Oracle® APEX

Installation Guide

Release 24.2

G12910-03

August 2025

Oracle APEX Installation Guide, Release 24.2

G12910-03

Primary Author: John Godfrey

Contributors: Terri Jennings, David Bliss, Christina Cho, Hilary Farrell , Salim Hlayel, Christian Neumueller, Marc Sewtz, Jason Straub, Vlad Uvarov, Patrick Wolf

Contents

Preface

1 Changes in Release 24.2 for Oracle APEX Installation Guide

2 Oracle APEX Installation Requirements

3 APEX Installation Overview

4 Upgrading from a Previous APEX Release

5 Utilizing Multitenant Architecture

6 Installing and Configuring APEX and Oracle REST Data Services

A Automating the APEX Installation Process

B Maximizing Uptime During an APEX Upgrade

C APEX Installation Troubleshooting

Index

Preface

This guide explains how to install and configure Oracle APEX.

- Audience
- Related Documents
- Conventions
- Third-Party License Information

Audience

Oracle APEX Installation Guide is intended for anyone responsible for installing Oracle APEX.

To use this manual, you must have administrative privileges on the computer where you installed your Oracle database and familiarity with object-relational database management concepts.

Related Documents

For more information, see these Oracle resources:

- *Oracle APEX Release Notes*
- *Oracle APEX App Builder User's Guide*
- *Oracle APEX End User's Guide*
- *Oracle APEX Administration Guide*
- *Oracle APEX SQL Workshop Guide*
- *Oracle APEX API Reference*
- *Oracle Database Concepts*
- *Oracle Database Administrator's Guide*
- *Oracle Database SQL Language Reference*
- *Oracle SQLcl User's Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

Convention	Meaning
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Third-Party License Information

Oracle APEX contains third-party code. Please see the *Oracle APEX Licensing Information User Manual* for notices Oracle is required to provide.

Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the third-party software, and the terms contained in the following notices do not change those rights.

1

Changes in Release 24.2 for *Oracle APEX Installation Guide*

All content in *Oracle APEX Installation Guide* has been updated to reflect release 24.2 functionality.

Deprecated and Desupported Features

See Deprecated Features and Desupported Features in *Oracle APEX Release Notes*.

2

Oracle APEX Installation Requirements

Before installing Oracle APEX in a local self-managed installation you must verify your configuration meets the minimum installation requirements.

- Oracle Database Requirements
  Oracle APEX release 24.2 requires an Oracle Database release 19c or later. APEX runs on all database editions, including Enterprise Edition (EE), Standard Edition (SE), and Oracle Database Free. APEX can be installed in single-instance database and in Oracle Real Application Clusters (Oracle RAC) database.
- Browser Requirements
  Oracle APEX requires a JavaScript-enabled browser and supports the current and prior major release of Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge.
- Web Server Requirements
  Oracle APEX requires Oracle REST Data Services (ORDS) 23.3 or later.
- Disk Space Requirement
  Oracle APEX disk space requirements are described in this section.
- Oracle XML DB Requirement
  Oracle XML DB must be installed in the Oracle database that you want to use if you are installing a full development environment. If you are using a preconfigured database created either during an installation or by Database Configuration Assistant (DBCA), Oracle XML DB is already installed and configured.

2.1 Oracle Database Requirements

Oracle APEX release 24.2 requires an Oracle Database release 19c or later. APEX runs on all database editions, including Enterprise Edition (EE), Standard Edition (SE), and Oracle Database Free. APEX can be installed in single-instance database and in Oracle Real Application Clusters (Oracle RAC) database.

- Checking the Memory Settings of the Target Database
  Oracle APEX requires the system global area (SGA) to be at least 1200 MB and program global area (PGA) to be at least 300 MB.
- Checking the WORKAREA_SIZE_POLICY of the Target Database
  For the Oracle APEX installation or upgrade process, the `WORKAREA_SIZE_POLICY` session parameter must be set to `AUTO`.
- Installing APEX in Databases where Database Vault is Enabled
  Complete additional configuration steps to install Oracle APEX in databases where Oracle Database Vault is enabled.

2.1.1 Checking the Memory Settings of the Target Database

Oracle APEX requires the system global area (SGA) to be at least 1200 MB and program global area (PGA) to be at least 300 MB.

Databases typically have two parameters to control the memory allocation: `SGA_TARGET` and `PGA_AGGREGATE_TARGET`. For full details on Oracle Database memory configuration, consult the *Oracle Database Administrator's Guide* to find out how to configure manual memory parameters (for example, `SGA_TARGET`, `PGA_AGGREGATE_TARGET`, `SHARED_POOL_SIZE`).

To check the memory parameters of the target database:

1. Start the database:

   ```
   SQL> STARTUP
   ```

2. (Optional) Enter the following command to determine whether the system uses an initialization parameter file (`initsid.ora`) or a server parameter file (`spfiledbname.ora`):

   ```
   SQL> SHOW PARAMETER PFILE;
   ```

   This command displays the name and location of the server parameter file or the initialization parameter file.

3. Determine the current values of the memory parameters, then do one of the following:

   ```
   SQL> SHOW PARAMETER SGA_TARGET
   SQL> SHOW PARAMETER PGA_AGGREGATE_TARGET
   ```

   - If the value of `SGA_TARGET` is `0`, your database is using **automatic memory management**. Consult Using Automatic Memory Management in *Oracle Database Administrator's Guide* to learn how to configure an equivalent memory size using the `MEMORY_TARGET` parameter. Stop this procedure.
   - If the values of `SGA_TARGET` or `PGA_AGGREGATE_TARGET` are lower than the recommended sizes, then you must **increase** them depending on whether your database is using a server parameter file or the initialization parameter file. Proceed to the next step.

4. To determine whether your system uses a server parameter file or initialization parameter file:

   ```
   SQL> SHOW PARAMETER PFILE;
   ```

   - If the file begins with the prefix `sp` for **server parameter**, then issue:

     ```
     SQL> ALTER SYSTEM SET SGA_TARGET='1200M' SCOPE=spfile;
     SQL> ALTER SYSTEM SET PGA_AGGREGATE_TARGET='300M' SCOPE=spfile;
     ```

   - Otherwise, the system is using an **initialization parameter** file. Change the value of the `MEMORY_TARGET` parameter to at least 300 MB in the initialization parameter file (`init<sid>.ora`).

5. Shut down the database:

   ```
   SQL> SHUTDOWN
   ```

6. Restart the database:

   ```
   SQL> STARTUP
   ```

7. Confirm the new memory settings:

   ```
   SQL> SHOW PARAMETER SGA_TARGET
   SQL> SHOW PARAMETER PGA_AGGREGATE_TARGET
   ```

> **See Also**
>
> Using Automatic Memory Management in *Oracle Database Administrator's Guide*

2.1.2 Checking the WORKAREA_SIZE_POLICY of the Target Database

For the Oracle APEX installation or upgrade process, the `WORKAREA_SIZE_POLICY` session parameter must be set to `AUTO`.

To check the `WORKAREA_SIZE_POLICY` of the target database:

1. Start SQLcl and connect to the database as `SYS` specifying the `SYSDBA` role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

2. Check the current value of the `WORKAREA_SIZE_POLICY` parameter:

   ```
   SQL> SHOW PARAMETER WORKAREA_SIZE_POLICY
   ```

3. If the value of the parameter is `MANUAL`, change it to `AUTO` for the current database session. For example:

   ```
   SQL> ALTER SESSION SET WORKAREA_SIZE_POLICY = AUTO;
   ```

4. Within the same database session, perform the installation or upgrade of Oracle APEX.

Note

If you are installing Oracle APEX in a CDB, `WORKAREA_SIZE_POLICY` must be set system-wide. For example:

```
SQL> ALTER SYSTEM SET WORKAREA_SIZE_POLICY=AUTO SCOPE=BOTH;
```

Then, if needed, change it back to `MANUAL` after Oracle APEX installation or upgrade.

See Also

WORKAREA_SIZE_POLICY in *Oracle Database Reference*

2.1.3 Installing APEX in Databases where Database Vault is Enabled

Complete additional configuration steps to install Oracle APEX in databases where Oracle Database Vault is enabled.

To begin, see Integrating Oracle Database Vault with Oracle APEX in *Oracle Database Vault Administrator's Guide*.

You must also complete additional configuration steps for some APEX features, including workspace provisioning and SQL Workshop.

The Database Vault security controls protect application data from unauthorized access and help you to comply with privacy and regulatory requirements. Database Vault is a licensable option of Oracle Database Enterprise Edition. Its purpose is to mitigate the potential impact of privileged account abuse, misuse, insider and external threats, and human error on your sensitive data.

2.2 Browser Requirements

Oracle APEX requires a JavaScript-enabled browser and supports the current and prior major release of Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge.

2.3 Web Server Requirements

Oracle APEX requires Oracle REST Data Services (ORDS) 23.3 or later.

Oracle REST Data Services (ORDS) is Java-based web server. Oracle REST Data Services features the ability to emit RESTful web services, offers improved file upload capability, and is certified with Oracle WebLogic Server and Apache Tomcat.

> **Tip**
>
> APEX-based REST Services were desupported in release 22.1. Oracle REST Data Services (ORDS) release 21.4.2 and newer now ship with migration scripts that enable you to upgrade any remaining APEX-based REST Services to ORDS-based Services. To learn more, see Migration of Oracle APEX RESTful Service Modules in Oracle REST Data Services Release Notes.

