available. The compartment OCID, region, and large language model (LLM) are environment specific.

```

begin
  DBMS_CLOUD_AI.CREATE_PROFILE(
    profile_name => 'OHI_OCI_PROFILE',
    attributes   => JSON_OBJECT(
      'provider'          value 'oci',
      'credential_name'   value 'OHI_OCI_CRED',
      'region'            value 'eu-frankfurt-1',
      'oci_compartment_id' value 'ocid1.compartment.oc1..example_anonymized_compartment_ocid',
      'model'              value 'google.gemini-2.5-flash',
      'max_tokens'         value 65536,
      'object_list'        value JSON_ARRAY(
        JSON_OBJECT('owner' value 'OHI_RAG_OWNER')
      ) format json
    )
  );
end;
/

```

Test the profile before running the RAG installation:

```

select DBMS_CLOUD_AI.GENERATE(
  prompt      => 'tell me something interesting about the city of paris?',
  profile_name => 'OHI_OCI_PROFILE',
  action       => 'chat'
) as response
from dual;

```

For Autonomous AI Database, also create Object Storage credentials in the owning schemas. You may reuse the same credential as created for providing access to the OCI Generative AI services, if that principal will also have access to the bucket.

We will refer to the credential stored in OHI_RAG_OWNER with the name OHI_RAG_OCI_CRED.

We will refer to the credential stored in the OHI_LLM_OWNER schema with the name OHI_LLM_OCI_CRED.

Use the same DBMS_CLOUD.CREATE_CREDENTIAL pattern and grant the underlying OCI principal read access only to the RAG bucket or prefix.

The installation script will ask for the actual credential names and will suggest the credential names above as default credential name.

Input files: regular database or Autonomous AI Database

The installer detects the target database type automatically. On a regular database it reads the input files through OHI_RAG_DIR. On Autonomous AI Database it retrieves them directly from OCI Object Storage, not from a customer-created database directory.

Input files for both database types

- documents.ndjson
- terms.ndjson
- all_MiniLM_L12_v2.onnx

Copy documents.ndjson and terms.ndjson from \$OZG_BASE/help/nl_nl/Output/OHIHelp/RAG. Download all_MiniLM_L12_v2.onnx from the link below:

https://adwc4pm.objectstorage.us-ashburn-1.oci.customer-oci.com/p/TtH6hL2y25EypZ0-rrczRZ1aXp7v1ONbRBfCiT-BDBN8WLKO3lgyW6RxCfIFLdA6/n/adwc4pm/b/OML-ai-models/o/all_MiniLM_L12_v2_augmented.zip

and unzip the downloaded file.

Regular database

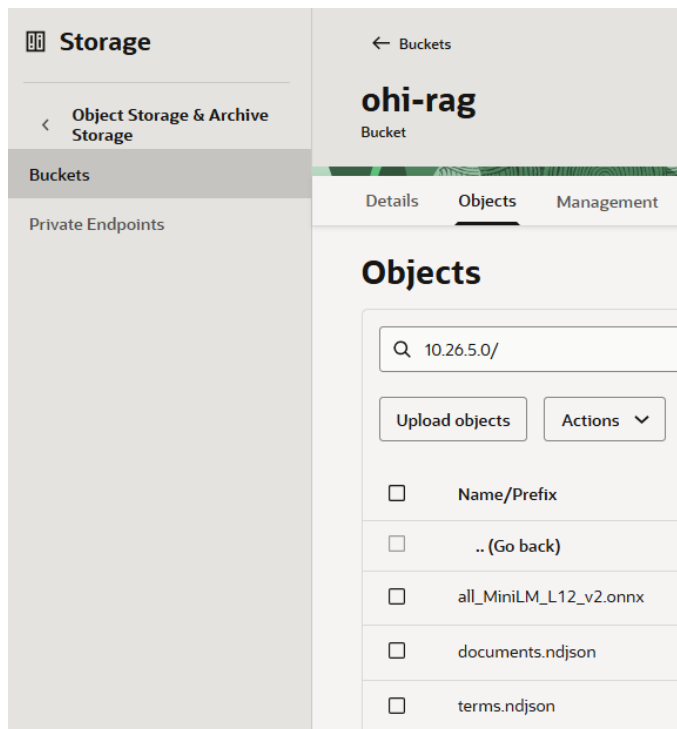
Create a shared directory object and grant access to both installation schemas:

```
create or replace directory OHI_RAG_DIR as '<DATABASE_HOST_PATH>';
grant read on directory OHI_RAG_DIR to OHI_LLM_OWNER;
grant read on directory OHI_RAG_DIR to OHI_RAG_OWNER;
Place the three input files in this server path before starting the installer.
```