2.4 Disk Space Requirement

Oracle APEX disk space requirements are described in this section.

APEX disk space requirements are as follows:

- Free space for APEX software files on the file system: 670 MB if using English only download (`apex_24.2_en.zip`) and 1,178 MB if using full download (`apex_24.2.zip`).
- Free space in APEX tablespace: 260 MB
- Free space in `SYSTEM` tablespace: 128 MB
- Free space in APEX tablespace for each additional language (other than English) installed: 60 MB

2.5 Oracle XML DB Requirement

Oracle XML DB must be installed in the Oracle database that you want to use if you are installing a full development environment. If you are using a preconfigured database created either during an installation or by Database Configuration Assistant (DBCA), Oracle XML DB is already installed and configured.

> **Tip**
>
> The installer does a prerequisite check for Oracle XML DB and will exit if it is not installed.

> **Tip**
>
> The installation of Oracle XML DB creates the user ANONYMOUS. In order for Oracle APEX workspace provisioning to work properly, the ANONYMOUS user must not be dropped from the database.

> **Tip**
>
> For more information about manually adding Oracle XML DB to an existing database, see Administration of Oracle XML DB in *Oracle XML DB Developer's Guide*

3

APEX Installation Overview

Oracle APEX Installation Guide describes how to install Oracle APEX in a self-managed (local) database.

How you sign in and access APEX depends upon your user role and where APEX resides. APEX may reside in a local self-managed Oracle or in an Oracle Cloud Service.

- About APEX Architecture
  Oracle APEX uses a simple architecture in which pages are dynamically generated using metadata stored within the Oracle database.
- About Accessing APEX in Oracle Cloud
  Learn about accessing Oracle APEX in Oracle Cloud.
- Understanding the Installation Process
  Installing Oracle APEX is a multiple step process. You follow the same instructions for new or upgrade installations.
- About the APEX Runtime Environment
  Learn about the Oracle APEX runtime environment.
- About APEX Database Accounts
  Oracle APEX creates the following database accounts at installation.

See Also

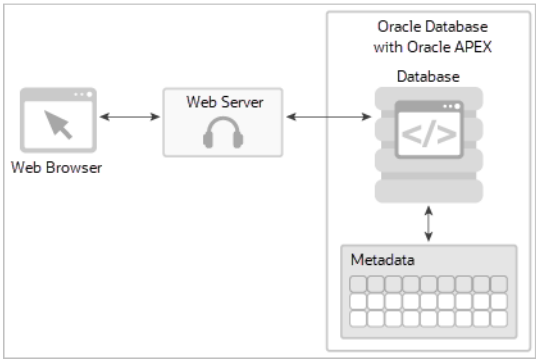
Upgrading from a Previous APEX Release

3.1 About APEX Architecture

Oracle APEX uses a simple architecture in which pages are dynamically generated using metadata stored within the Oracle database.

About the APEX Architecture

The APEX architecture consists of a web browser, Oracle REST Data Services (the web server), and an Oracle database containing APEX. The major advantage of this architecture is the separation of the mid-tier and the database tier.



The web server, Oracle REST Data Services, functions as a communications broker between the web browser and the APEX objects in the Oracle database by mapping browser requests into database stored procedure calls.

Once fully installed, a Uniform Resource Locator (URL) is defined for both developers and end users to access APEX. Users require only a web browser and the required URL. No additional client software is required.

About Oracle REST Data Services

Oracle REST Data Services (ORDS) (formerly known as Oracle Application Express Listener) is a J2EE application which communicates with the Oracle database by mapping browser requests to the APEX engine database over a SQL*Net connection.

Oracle REST Data Services is fully supported when deployed in a standalone mode as well as when deployed into Oracle WebLogic Server or Apache Tomcat application servers.

Note

There are licensing costs associated with Oracle WebLogic Server.

See Also

- Web Server Requirements
- Installing and Configuring APEX and Oracle REST Data Services
- Introduction to Oracle REST Data Services in *Oracle REST Data Services Developer's Guide*
- Installing and Configuring Oracle REST Data Services in *Oracle REST Data Services Installation and Configuration Guide*

3.2 About Accessing APEX in Oracle Cloud

Learn about accessing Oracle APEX in Oracle Cloud.

APEX may reside in a local self-managed Oracle database or in a hosted environment (such as an Oracle Cloud service). The sign in process differs depending on where APEX resides.

In Oracle Cloud, APEX is installed, enabled, and ready to use in the following products:

- Oracle APEX Application Development (APEX Service)
- Oracle Autonomous Transaction Processing
- Oracle Autonomous Data Warehouse

APEX is available in Exadata Cloud Service and Database Cloud Service in Oracle Cloud Infrastructure. However, you need to manually customize your databases to install and enable APEX by following the self-managed database installation process or using cloud tooling such as Terraform.

Related Topics

- Get an Environment
- Welcome to Oracle APEX Application Development Service in *Getting Started with Oracle APEX Application Development*
- Creating Applications with Oracle APEX on Autonomous Database in *Using Oracle Autonomous Database Serverless*

3.3 Understanding the Installation Process

Installing Oracle APEX is a multiple step process. You follow the same instructions for new or upgrade installations.

- About Planning Your Installation
  Learn about the steps needed to install Oracle APEX.
- About Patch Sets
  Patch sets provide bug fixes only. Oracle may release one or several patch set bundles in between major product releases.
- About the Installation Scripts
  You can install Oracle APEX or update from previous release using the same installation procedure and the installation scripts.
- About Accessing APEX
  You access the Oracle APEX development environment, by signing in to a shared work area called a workspace.
- Requesting a Workspace from the Sign In Dialog
  Request a workspace from the Sign In dialog.
- Resetting Your Password from the Sign In Page
  Reset your password by clicking a link on Oracle APEX Sign In page.
- Recovering Your Workspace Name
  Recover your workspace name from the Oracle APEX Sign In page.

> **See Also**
>
> Upgrading from a Previous APEX Release

3.3.1 About Planning Your Installation

Learn about the steps needed to install Oracle APEX.

Oracle recommends you take the time to carefully plan your installation.

Installing Oracle APEX involves the following steps:

1. **Decide on a Full or Runtime Environment -** Determine whether to install a full development environment or runtime environment. A **full development** environment provides complete access to the App Builder development environment to develop applications. A **runtime environment** is the appropriate choice for production implementations in which you want to run applications that cannot be modified.

   See About the APEX Runtime Environment.

2. **Verify installation requirements-** Before installing, verify your system meets the minimum requirements.

   See Oracle APEX Installation Requirements .

3. **Install the software** - Install or upgrade Oracle APEX by downloading a ZIP file from the Oracle APEX download page and then downloading and installing Oracle REST Data Services (ORDS) as described in Installing and Configuring APEX and Oracle REST Data Services.

See Also

Upgrading from a Previous APEX Release

3.3.2 About Patch Sets

Patch sets provide bug fixes only. Oracle may release one or several patch set bundles in between major product releases.

Patch sets are a mechanism for delivering fully tested and integrated product fixes. Patch sets provide bug fixes only. Patch sets typically do not include new functionality and they do not require certification on the target system. Patch sets include all of the libraries that have been rebuilt to implement the bug fixes in the set. All of the fixes in the patch set have been tested and are certified to work with each other.

In between major product releases, Oracle may release one or several patch set bundles. Visit Oracle APEX Downloads to learn about the contents of each patch set.

In fully managed Cloud databases, such as an Autonomous Database or Oracle APEX Application Development (APEX Service), Oracle automatically applies patch set bundles. For all other environments, customers can download patch set bundles from My Oracle Support.

To review the history of APEX patch sets installed in your database, query the APEX_PATCHES dictionary view:

```
select patch_number, patch_version, installed_on
  from APEX_PATCHES
 order by installed_on;
```

See Also

- Upgrading from a Previous APEX Release
- Oracle APEX Downloads
- My Oracle Support

3.3.3 About the Installation Scripts

You can install Oracle APEX or update from previous release using the same installation procedure and the installation scripts.

The installation script checks for the latest existing Oracle APEX schema and automatically copies the instance metadata, workspaces, and applications from the previous schema into the current schema. The original schema associated with the previous release is left completely unaltered. Following best practices, Oracle recommends that you create new tablespaces for a new release of Oracle APEX and follow the appropriate installation instructions as outlined in this document.

3.3.4 About Accessing APEX

You access the Oracle APEX development environment, by signing in to a shared work area called a workspace.

How you sign in and access APEX depends upon your user role.

A workspace enables multiple users to work within the same APEX installation while keeping their objects, data, and applications private. Each workspace has a unique ID and name. An instance administrator can create a workspace manually within APEX Administration Services or have users submit requests. APEX Administration Services is a separate application for managing an entire APEX instance.

Users are divided into four primary roles:

- **Instance administrators** are *superusers* that manage an entire hosted instance using a separate application called APEX Administration Services. Instance administrators manage workspace provisioning, configure features and instance settings, and manage security.
- **Workspace administrators** can perform administrator tasks specific to a workspace such as configuring workspace preferences, managing user accounts, monitoring workspace activity, and viewing log files.
- **Developers** are users who sign in to a workspace and create and edit applications.
- **End users** can only run existing applications.

If you are a developer, an administrator must grant you access to shared work area called a workspace. If you are an Instance administrator, you must sign in to APEX Administration Services, determine whether to specify a provisioning mode, create a workspace, and then sign in to that workspace.

About Specifying a Provisioning Mode

The Instance administrator determines how the process of provisioning (or creating) a workspace works for a specific APEX instance. To determine how provisioning works, an Instance Administrator selects a Provisioning Methods on the Instance Settings page:

- **Manual** - An Instance administrator creates new workspaces and notifies the Workspace administrator regarding the Sign In credentials.
- **Request** - Users request a workspace. Once an administrator approves the request, the user receives an email containing an email verification link. After the user clicks the email verification link, the workspace is created.
- **Automatic** - Works similar to Request except requests are automatically approved with no administrator review required

See Also

About Specifying How Workspaces Are Created and Selecting a Workspace Provisioning Mode in *Oracle APEX Administration Guide*

About Creating Workspaces and Users

Before you can develop or install applications, an administrator must create a workspace and add APEX users. To learn more contact your administrator, or see Creating a Workspace and Adding APEX Users.

See Also

- Creating Workspaces in Administration Services in *Oracle APEX Administration Guide*
- Making a Service Request in *Oracle APEX Administration Guide*
- Managing Requests in *Oracle APEX Administration Guide*

3.3.5 Requesting a Workspace from the Sign In Dialog

Request a workspace from the Sign In dialog.

Your administrator determines how you request a new workspace. If your administrator has set Provisioning Method to either **Request** or **Automatic** and has configured email, you can request a workspace on the Sign In dialog.

To request a workspace from the Sign In dialog:

1. Navigate to the Oracle APEX Sign in dialog.
2. Under Sign In, click **Request a Workspace**.

   The Request a Workspace Wizard appears.
3. For Identification:
   a. First Name - Enter your first name.
   b. Last Name - Enter your last name.
   c. Email - Enter the email address. A link to activate your workspace will be sent to this email address.
   d. Workspace - Enter a workspace name that name uniquely identifies your development environment.

   e. Click **Next**.
4. If defined, review and accept the service agreement and click **Next**.
5. Verify your request and click **Submit Request**.

   Once you complete the Identification form, the following events occur:

   a. You will receive an email containing a verification link.
   b. When you click the verification link, the workspace is created.
   c. You will receive another email containing Sign In credentials (that is, the workspace name, username, and password).

Once you complete the Identification form, the following events occur:

1. You will receive an email containing a verification link.
2. When you click the verification link, the workspace is created.
3. You will then receive another email containing Sign In credentials (that is, the workspace name, username, and password).

> ⓘ **See Also**
>
> About Specifying How Workspaces Are Created in *Oracle APEX Administration Guide*

3.3.6 Resetting Your Password from the Sign In Page

Reset your password by clicking a link on Oracle APEX Sign In page.

> ✓ **Tip**
>
> To reset your password from the Sign In page, you must provide your email address and the workspace name.

To reset your password from the Sign In Page:

1. In a web browser, navigate to the Oracle APEX Sign In page.

   The Sign In page appears.
2. Under **Sign In**, click **Reset Password**.
3. In the Reset Password form, enter your email address, workspace name, and click **Reset Password**.

   You will receive an email confirming your workspace name and username and containing a **Reset Password URL** link.
4. In the email, click the **Reset Password URL** link.
5. In the Change Password form:

   a. New Password - Enter your new password.

> **Tip**
>
> Passwords are case sensitive.

  b. Confirm Password - Enter your new password again.
  c. Click **Apply Changes**.

> **Tip**
>
> You can also reset your password within Oracle APEX. See Changing Your Profile or Password in *Oracle APEX App Builder User's Guide*

3.3.7 Recovering Your Workspace Name

Recover your workspace name from the Oracle APEX Sign In page.

If you cannot remember your workspace name, you can request a list of all workspace names associated with your email address.

To find your workspace name:

1. In a web browser, navigate to the Oracle APEX Sign In page.
2. On the Sign In page, click **Reset Password**.
3. Click **Find My Workspace**.
4. Enter your email address and click **Find Workspace**.

   You will receive an email listing all workspaces associated with the email address you provided.

3.4 About the APEX Runtime Environment

Learn about the Oracle APEX runtime environment.

As with any software development life cycle, Oracle strongly recommends that you have different environments for development, testing/QA, and production. For testing and production instances, APEX supports the ability to install just a runtime version of APEX. This runtime environment minimizes the installed footprint and privileges and improves application security since in a runtime instance developers cannot inadvertently update a production application.

An APEX runtime environment enables you to run applications, but it does not provide a Web interface for administration. A runtime environment only includes the packages necessary to run your application, making it a more hardened environment. You administer the APEX runtime environment using the `APEX_INSTANCE_ADMIN` API and SQL Developer or SQLcl.

Scripts are provided to remove or add the developer interface from an existing instance.

To ensure the security and performance of your development environment, APEX runtime environment is not available in Oracle Autonomous Database. To harden your testing and production instances, you can use the `RESTRICT_DEV_HEADER` instance parameter, which disables access to APEX App Builder and APEX Administration Services.

Related Topics

- Converting Between Runtime and Full Development Environments
  Learn about converting between runtime and full development environments.
- About the Advantages of the Oracle APEX Runtime Environment in *Oracle APEX App Builder User's Guide*
- About the Advantages of the Oracle APEX Runtime Environment in *Oracle APEX API Reference*

3.5 About APEX Database Accounts

Oracle APEX creates the following database accounts at installation.

- `APEX_240200` - This schema-only account owns the APEX platform schema and metadata. This account is always locked.
- `FLOWS_FILES` - This schema-only account owns the APEX-uploaded files that are **not** stored in workspace schemas. This account is always locked.
- `APEX_PUBLIC_USER` - This minimally privileged account is used for APEX configuration with Oracle REST Data Services.
- `APEX_PUBLIC_ROUTER` - This minimally privileged account is used for an ORDS REST module to resolve friendly URLs.

If you configured RESTful Web services, APEX also creates the following additional accounts:

- `APEX_REST_PUBLIC_USER` - This account is used when invoking RESTful Services definitions stored in APEX.
- `APEX_LISTENER` - This account is used to query RESTful Services definitions stored in APEX.

If you are upgrading from a previous release, all of the above accounts may already exist **except** `APEX_240200`.

4

Upgrading from a Previous APEX Release

Upgrading Oracle APEX creates new database objects in a new schema and migrates the application metadata to the new release.

If you have APEX release 24.1 or earlier, following any of the installation scenarios in this guide upgrades your APEX instance to the current release, creates APEX 24.2 database objects in a new schema, and migrates the application metadata to the new release.

- About Release Numbering Conventions
  New releases of Oracle APEX correlate to the calendar year.
- Sample Upgrade Scenarios
  Review common upgrade scenarios including upgrading from a prior release and upgrading when an Oracle Database release includes Oracle APEX.
- Viewing the APEX Release Number
  View your Oracle APEX release number on the Workspace home page or on the About APEX page.
- Viewing the Oracle REST Data Services Release Number
  View the Oracle REST Data Services release number on the About Oracle APEX page.
- About Installing an APEX Release Included with the Oracle Database
  Learn about which Oracle Database releases include Oracle APEX and the importance of updating to the lastest APEX release.
- About Upgrading Existing Applications
  Installing a new release of Oracle APEX, updates existing applications to the latest release, but does not alter application user interface or application components.
- About Testing Requirements
  Determining the appropriate amount of regression testing when upgrading Oracle APEX depends upon the complexity, size, and number of applications you are upgrading.
- About Cleaning Up Your Environment
  Following the successful upgrade of all of the environments to the latest release of Oracle APEX, you should clean up the environments.
- About Reverting to a Previous Release
  You can revert to a previous release of Oracle APEX.

ⓘ See Also

- Understanding the Installation Process
- Maximizing Uptime During an APEX Upgrade

4.1 About Release Numbering Conventions

New releases of Oracle APEX correlate to the calendar year.

In 2018 and starting with release 18.1 and 18.2, APEX introduced correlating the release number to the calendar year.

In addition, APEX now only offers full releases and no longer provides patch set releases (such as 5.1.1). Eliminating patch set releases reduces downtime when updating existing installations. APEX architecture also enables developers to revert releases if necessary.

Patch set exceptions (PSEs) may still be delivered for major defects. To learn more about PSEs, visit the Oracle APEX 24.2 Known Issues page or the Prior Release Archives for earlier releases.

4.2 Sample Upgrade Scenarios

Review common upgrade scenarios including upgrading from a prior release and upgrading when an Oracle Database release includes Oracle APEX.

Table 4-1 lists common upgrade scenarios.

Table 4-1 Sample Upgrade Scenarios

Upgrade Scenarios	Action
Upgrade from a prior Oracle APEX release.	Download the most recent ZIP file from the Oracle APEX download page and run a script to upgrade to the latest release. For details, see Installing APEX .
You install an Oracle Database which includes Oracle APEX.	Download the most recent ZIP file from the Oracle APEX download page and run a script to upgrade to the latest release. For details, see Installing APEX .

> ⚠ **Caution**
>
> When upgrading from APEX 18.1 to APEX 24.2, upgrade to APEX 24.1 first, and then to APEX 24.2. For more information, review Oracle APEX 24.2 Known Issues.