Autonomous AI Database

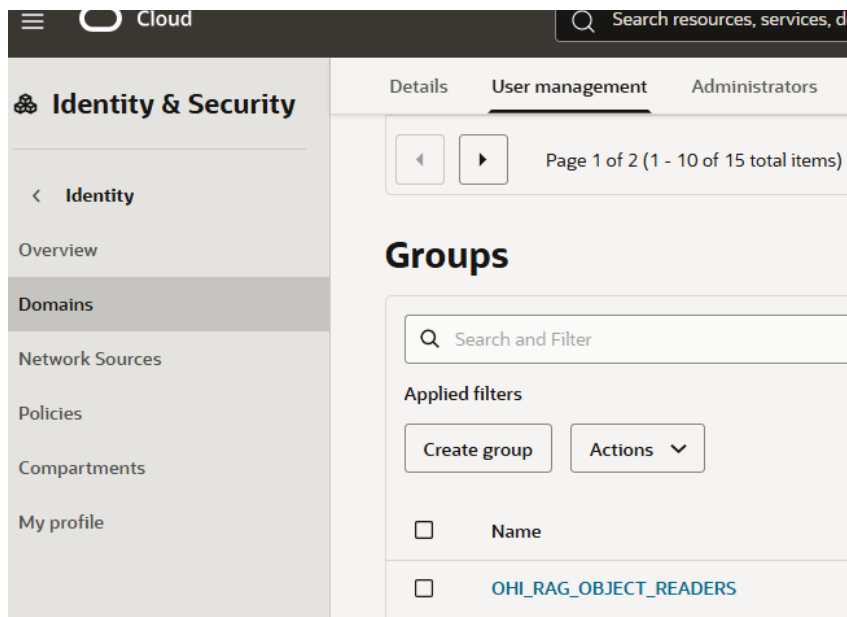
Create a bucket and a release-specific prefix, for example ohi-rag/10.XX.X.0/. Upload the three input files under that prefix. The Object Storage URI prefix supplied to the installer must end in "/". Do not use a pre-authenticated request (PAR) URL or embed credentials in the URI.

See the picture below for an example of such a bucket with relevant files:

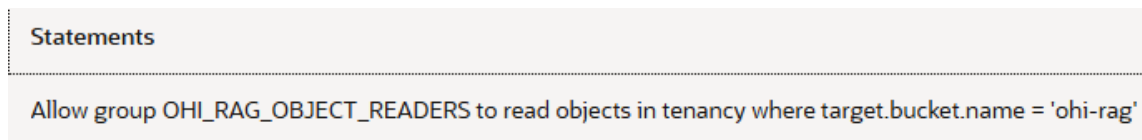


Grant the OCI principal(s) used by OHI_RAG_OCI_CRED and OHI_LLM_OCI_CRED read access only to this bucket or prefix. The credentials are schema-specific (but might of course reference the same OCI principal): the RAG data loaders use the OHI_RAG_OWNER credential (default name OHIRAG_OCI_CRED) and the model loader uses the OHI_LLM_OWNER (default name OHI_LLM_OCI_CRED) credential.