Related Topics

- Downloading and Installing APEX
  Learn about downloading and installing Oracle APEX.
- *Oracle APEX Release Notes*

4.3 Viewing the APEX Release Number

View your Oracle APEX release number on the Workspace home page or on the About APEX page.

You can view the APEX release number on the Workspace home page or on the About APEX page:

- Workspace home page:
  - Sign in to APEX.

    On the Workspace home page, the current release number displays in the bottom right corner.
- About APEX page:
  - Sign in to APEX.
  - Click the **Help** menu in the upper right and select **About**.

    On the About APEX page, the release number appears next to **Product Build**.

4.4 Viewing the Oracle REST Data Services Release Number

View the Oracle REST Data Services release number on the About Oracle APEX page.

Oracle APEX requires access to the web server, Oracle REST Data Services (ORDS) 23.3 or later.

To view the Oracle REST Data Services release number:

1. Sign in to Oracle APEX.
2. Click the **Help** menu in the upper right and select **About**.
3. Under the **CGI Environment** section, find `APEX_LISTENER_VERSION`.

4.5 About Installing an APEX Release Included with the Oracle Database

Learn about which Oracle Database releases include Oracle APEX and the importance of updating to the lastest APEX release.

APEX is included with the following Oracle Database releases:

- Oracle Database 19*c* - Oracle Application Express release 18.1.
- Oracle Database 18*c* - Oracle Application Express release 5.1.
- Oracle Database 12*c* Release 2 (12.2)- Oracle Application Express release 5.0.
- Oracle Database 12*c* Release 1 (12.1) - Oracle Application Express release 4.2.
- Oracle Database 11*g* Release 2 (11.2) - Oracle Application Express release 3.2.
- Oracle Database 11*g* Release 1 (11.1) - Oracle Application Express release 3.0.

Since Oracle Database releases less frequently than APEX, Oracle recommends updating to the latest APEX release available. To learn more, see Downloading and Installing APEX.

> **Note**
>
> If upgrading APEX from a release that ships with the database, do not alter any APEX files in the Oracle home directory (for example, `/u01/app/oracle/product/18.0.0/dbhome_1/apex`) .

4.6 About Upgrading Existing Applications

Installing a new release of Oracle APEX, updates existing applications to the latest release, but does not alter application user interface or application components.

Once you upgrade an Oracle APEX instance from a previous release, existing applications will work without modification. However, to keep applications maintainable, up-to-date, and to leverage new functionality, developers should perform the steps outlined in Upgrading APEX Applications in *Oracle APEX App Builder User’s Guide*.

4.7 About Testing Requirements

Determining the appropriate amount of regression testing when upgrading Oracle APEX depends upon the complexity, size, and number of applications you are upgrading.

You should include the majority of complex pages, particularly those that incorporate significant JavaScript or extensive PL/SQL computations or processes. Developers should ensure pages which they manually update based on the Upgrade Application or Advisor are also included in regression tests. Not all remaining pages have to be included in regression testing. Oracle recommends you include a good representation of different page types includes reports, charts, and forms. An application should always be included in regression testing if its compatibility mode was modified post-upgrade.

While regression testing of upgraded applications is imperative to minimize risk of disrupting the end users, it is important that testing is not drawn out for an extended period. As a general rule:

- Step 1: Upgrade your development environment first. Allow developers to review the applications and make initial updates as needed.
- Step 2: Upgrade your QA/Test environment.
- Step 3: Upgrade applications from development are built into this environment.
- Step 4: Upgrade your production environment.
- Step 5: Build upgraded applications into this environment.

4.8 About Cleaning Up Your Environment

Following the successful upgrade of all of the environments to the latest release of Oracle APEX, you should clean up the environments.

Once you start developing with the newer release, the Oracle APEX schema associated with the prior release can be deleted. If you installed the prior release into a separate tablespace, you can simply drop the specific tablespace. Oracle recommends leaving the older Oracle APEX schema(s) for a few weeks and then remove them from the development, test, and production environments. This cleanup process releases disk space and ensures that no one accesses an outdated schema using tools such as SQL Developer or SQLcl.

4.9 About Reverting to a Previous Release

You can revert to a previous release of Oracle APEX.

Because Oracle APEX creates a new schema for each major release, reverting back to a prior release is a relatively simple process. If you revert to a prior release, any modifications made in the current Oracle APEX instance are lost. The main task is to switch the public synonyms and grants to point at the previous schema instead of the new schema.

See Also

Reverting to a Previous Release After a Failed Upgrade Installation

5

Utilizing Multitenant Architecture

Learn about installation choices and different scenarios associated with copying and moving pluggable databases introduced by the multitenant architecture with respect to Oracle APEX.

- Understanding the Installation Choices
  Learn about the installation choices in Oracle APEX.
- Installing APEX into an Application Container
  Learn about the Application Container that stores data and metadata for Oracle APEX application back ends.
- Installing APEX into Different PDBs
  You can install different versions of Oracle APEX into different PDBs.
- Plugging in a PDB When APEX Is Installed in the Root Container
  Learn about scenarios in which the target database has Oracle APEX installed into the root container, `CDB$ROOT` - the default installation option.
- Plugging in a PDB When APEX Is Not in the Root Container of the Target CDB
  The scenarios in this section describe when Oracle APEX is not installed in the root container, `CDB$ROOT`, by explicitly removing it as described in "Uninstalling APEX from a CDB."

5.1 Understanding the Installation Choices

Learn about the installation choices in Oracle APEX.

The multitenant database architecture has a multitenant container database (CDB) that includes a root container, `CDB$ROOT`, a seed database, `PDB$SEED`, and multiple pluggable databases (PDBs). Each pluggable database is equivalent to a separate database instance in Oracle Database release 11*g*. The root container, `CDB$ROOT`, holds common objects that are accessible to every PDB utilizing metadata links or object links. The seed database, `PDB$SEED`, is used when creating a new PDB to seed the new database. The key benefit of the multitenant architecture is that the database resources, such as CPU and memory, can be shared across all of the PDBs. This architecture also enables many databases to be treated as one for tasks such as upgrades or patches, and backups.

> **Tip**
>
> Oracle recommends installing APEX in individual PDBs for the majority of use cases, except for hosting companies or installations where all PDBs utilize APEX and they all need to run the exact same release and patch set of APEX.
>
> When APEX is installed in the CDB, you still install and run APEX applications from individual PDBs. Running APEX applications or accessing an APEX development environment in the `CDB$ROOT` container is **not** supported.

See Also

Installing APEX into Different PDBs

5.2 Installing APEX into an Application Container

Learn about the Application Container that stores data and metadata for Oracle APEX application back ends.

- About Application Containers
  An application container is a CDB component that stores data and metadata for application backends.
- Creating an Application Container
  To create a PDB within a CDB as an application container, you use the `AS APPLICATION CONTAINER` clause of the create PDB command.
- Installing or Upgrading APEX in an Application Container
- Verifying the Application Container Installation
  Verify the application container by inpsecting the log file for `ORA-` or `PLS-` errors and compiling invalid objects.
- Creating an Application Seed
  An application seed is used to provision application PDBs with the Oracle APEX application root's applications pre-installed.
- Creating an Application PDB from the Application Root Seed
  An application PDB is created by issuing the `CREATE PLUGGABLE DATABASE` statement from the application root.
- Configure HTTP Access to the Application PDB
  Configure a new application PDB for HTTP access.

5.2.1 About Application Containers

An application container is a CDB component that stores data and metadata for application backends.

Oracle APEX can be installed into an application container using the `apxappcon.sql` script. An application container consists of an application root where the application is defined and one or more PDBs that share data and metadata about the application from the application root. You can have multiple application containers within a CDB and each container can have a different version of Oracle APEX.

Patching or upgrading Oracle APEX in an application container is simplified, because these actions are done against the application root. When an application PDB wishes to uptake the patch or upgraded version, it simply syncs with the application root. Oracle APEX continues to run in the application PDB at the existing version until the application PDB syncs with the application root.

5.2.2 Creating an Application Container

To create a PDB within a CDB as an application container, you use the `AS APPLICATION CONTAINER` clause of the create PDB command.

To create Application Container:

1. Use the `AS APPLICATION CONTAINER` clause of the `CREATE PLUGGABLE DATABASE` statement to create an application container.
2. Open the application container.

   For Example:

   ```
   CREATE PLUGGABLE DATABASE apex_approot1 AS APPLICATION CONTAINER admin
   user admin IDENTIFIED
         BY <admin_password> FILE_NAME_CONVERT=('pdbseed','apex_approot1');
   ALTER PLUGGABLE DATABASE apex_approot1 open;
   ```

Note

`apex_approot1` and the `admin user` in the previous example can be any valid ORACLE identifier.

5.2.3 Installing or Upgrading APEX in an Application Container

To install or upgrade Oracle APEX in an Application Container:

1. Connect to Application Container.
2. Run `apxappcon.sql`.

   `apxappcon.sql` installs APEX as an application named `APEX` into the application root.

   The script takes the exact same first four arguments as the `apexins.sql script`, with the addition of a fifth parameter which is the password to use for the `APEX_PUBLIC_USER` password. In an upgrade installation, the fifth argument is ignored because the `APEX_PUBLIC_USER` database user will already exist.