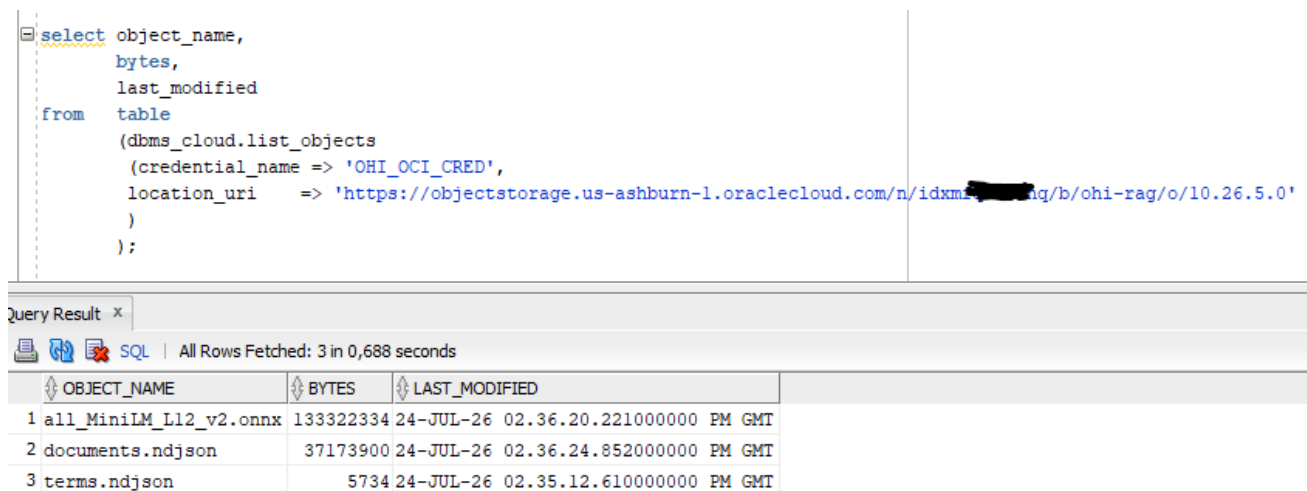
To provide the access you can create a group within your (default) domain under the User Management tab and grant this group access. Below, an example group is shown. Assign the relevant users/principals to the group.



After that, define a policy and add a statement that provides access to the bucket. An example statement is shown below, referencing the bucket name that was shown in the earlier screen print:



To test whether the bucket can be read you can use a similar SQL statement as below, using the RAG or LLM owner and the credential name you used (query it through `select credential_name, username, comments, enabled from user_credentials`):



Wallet aliases

Create Oracle wallet aliases and accompanying tnsnames aliases for the three OHI RAG database accounts, OHI_RAG_OWNER, OHI_RAG_USER, OHI_LLM_OWNER.

The installer prompts for these aliases and validates that each alias connects to the expected database account.

The following naming convention is recommended:

```
/@<environment>_ohi_rag_owner
/@<environment>_ohi_rag_user
/@<environment>_ohi_llm_owner
```

For example: /@prod01_ohi_rag_owner

Including the environment name in the alias helps distinguish credentials across environments and reduces the risk of accidentally connecting to the wrong database.

For information on creating wallet aliases, refer to section 6.1.3.2 Wallet for OS Account BATCH of the "Oracle Health Insurance Back Office Installation, Configuration and DBA Manual". The same procedure can be used for these three accounts.

When you use an autonomous database environment make sure you configure your SQLcl or sqlplus environment so the (mTLS) Cloud Wallet can be used.

For instructions please see:

<https://docs.oracle.com/en/cloud/paas/autonomous-database/serverless/adbsb/connect-tools.html>

When you create/add your TNS aliases in your tnsnames.ora you can choose whether to use a low, medium, high, tp or tpurgent service name. For the RAG and LLM owner connections you may choose a medium or high service, depending on how important it is to speed up the OHI RAG object installation step of the RAG database. It can take up to about two hours on slower environments to load and process the LLM ONNX file and the NDJSON. Of course, the speed is also dependent on the configuration of your ATP database (nr of OCPU's, etc.).

For the RAG user connection it is advised to use a tp or even tpurgent service connection.

Preparing the SQL scripts

Copy and unzip the contents of OHI_OLH_AI_ASSISTANT.zip from the directory \$OZG_BASE/conf/generic to a location where SQL*Plus or SQLcl can read the scripts and can use the created wallet entries through the created aliases. Start the full installation from that directory so that relative @@ includes work correctly.

Running the installation

Run the installer with SQL*Plus or SQLcl:

```
sqlplus -L /nolog @OHI_RAG_DB_INSTALL.sql
```

The script prompts for the following values:

- Wallet alias for OHI_RAG_OWNER including "/@".
- Wallet alias for OHI_RAG_USER including "/@".
- Wallet alias for OHI_LLM_OWNER including "/@".
- Optional ONNX file name (for example all_MiniLM_L12_v2.onnx).
- On Autonomous AI Database only:
Object Storage URI prefix and the two credential names. The defaults are OHI_RAG_OCI_CRED and OHI_LLM_OCI_CRED.

The ONNX file name can be omitted when the embedding model is already present in the database and only the RAG data and agent specification need to be updated.

To verify whether the embedding model is already installed, execute the following query using an account with DBA privileges:

```
SELECT m.owner,
       m.model_name
FROM   dba_mining_models m
WHERE  m.model_name = 'ALL_MINILM_L12_V2';
```

If the query returns a row for owner OHI_LLM_OWNER and model ALL_MINILM_L12_V2, the embedding model is already available in the database and the ONNX file name can be omitted.

Typically, this is the case if you update the NDJSON files because of a newer OHI release with newer Online Help information.

If no rows are returned, the installer requires the ONNX file from the configured directory or Object Storage prefix.

The picture below shows a snippet of the output when an autonomous database is used and the files are stored in an object store bucket:

```
Detecting input mode

1 row selected.

OCI Object Storage URI prefix (ending in /): https://objectstorage.us-ashburn-1.oraclecloud.com/n/ia[REDACTED]q/b/ohi-rag/o/10.26.5.0/
Object Storage credential for OHI_RAG_OWNER [OHI_RAG_OCI_CRED]: OHI_OCI_CRED
Object Storage credential for OHI_LLM_OWNER [OHI_LLM_OCI_CRED]: OHI_OCI_CRED
Input mode: OBJECT_STORE
```

Installer activities

OHI_RAG_DB_INSTALL.sql performs the following steps:

1. Validate the wallet aliases for OHI_RAG_OWNER, OHI_RAG_USER, and OHI_LLM_OWNER.
2. Install tables, Oracle Text settings, views, and packages in OHI_RAG_OWNER.
3. Create or update the scheduler job OHI_RAG_TRACE_PURGE_JOB.
4. Compile the Select AI Agent package from separate *.pks and *.pkb scripts.
5. Register the Select AI Agent definition through OHI_RAG_AGENT_DEFINITION.sql.
6. Optionally load the ONNX model in OHI_LLM_OWNER, depending on the optional file name.
7. Align model access from OHI_LLM_OWNER to OHI_RAG_OWNER.
8. Load documents.ndjson through install/OHI_RAG_LOAD_DOCUMENTS_FROM_NDJSON.sql.
9. Load terms.ndjson through install/OHI_RAG_LOAD_TERMS_FROM_NDJSON.sql.
10. Build the HNSW index.

This last step can take quite some time, depending on the database environment you use. Some runs took around 2 hours to complete.

Post-installation validation

Before executing the validation queries below, review the installation spool file generated by the installer and verify that:

- No 'ORA-', 'SP2-', 'PLS-' or 'ERROR' messages are present.
- All expected verification messages reporting successful execution are present.
- The spool file confirms successful completion of:
 - Input-file retrieval and RAG data loading.
 - Embedding model access verification or embedding model loading.

- Hybrid vector index creation.
- Agent deployment.

Proceed with the validation steps only after the installation has completed successfully and the spool file confirms that all installation phases have been completed.

Connect as OHI_RAG_OWNER before executing the validation queries. The validation is performed using the same schema that is used by the OHI RAG application and therefore verifies the installed functionality from an application perspective.

1. Verify that the embedding model can generate vectors and returns the expected vector dimensions:

```
SELECT vector_dimension_count(
        vector_embedding(OHI_LLM_OWNER.ALL_MINILM_L12_V2 USING 'smoke test' AS data)
    ) AS embedding_dimensions
FROM dual;
```

The query should return a single row with a positive dimension count. For the 'ALL_MINILM_L12_V2' model, the expected value is typically "384".

2. Verify that the hybrid vector index can be used to retrieve documents from the OHI knowledge base:

```
SELECT count(*) AS result_count
FROM json_table(
    dbms_hybrid_vector.search(
        json('{
            "hybrid_index_name": "OHI_DOC_HV_IND01",
            "vector": {
                "search_text": "Virtual Private Database",
                "search_mode": "DOCUMENT",
                "result_max": 5
            },
            "return": {
                "topN": 5,
                "values": ["rowid", "score"]
            }
        }')
    ),
    '$[*]'
    COLUMNS (score NUMBER PATH '$.score')
);
```

The query should return a value greater than zero, indicating that the vector index and indexed documentation content are available for retrieval.

3. Verify that the OCI credential, AI profile, network ACL configuration, and OCI Generative AI connectivity are functioning correctly:

```
SELECT DBMS_CLOUD_AI.GENERATE(
    prompt      => 'Reply with the word OK only.',
    profile_name => 'OHI_OCI_PROFILE',
    action      => 'chat'
) AS response
FROM dual;
```

The query should return a response containing the word: "OK".