   For example:

   ```
   ALTER SESSION SET CONTAINER = apex_approot1;

    @apxappcon.sql SYSAUX SYSAUX TEMP /i/ P@ssw0rd!
   ```

5.2.4 Verifying the Application Container Installation

Verify the application container by inpsecting the log file for `ORA-` or `PLS-` errors and compiling invalid objects.

To verify the Application Container installation:

1. Manually inspect the installation log file for `ORA-` or `PLS-` errors.
2. Compile invalid objects by running the following command:

   For example:

   ```
   ALTER SESSION SET CONTAINER=apex_approot1;

   begin
       sys.dbms_utility.compile_schema( 'APEX_240200', false );
       sys.dbms_utility.compile_schema( 'FLOWS_FILES', false );
   ```

```
end;
/
```

3. Query `dba_applications` and `dba_app_errors`.

```
SQL> select app_name, app_version, app_status from dba_applications where
app_name = 'APEX';


APP_NAME                       APP_VERSION                    APP_STATUS

------------------------------ ------------------------------ ------------

APEX                           24.2                           NORMAL

SQL> select app_name, app_statement, errornum, errormsg from
dba_app_errors where app_name = 'APEX';


no rows selected
```

5.2.5 Creating an Application Seed

An application seed is used to provision application PDBs with the Oracle APEX application root's applications pre-installed.

To create an Application Seed:

1. Connect to `CDB$ROOT` as `sysdba`.
2. Alter session and set container to the application root.
3. Use the `AS SEED` clause of the `CREATE PLUGGABLE DATABASE` statement to create an application seed.
4. Sync the APEX application with the application seed.
5. Compile invalid objects.
6. Open the application seed in read only mode.

   For example:

```
ALTER SESSION SET CONTAINER=apex_approot1;

CREATE PLUGGABLE DATABASE as seed admin user admin identified by
<admin_password> file_name_convert=('pdbseed','apex_approot1_seed');

ALTER PLUGGABLE DATABASE apex_approot1$seed open;

ALTER SESSION SET CONTAINER=apex_approot1$seed;

ALTER PLUGGABLE DATABASE application APEX sync;

begin
    sys.dbms_utility.compile_schema( 'APEX_240200', false );
    sys.dbms_utility.compile_schema( 'FLOWS_FILES', false );
end;
```

```
/

ALTER PLUGGABLE DATABASE close immediate;

ALTER PLUGGABLE DATABASE open read only;
```

> **Note**
>
> `apex_approot1` and the `admin user` in the previous example can be any valid ORACLE identifier.

5.2.6 Creating an Application PDB from the Application Root Seed

An application PDB is created by issuing the `CREATE PLUGGABLE DATABASE` statement from the application root.

The `PLUGGABLE DATABASE` is created from the application container seed so the APEX application is already installed and ready for configuration.

To create an Application PDB from the Application Root Seed:

1. Connect to `CDB$ROOT` as `sysdba`.
2. Alter session and set container to the application root.
3. Use the `CREATE PLUGGABLE DATABASE` command to create a PDB from the application seed.

   For example:

   ```
   ALTER SESSION SET CONTAINER=apex_approot1;

   CREATE PLUGGABLE DATABASE apex_pdb1 admin user admin identified by <admin
   password> file_name_convert=('apex_approot1_seed','apex_pdb1');

   ALTER PLUGGABLE DATABASE apex_pdb1 open;


   ALTER SESSION SET CONTAINER=apex_pdb1;

   SQL> select app_name, app_version, app_status from dba_applications where
   app_name = 'APEX';


   APP_NAME                       APP_VERSION                    APP_STATUS

   ------------------------------ ------------------------------ ------------

   APEX                           24.2                           NORMAL
   ```

Note

`apex_approot1` and the `admin user` in the previous example can be any valid ORACLE identifier.

5.2.7 Configure HTTP Access to the Application PDB

Configure a new application PDB for HTTP access.

Configure the new application PDB for HTTP access by following the instructions starting with the section Downloading and Installing Oracle REST Data Services.

See Also

Setting Up ORDS in an Application Container

5.3 Installing APEX into Different PDBs

You can install different versions of Oracle APEX into different PDBs.

If APEX is not installed in the container database, you can install a local APEX within each PDB as required. When APEX is installed locally there are no APEX metadata linked objects and all packages, views, and tables are created within the `APEX_240200` schema, within each PDB where APEX is installed.

- Uninstalling APEX from a CDB
  Learn how to uninstall Oracle APEX from a CDB.
- Installing APEX Locally in a PDB
  Learn how to install Oracle APEX locally in a PDB.
- Installing APEX into a CDB

5.3.1 Uninstalling APEX from a CDB

Learn how to uninstall Oracle APEX from a CDB.

To uninstall APEX from a CDB:

Note

Installing or removing APEX from a CDB requires a local connection to the database.

This section describes removing APEX from a CDB.

1. Change to the `apex` directory in the location where you unzipped the distribution.
2. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role. For example:

- On Windows:

  ```
  SYSTEM_DRIVE:\ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

3. Run `apxremov.sql`.

   For example:

   ```
   @apxremov.sql
   ```

> **Note**
>
> If you run `apexremov.sql` after PDBs have been added to the CDB, then APEX uninstalls from all of the PDBs, as well as `CDB$ROOT` and `PDB$SEED`. Any applications defined in any of the PDBs will be removed.

5.3.2 Installing APEX Locally in a PDB

Learn how to install Oracle APEX locally in a PDB.

Once you have removed APEX from the container database by following the instructions in Uninstalling APEX from a CDB, you can install APEX locally in a PDB.

To install APEX locally in a PDB:

1. Change the `apex` directory in the location where you unzipped the distribution.
2. Start SQLcl and connect to the database where APEX is installed as SYS specifying the SYSDBA role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Set the container to the PDB you want to install APEX locally:

   ```
   ALTER SESSION SET CONTAINER = <PDB_name>;
   ```

4. Select the appropriate installation option.

Full development environment provides complete access to the App Builder environment to develop applications. A **Runtime environment** enables users to run applications that cannot be modified.

Available installation options include:

- **Full development environment** - Run `apexins.sql` passing the following four arguments in the order shown:

  ```
  @apexins.sql tablespace_apex tablespace_files tablespace_temp images
  ```

  Where:

  - `tablespace_apex` is the name of the tablespace for the APEX platform schema.
  - `tablespace_files` is the name of the tablespace for the APEX files schema.
  - `tablespace_temp` is the name of the temporary tablespace or tablespace group.
  - `images` is the virtual directory for APEX images. To support future APEX upgrades, define the virtual image directory as `/i/`.

  For example:

  ```
  @apexins.sql SYSAUX SYSAUX TEMP /i/
  ```

- **Runtime environment** - Run `apxrtins.sql` passing the following four arguments in the order shown:

  ```
  @apxrtins.sql tablespace_apex tablespace_files tablespace_temp images
  ```

  Where:

  - `tablespace_apex` is the name of the tablespace for the APEX platform schema.
  - `tablespace_files` is the name of the tablespace for the APEX files schema.
  - `tablespace_temp` is the name of the temporary tablespace or tablespace group.
  - `images` is the virtual directory for APEX images. To support future APEX upgrades, define the virtual image directory as `/i/`.

  For example:

  ```
  @apxrtins.sql SYSAUX SYSAUX TEMP /i/
  ```

5. Complete the appropriate steps in Installing and Configuring APEX and Oracle REST Data Services.

See Also

- Installing and Configuring APEX and Oracle REST Data Services
- About APEX Database Accounts
- Working with SQLcl in *Oracle SQLcl User's Guide* for more information about SQLcl

5.3.3 Installing APEX into a CDB

To install Oracle APEX into a CDB:

> **Note**
>
> Installing or removing APEX from a CDB requires a local connection to the database.

1. Change your working directory to the `apex` directory in the location where you unzipped the distribution.
2. Start SQLcl and connect to `CDB$ROOT` of the database where APEX is installed as `SYS` specifying the `SYSDBA` role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Select the appropriate installation option.

   Full development environment provides complete access to the App Builder environment to develop applications. A **Runtime environment** enables users to run applications that cannot be modified.

   Available installation options include:

   - **Full development environment**. Run `apexins.sql` passing the following four arguments in the order shown:

     ```
     @apexins.sql tablespace_apex tablespace_files tablespace_temp images
     ```

     Where:

     - *`tablespace_apex`* is the name of the tablespace for the APEX platform schema.
     - *`tablespace_files`* is the name of the tablespace for the APEX files schema.
     - *`tablespace_temp`* is the name of the temporary tablespace or tablespace group.
     - *`images`* is the virtual directory for APEX images. To support future APEX upgrades, define the virtual image directory as `/i/`.

     Example:

     ```
     @apexins.sql SYSAUX SYSAUX TEMP /i/
     ```

- **Runtime environment**. Run `apxrtins.sql` passing the following arguments in the order shown:

  ```
  @apxrtins.sql tablespace_apex tablespace_files tablespace_temp images
  ```

  Where:

  - *`tablespace_apex`* is the name of the tablespace for the APEX platform schema.
  - *`tablespace_files`* is the name of the tablespace for the APEX files schema.
  - `tablespace_temp` is the name of the temporary tablespace or tablespace group.
  - `images` is the virtual directory for APEX images. To support future APEX upgrades, define the virtual image directory as `/i/`.

  Example:

  ```
  @apxrtins.sql SYSAUX SYSAUX TEMP /i/
  ```

4. Complete appropriate steps in Installing and Configuring APEX and Oracle REST Data Services.

See Also

- Patching or Upgrading APEX in a CDB
- About the APEX Runtime Environment
- Installing and Configuring APEX and Oracle REST Data Services
- About APEX Database Accounts
- Working with SQLcl in *Oracle SQLcl User's Guide*

5.4 Plugging in a PDB When APEX Is Installed in the Root Container

Learn about scenarios in which the target database has Oracle APEX installed into the root container, `CDB$ROOT` - the default installation option.

This section describes scenarios in which the target database has APEX installed into the root container, `CDB$ROOT` - the default installation option. Note there are multiple scenarios related to where the database being plugged in originated from and how APEX was configured in the originating database.

- Scenario 1: Plug-in Non-CDB with APEX
  Plug-in Non-CDB with Oracle APEX.
- Scenario 2: Plug-in PDB with a Common APEX from Another CDB
  Plug-in a PDB with Oracle APEX from another CDB.
- Scenario 3: Plug-in PDB with a Local APEX from Another CDB
  Plug-in a PDB with a local Oracle APEX from another CDB.
- Scenario 4: Plug-in Non-CDB or PDB with No APEX
  Plug-in a Non-CDB or PDB if Oracle APEX is not installed.

- Working with Incompatible APEX Versions
  Learn how to work with the incompatible versions of Oracle APEX.

5.4.1 Scenario 1: Plug-in Non-CDB with APEX

Plug-in Non-CDB with Oracle APEX.

If you are upgrading from a previous Oracle Database release, then you first need to upgrade to a Oracle Database 12*c* non-CDB (or PDB with locally installed APEX) or and then plug the database into your CDB. Alternatively, if you have configured a non-CDB Oracle Database 12*c* or later (or PDB with locally installed APEX), you may now want to plug this database into a CDB. In both cases, the originating database has APEX installed and was not formerly a PDB.

As described in the *Oracle Database Installation Guide* for your operating system, when plugging in a standalone database you need to run the `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql script`. This script creates the necessary metadata linked objects, instead of local objects and recompiles the database objects for all common database options, including APEX.

After installing APEX, you need to configure the web server for the PDB.

If the version of APEX installed in the originating database (which is now a PDB) is different from what is installed into the root container of the target, an error will be raised when trying to open the PDB.

See Also

- Installing and Configuring APEX and Oracle REST Data Services
- Working with Incompatible APEX Versions

5.4.2 Scenario 2: Plug-in PDB with a Common APEX from Another CDB

Plug-in a PDB with Oracle APEX from another CDB.

If you are copying or moving a PDB from an existing Oracle Database 12*c* or later where the originating CDB had APEX installed in the root container, you will not need to perform any additional steps, other than configuring the web server for the PDB.

This scenario assumes APEX release 24.2 is installed and the `APEX_240200` schema within the PDB being plugged in already has the metadata linked objects defined and will compile without error against the metadata linked objects within the target CDB.

If the version of APEX installed in the originating database is different from what is installed in the root container of the target an error is raised when trying to open the PDB.

See Also

- Installing and Configuring APEX and Oracle REST Data Services
- Working with Incompatible APEX Versions

5.4.3 Scenario 3: Plug-in PDB with a Local APEX from Another CDB

Plug-in a PDB with a local Oracle APEX from another CDB.

If you are copying or moving a PDB from an existing Oracle Database 12*c* or later where APEX was not installed in the root container but is installed locally, then you need to perform additional steps before the PDB can be opened without errors.

This scenario assumes APEX release 24.2 is installed and the `APEX_240200` schema within the PDB being plugged in contains all of the APEX database objects and has no metadata linked objects. Therefore, you need to run `$ORACLE_HOME/rdbms/admin/apex_to_common.sql` to remove the common objects and create the metadata links for the packages, views and so forth.

To replace local objects with metadata links in the PDB:

1. Change your working directory to `$ORACLE_HOME/rdbms/admin`.
2. Start SQLcl and connect to the database where APEX is installed as SYS specifying the SYSDBA role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Set the container to the PDB to be configured:

   ```
   ALTER SESSION SET CONTAINER = <PDB_name>;
   ```

4. Run `apex_to_common.sql`. For example:

   ```
   @apex_to_common.sql
   ```

If the version of APEX installed in the originating database is different from what is installed in the root container of the target an error is raised when trying to open the PDB.

See Also

Working with Incompatible APEX Versions

5.4.4 Scenario 4: Plug-in Non-CDB or PDB with No APEX

Plug-in a Non-CDB or PDB if Oracle APEX is not installed.

In this scenario, the APEX schema, such as `APEX_240200` for Oracle APEX release 24.2, will not be present in the originating database or the PDB being plugged in. In order to open the

PDB without issue and be able to run APEX within the new PDB, you must install APEX into the originating database or PDB before attempting to plug in to the target database. You should install the same version of APEX into the originating database or PDB as the version installed into the target database.

5.4.5 Working with Incompatible APEX Versions

Learn how to work with the incompatible versions of Oracle APEX.

If the version of APEX in the root container, `CDB$ROOT`, is not the same as the APEX version in the PDB then an error is raised every time the PDB is opened preventing normal database operations within the PDB. The PDB can only be opened in restricted mode by users with RESTRICTED SESSION privilege, until the versions are compatible.

- Patching or Upgrading APEX in a CDB
  Learn how to patch or upgrade Oracle APEX in the root container.
- Patching or Upgrading APEX in a PDB
  Learn how to patch or upgrade Oracle APEX in a PDB.

5.4.5.1 Patching or Upgrading APEX in a CDB

Learn how to patch or upgrade Oracle APEX in the root container.

If the version of APEX in the PDB is a later minor release version than the version of APEX in the root container (for example, the PDB contains APEX release 5.1.4 and the CDB contains APEX release 5.1.3) then you must patch the version of APEX in the root container to be able to open the PDB without error. If the major version of APEX in the PDB is higher than the version in the CDB (for example the PDB has APEX release 19.2 and the CDB has APEX release 18.1) then you must upgrade the version of APEX in the CDB to be able to open the PDB without error.

To patch APEX in the root container:

1. Download the appropriate patch from My Oracle Support.
2. Unzip and extract the installation files.
3. Change your working directory to where the installation files were extracted
4. Start SQLcl and connect to the database where APEX is installed as SYS specifying the SYSDBA role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run `apxpatch_con.sql` for example:

   ```
   @apxpatch_con.sql
   ```

6. Follow the instructions outlined in the Patch Set Notes for updating the images directory.

> **See Also**
>
> Installing APEX into a CDB

5.4.5.2 Patching or Upgrading APEX in a PDB

Learn how to patch or upgrade Oracle APEX in a PDB.

If the minor version of APEX in the PDB is lower than the version of APEX in the root container (for example the PDB has APEX release 4.2.0 and the CDB has APEX release 4.2.6) then it will be necessary to patch the version of APEX in the PDB. If the major version of APEX in the PDB is lower than the version in the root container (for example, the PDB has APEX release 4.2 and the CDB has APEX release 19.2) then the version of APEX in the PDB will need to be upgraded.

- Patching APEX in a PDB
  Learn how to patch Oracle APEX in a PDB.
- Upgrading APEX in a PDB
  Learn how to upgrade Oracle APEX in a PDB.

5.4.5.2.1 Patching APEX in a PDB

Learn how to patch Oracle APEX in a PDB.

To patch APEX in a PDB:

1. Download the appropriate patch from My Oracle Support.
2. Unzip and extract the installation files.
3. Change your working directory to where the installation files were extracted.
4. Start SQLcl and connect to the database where APEX is installed as SYS specifying the SYSDBA role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run apxpatch.sql using catcon.pl like the following example:

   ```
   host &OH_HOME/perl/bin/perl -I

   &OH_HOME/rdbms/admin &OH_HOME/rdbms/admin/catcon.pl -b apxpatch -c
   '<PDB_name>'  apxpatch.sql
   ```

Where:

- *&OH_HOME* represents the full path to the Oracle home
- <*PDB_name*> is the name of the PDB you are patching

6. Follow the instructions outlined in the patch set notes for updating the images directory.

5.4.5.2.2 Upgrading APEX in a PDB

Learn how to upgrade Oracle APEX in a PDB.

To upgrade APEX in a PDB:

1. Unzip and extract the installation files.
2. Change your working directory to where the installation files were extracted.
3. Start SQLcl and connect to the database where APEX is installed as SYS specifying the SYSDBA role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Run `apexins_nocdb.sql` or `apxrtins_nocdb.sql` using catcon.pl like the following example:

   ```
   host &OH_HOME/perl/bin/perl -I