Trace retention

Tracing for the OHI Online Help AI assistant can be configured through the OHI Back Office parameter "Tracelevel" in the Agent parameter group. Valid values are NONE, ERROR, and INFO. When set to NONE, no records are written to table OHI_RAG_TRACE in the 26ai PDB.

The installer creates or updates the scheduler job OHI_RAG_TRACE_PURGE_JOB in the 26ai PDB. The job runs daily at 02:00 and purges records from OHI_RAG_TRACE for which TRACE_MOMENT is more than 30 days old.

OHI Back Office integration

After the OHI RAG schemas have been created and the installation scripts have completed successfully, a database link must be created from the OHI Back Office schema to the OHI_RAG_USER account in the Oracle Database 26ai PDB hosting the OHI Online Help AI Assistant.

The OHI Back Office application uses this database link to access the functionality which is exposed through the OHI_RAG_USER schema. Create the database link in the OHI Back Office schema using the naming conventions and connectivity settings applicable to the target environment.

If the 26ai database is an autonomous database, you may want to specify the cloud wallet and connection details in your create database link command:

```
CREATE DATABASE LINK <26aiPDBname>_ohi_rag_user
  CONNECT TO ohi_rag_user IDENTIFIED BY "<pwd>"
  USING
  '(description=
    (retry_count ... <copy connection details from your tp or turgent definition
    as stored in the tnsnames.ora in the Cloud Wallet zip> ...
  )';
```

After the database link has been created, configure the OHI Back Office parameter Agent Database Link in the Agent parameter group with the name of the database link.

This parameter is used by the OHI Online Help AI Assistant integration to locate and access the remote OHI_RAG_USER schema.

After creating or modifying the database link and configuring the Agent Database Link parameter, execute the following command while connected as the OHI Back Office schema owner, or through a DBA account with sufficient privileges:

```
BEGIN
  sys_install_agent_pck.run;
END;
/
```

This step completes the integration between OHI Back Office and the OHI Online Help AI Assistant.

You can test whether the OHI Online Help AI assistant does work by starting the OHIJET user interface and use the AI icon in the upper right corner of the browser window:



Ask what the agent can do for you and verify if you get a decent reaction.

The icon can only be used if you are logged in in OHI BO with an account that has received access to the AI Chat Client module (module code SYS1160J).

Installing a higher major release of the Online Help data

After an initial installation of the OHI Online Help AI Assistant, the Online Help data needs to be refreshed for each major Release of OHI Back Office.

This is done by executing the following steps:

- Copy the new versions of the files documents.ndjson and terms.ndjson from the directory "\$OZG_BASE/help/nl_nl/Output/OHIHelp/RAG to the directory used by OHI_RAG_DIR or to your bucket in OCI.
- PREPARING THE SQL SCRIPTS (see above)
- RUNNING THE INSTALLATION (see above)
- POST-INSTALLATION VALIDATION (see above)

Cloning an OHI Back Office environment

When an OHI Back Office environment is cloned, the database link configuration should be reviewed. Depending on the cloning procedure and target environment, the database link may need to be:

- updated to point to the correct target PDB.
- recreated with the correct credentials.
- removed if the cloned environment should not have access to the OHI Online Help AI Assistant.

Failure to review the database link after cloning may result in the OHI Back Office environment connecting to an unintended AI Assistant environment.

Patching an OHI Back Office environment

If you patch an OHI Back Office environment to a higher major release, you should determine:

- if you need to update the Online Help data in the linked database 26ai PDB (see above), or
- if you need to point the database link to another database 26ai PDB with Online Help data of the same OHI major release, or
- if you want to disable the OHI Online Help AI Assistant by removing the database link

Copyright © 2026, Oracle and/or its affiliates. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

Some regulatory certifications or registrations to products or services referenced on this website are held by Cerner Corporation. Cerner Corporation is a wholly-owned subsidiary of Oracle. Cerner Corporation is an ONC-certified health IT developer and a registered medical device manufacturer in the United States and other jurisdictions worldwide.

This document may include some forward-looking content for illustrative purposes only. Some products and features discussed are indicative of the products and features of a prospective future launch in the United States only or elsewhere. Not all products and features discussed are currently offered for sale in the United States or elsewhere. Products and features of the actual offering may differ from those discussed in this document and may vary from country to country. Any timelines contained in this document are indicative only. Timelines and product features may depend on regulatory approvals or certification for individual products or features in the applicable country or region.