   &OH_HOME/rdbms/admin &OH_HOME/rdbms/admin/catcon.pl -b apexins -c
   '<PDB_name>'  apexins_nocdb.sql --pSYSAUX --pSYSAUX --pTEMP --p/i/ --p1,2,3
   ```

   Where:

   - *&OH_HOME* represents the full path to the Oracle home
   - <*PDB_name*> is the name of the PDB you are patching

5. Follow the instructions outlined in the patch set notes for updating the images directory.

5.5 Plugging in a PDB When APEX Is Not in the Root Container of the Target CDB

The scenarios in this section describe when Oracle APEX is not installed in the root container, `CDB$ROOT`, by explicitly removing it as described in "Uninstalling APEX from a CDB."

In such cases, you can optionally install a APEX into each PDB independently. If APEX is installed into a PDB it is considered to be installed locally and has no metadata linked objects.

There are multiple scenarios related to where the database being plugged in originated from and how APEX was configured in the originating database.

- Scenario 1: Plug-in a Non-CDB or PDB with Locally Installed APEX
  Plug-in a Non-CDB or PDB with locally installed Oracle APEX.
- Scenario 2: Plug-in PDB with Common APEX from Another CDB
  Plug-in a PDB with Oracle APEX from another CDB.
- Scenario 3: Plug-in PDB with a Local APEX from Another CDB
  Plug-in a PDB with local Oracle APEX from another CDB.
- Scenario 4: Plug-in a Non-CDB or PDB with No APEX
  Plug-in a Non-CDB or PDB if Oracle APEX is not installed.

See Also

Uninstalling APEX from a CDB

5.5.1 Scenario 1: Plug-in a Non-CDB or PDB with Locally Installed APEX

Plug-in a Non-CDB or PDB with locally installed Oracle APEX.

As described in the *Oracle Database Installation Guide* for your operating system, when plugging in a standalone database you need to run the `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql` script. This script creates the necessary metadata linked objects (instead of local objects) and recompiles the database objects for all common database options. However, because APEX has been removed from the root container, the script will not create any metadata links for any of the APEX objects. The script does not change the APEX installation from the originating database and no additional steps are needed other than configuring the web server.

See Also

Installing and Configuring APEX and Oracle REST Data Services

5.5.2 Scenario 2: Plug-in PDB with Common APEX from Another CDB

Plug-in a PDB with Oracle APEX from another CDB.

If you are copying or moving a PDB from an existing Oracle Database where the originating CDB had APEX installed in the root container, then an error is raised whenever you try to open the PDB. The error is due to the originating PDB included metadata links to objects in the originating root container which cannot be recompiled because the target root container does not include APEX. You will not be able to open the PDB unless you remove APEX from the PDB or if APEX is already installed in the target root container. Oracle does not support installing APEX in the root container if it contains PDBs with locally installed APEX.

5.5.3 Scenario 3: Plug-in PDB with a Local APEX from Another CDB

Plug-in a PDB with local Oracle APEX from another CDB.

If you are copying or moving a PDB from an existing Oracle Database where the originating PDB had a local APEX installed (not in the CDB) then you do not need to perform any additional steps, other than configuring the web server in the PDB.

This scenario assumes APEX release 24.2 is installed and the `APEX_240200` schema within the PDB being plugged in, already has all of the APEX objects defined locally and no metadata links.

> **See Also**
>
> Installing and Configuring APEX and Oracle REST Data Services

5.5.4 Scenario 4: Plug-in a Non-CDB or PDB with No APEX

Plug-in a Non-CDB or PDB if Oracle APEX is not installed.

If you are plugging in a non-CDB, or copying or moving a PDB from another CDB, where APEX was not installed in the originating database or PDB then you do not need to perform any additional steps. There will be no APEX engine schema, such as `APEX_240200`, within the PDB, and the PDB can be started without error.

6

Installing and Configuring APEX and Oracle REST Data Services

Install or upgrade Oracle APEX by downloading a ZIP file from the APEX download page and then downloading and installing Oracle REST Data Services (ORDS). These instructions apply to both new and upgrade installations.

- Performing Pre-installation Tasks for APEX
  Review and perform pre-installation tasks before installing Oracle APEX.
- About SQLcl Support
  Oracle SQL Developer Command Line (SQLcl) is a Java-based command-line interface for Oracle Database.
- Downloading and Installing APEX
  Learn about downloading and installing Oracle APEX.
- Downloading and Installing Oracle REST Data Services (ORDS)
  Learn about downloading and installing Oracle REST Data Services (ORDS).
- Configuring Oracle REST Data Services
  Configuring Oracle REST Data Services requires that you copy the images directory, if you are using an older release validate the Oracle REST Data Services installation, configure static files support, and secure Oracle REST Data Services.
- Enabling Network Services in Oracle Database
  Database administrators must enable network services in Oracle Database to send outbound mail, invoke web services, or use template-based PDF report printing with external print servers in Oracle APEX.
- Performing Security Tasks
  Oracle recommends configuring and using Secure Sockets Layer (SSL) to ensure that passwords and other sensitive data are not transmitted in clear text in HTTP requests.
- Controlling the Number of Concurrent Jobs
  Learn about specifying the number of concurrently running jobs.
- About Running APEX in Other Languages
  You can install a single instance of Oracle APEX with one or more translated versions.
- Installing Translated Versions of APEX
  Learn about installing translated versions of Oracle APEX.
- Creating a Workspace and Adding APEX Users
  Before you can develop or install applications, you must create a workspace, add Oracle APEX users, and sign in to your workspace.
- Performing Post Installation Tasks for Upgrade Installations
  Once you have verified that your upgrade installation was successful and all upgraded applications function properly, you should remove schemas from prior Oracle APEX installations.
- About Performance Optimization Tasks
  Learn about performance optimization.

- Converting Between Runtime and Full Development Environments
  Learn about converting between runtime and full development environments.

See Also

Web Server Requirements

6.1 Performing Pre-installation Tasks for APEX

Review and perform pre-installation tasks before installing Oracle APEX.

Before installing APEX, Oracle recommends that you complete the following steps:

1. Review and satisfy all APEX installation requirements.
2. If you are actively using APEX and upgrading the current installation, then shut down with normal or immediate priority the Oracle Database instances where you plan to install APEX. On Oracle Real Application Clusters (Oracle RAC) systems, shut down all instances on each node.

   An alternative to shutting down the database, you can prevent all users from accessing APEX when upgrading your installation from a previous release of APEX. Oracle only recommends this option in high availability production environments where planned outages are not available. For all other scenarios, the database should be shut down.

   To disable access to APEX, shut down the appropriate application server where Oracle REST Data Services is deployed.

   Once you have prevented access from APEX users, log in to SQLcl as `SYS`, connecting to the database where APEX is installed, and query `V$SESSION` to ensure there are no long running sessions which would interfere with the upgrade process.
3. Back up the Oracle Database installation.

   Oracle recommends that you create a backup of the current Oracle Database installation before you install APEX. You can use Oracle Database Recovery Manager, which is included in the Oracle Database installation, to perform the backup.
4. Start the Oracle Database instance that contains the target database.

   After backing up the system, you must start the Oracle instance that contains the target Oracle Database.

See Also

- Oracle APEX Installation Requirements
- *Oracle Database Backup and Recovery User's Guide*

6.2 About SQLcl Support

Oracle SQL Developer Command Line (SQLcl) is a Java-based command-line interface for Oracle Database.

It should be used as the command line interface for running the SQL scripts mentioned in this book.

> **See Also**
>
> Working with SQLcl in *Oracle SQLcl User's Guide*

6.3 Downloading and Installing APEX

Learn about downloading and installing Oracle APEX.

How you install Oracle APEX depends upon by the type of database into which you are installing. This chapter describes how to download and install Oracle APEX in self-managed databases, such as your laptop or your data center, or co-managed Cloud databases such as Database Cloud Service (DBaaS) and Exadata Cloud Service.

> **Tip**
>
> In fully managed Cloud databases, such as an Autonomous Database or Oracle APEX Application Development (APEX Service), APEX is pre-installed and pre-configured, so you may skip the steps listed in this chapter. To learn more, refer to the documentation for your service.

- Installing APEX
  Install Oracle APEX by downloading a ZIP file from the APEX download page.
- Creating or Updating Your Instance Administration Account
  Learn how to create or update Instance Administrator account.
- Restarting Processes
  Restart the processes that you stopped before you began the installation.
- Configuring the APEX_PUBLIC_USER Account
  It is important to correctly configure the `APEX_PUBLIC_USER` account to enable proper operation of Oracle APEX.
- Configuring the APEX_PUBLIC_ROUTER Account
  The APEX_PUBLIC_ROUTER account is a minimally privileged account used for an ORDS REST module to resolve friendly URLs.

Related Topics

- Utilizing Multitenant Architecture
  Learn about installation choices and different scenarios associated with copying and moving pluggable databases introduced by the multitenant architecture with respect to Oracle APEX.
- Welcome to Oracle APEX Application Development Service in *Getting Started with Oracle APEX Application Development*

- Create Applications with Oracle APEX on Autonomous Database in *Using Oracle Autonomous Database Serverless*

6.3.1 Installing APEX

Install Oracle APEX by downloading a ZIP file from the APEX download page.

> **Tip**
>
> APEX must be installed from a writable directory on the file system. See Reviewing a Log of an Installation Session.

To install APEX:

1. For installations where the development will be in English only, download the file `apex_24.2_en.zip` from the APEX download page. If the development will include languages other than English, download `apex_24.2.zip` from the APEX download page. See:

   https://www.oracle.com/tools/downloads/apex-downloads.html

   Note that the actual file name may differ if a more recent release has shipped since this document was published.
2. Unzip downloaded zip file:
   - If English only, unzip `apex_24.2_en.zip` as follows, preserving directory names:
     - UNIX and Linux: `$ unzip apex_24.2_en.zip`
     - Windows: Double click the file `apex_24.2_en.zip` in Windows Explorer
   - If multiple languages, unzip `apex_24.2.zip` as follows, preserving directory names:
     - UNIX and Linux: `$ unzip apex_24.2.zip`
     - Windows: Double click the file `apex_24.2.zip` in Windows Explorer

   > **Note**
   >
   > You should keep the directory tree where you unzip the files short and not under directories that contain spaces. For example, within Windows unzip to `C:\TEMP`.
3. Change your working directory to `apex`.
4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

5. Disable any existing password complexity rules for the default profile.
6. Select the appropriate installation option.

   Full development environment provides complete access to the App Builder environment to develop applications. A **Runtime environment** enables users to run applications that cannot be modified.

   Available installation options include:

   - **Full development environment**. Run `apexins.sql` passing the following four arguments in the order shown:

     ```
     @apexins.sql tablespace_apex tablespace_files tablespace_temp images
     ```

     Where:

     – ***`tablespace_apex`*** is the name of the tablespace for the APEX application user.
     – ***`tablespace_files`*** is the name of the tablespace for the APEX files user.
     – ***`tablespace_temp`*** is the name of the temporary tablespace or tablespace group.
     – ***`images`*** is the virtual directory for APEX images. To support future APEX upgrades, define the virtual image directory as `/i/`.

     Example:

     ```
     @apexins.sql SYSAUX SYSAUX TEMP /i/
     ```

     Note

     If you receive the following error, exit SQLcl and change your working directory to where you unzipped the installation file, for example `C:\TEMP` in Windows, before starting SQLcl:

     ```
     SP2-0310: unable to open file "apexins.sql"
     ```

   - **Runtime environment**. Run `apxrtins.sql` passing the following arguments in the order shown:

     ```
     @apxrtins.sql tablespace_apex tablespace_files tablespace_temp images
     ```

     Where:

     – ***`tablespace_apex`*** is the name of the tablespace for the APEX application user.
     – ***`tablespace_files`*** is the name of the tablespace for the APEX files user.
     – ***`tablespace_temp`*** is the name of the temporary tablespace or tablespace group.
     – ***`images`*** is the virtual directory for APEX images. To support future APEX upgrades, define the virtual image directory as `/i/`.

Example:

```
@apxrtins.sql SYSAUX SYSAUX TEMP /i/
```

See Also

- About the APEX Runtime Environment
- Configuring Password Protection in *Oracle Database Security Guide*
- Working with SQLcl in *Oracle SQLcl User's Guide*
- About APEX Database Accounts

6.3.2 Creating or Updating Your Instance Administration Account

Learn how to create or update Instance Administrator account.

This section describes how to create or update your Instance Administrator account.

Tip

Skip this section if you are upgrading from a previous release of Oracle APEX. In an upgrade scenario, the Instance Administrator account and password is preserved and carried over from the prior release.

- What Is an Instance Administrator?
  Instance administrators are superusers that are responsible for managing an entire Oracle APEX instance, including managing workspace provisioning, configuring features and instance settings, and managing security.
- About apxchpwd.sql
  Running the `apxchpwd.sql` script enables you to create or update your Instance Administrator account.
- Running apxchpwd.sql
  Run the `apxchpwd.sql` script to create and update your Instance Administrator account.

6.3.2.1 What Is an Instance Administrator?

Instance administrators are superusers that are responsible for managing an entire Oracle APEX instance, including managing workspace provisioning, configuring features and instance settings, and managing security.

To perform these tasks, an Instance administrator signs in to the Oracle APEX Administration Services application.

See Also

Oracle APEX Administration Services in *Oracle APEX Administration Guide*

6.3.2.2 About apxchpwd.sql

Running the `apxchpwd.sql` script enables you to create or update your Instance Administrator account.

> **Note**
>
> The `apxchpwd.sql` script is not supported on Oracle Autonomous Database Serverless and Oracle APEX Application Development (APEX Service).

You must run the `apxchpwd.sql` script in the following scenarios:

- **New Oracle APEX installations** - Run `apxchpwd.sql` to create an Instance Administrator account and password.
- **Converting of a runtime environment to a development environment** - Run `apxchpwd.sql` to change the Instance Administrator account password.
- **Changing Your Instance Administrator Password** -Run `apxchpwd.sql` to change the password for an existing Instance Administrator account.
- **Unlocking Your Instance Administrator Account** - Run `apxchpwd.sql` to unlock an existing Instance Administrator account.

> **Tip**
>
> You do not need to run `apxchpwd.sql` when upgrading from a previous release of Oracle APEX. In an upgrade scenario, the Instance Administrator account password is preserved and carried over from the prior release.

6.3.2.3 Running apxchpwd.sql

Run the `apxchpwd.sql` script to create and update your Instance Administrator account.

To create or update your Instance Administrator account:

1. Change your working directory to the `apex` directory where you unzipped the installation software.
2. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role. For example:
    - On Windows:

      ```
      SYSTEM_DRIVE:\ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

    - On UNIX and Linux:

      ```
      $ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

3. Run `apxchpwd.sql`. For example:

   ```
   @apxchpwd.sql
   ```

   Follow the on-screen instructions. You will be prompted provide a username, password, and email address. If the account username does not exist, it will be created for you.

> **See Also**
>
> Working with SQLcl in *Oracle SQLcl User's Guide*

6.3.3 Restarting Processes

Restart the processes that you stopped before you began the installation.

After you install Oracle APEX, you must restart the processes that you stopped before you began the installation.

6.3.4 Configuring the APEX_PUBLIC_USER Account

It is important to correctly configure the `APEX_PUBLIC_USER` account to enable proper operation of Oracle APEX.

- About the APEX_PUBLIC_USER Account
  The `APEX_PUBLIC_USER` account is created with a random password in a new installation of Oracle APEX.
- Unlocking the APEX_PUBLIC_USER Account
  Unlock the `APEX_PUBLIC_USER` account by running a SQL statement.
- Changing the Password for the APEX_PUBLIC_USER Account
  Change the password for the `APEX_PUBLIC_USER` account by running a SQL statement.
- About Password Expiration in Oracle Database
  You can set `PASSWORD_LIFE_TIME` parameter to unlimited by altering `APEX_PUBLIC_USER` to prevent password expiration. To do this create another profile in which the `PASSWORD_LIFE_TIME` parameter is set to unlimited and alter the `APEX_PUBLIC_USER` account and assign it to the new profile.

6.3.4.1 About the APEX_PUBLIC_USER Account

The `APEX_PUBLIC_USER` account is created with a random password in a new installation of Oracle APEX.

You must change the password for this account before configuring the database access descriptor (DAD) in a new installation.

6.3.4.2 Unlocking the APEX_PUBLIC_USER Account

Unlock the `APEX_PUBLIC_USER` account by running a SQL statement.

> **Tip**
>
> If you are upgrading from a prior release of Oracle APEX, this step is unnecessary.

To unlock the `APEX_PUBLIC_USER` account:

1. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role. If Oracle APEX is installed in the CDB, ensure you connect to `CDB$ROOT`. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

2. Run the following statement:

   ```
   ALTER USER APEX_PUBLIC_USER ACCOUNT UNLOCK
   ```

6.3.4.3 Changing the Password for the APEX_PUBLIC_USER Account

Change the password for the `APEX_PUBLIC_USER` account by running a SQL statement.

> **Tip**
>
> If you are upgrading from a prior release of Oracle APEX, this step is unnecessary.

To change the password for the `APEX_PUBLIC_USER` account:

1. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role. If Oracle APEX is installed in the CDB, ensure you connect to `CDB$ROOT`. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

2. Run the following statement:

   ```
   ALTER USER APEX_PUBLIC_USER IDENTIFIED BY new_password
   ```

   Where `new_password` is the new password you are setting for `APEX_PUBLIC_USER`. You will use this password when creating the DAD in the sections that follow.

6.3.4.4 About Password Expiration in Oracle Database

You can set `PASSWORD_LIFE_TIME` parameter to unlimited by altering `APEX_PUBLIC_USER` to prevent password expiration. To do this create another profile in which the `PASSWORD_LIFE_TIME` parameter is set to unlimited and alter the `APEX_PUBLIC_USER` account and assign it to the new profile.

In the default profile in Oracle Database, the parameter `PASSWORD_LIFE_TIME` is set to 180. If you are using Oracle Database with Oracle APEX, this causes the password for `APEX_PUBLIC_USER` to expire in 180 days. As a result, your APEX instance will become unusable until you change the password.

> **See Also**
>
> *Oracle Database Security Guide* for information on creating profiles and assigning them to database users

6.3.5 Configuring the APEX_PUBLIC_ROUTER Account

The APEX_PUBLIC_ROUTER account is a minimally privileged account used for an ORDS REST module to resolve friendly URLs.

About Sessions Per User in Oracle Database

On highly active systems, APEX_PUBLIC_ROUTER may reach the maximum number of sessions allowed in the database which results in a "Service Unavailable HTTP 503" error. You can set the `SESSIONS_PER_USER` parameter to "unlimited" by modifying APEX_PUBLIC_ROUTER to remove limit on the maximum number of sessions.

Create another profile in which the `SESSIONS_PER_USER` parameter is set to "unlimited" and modify the APEX_PUBLIC_ROUTER account and assign it to the new profile.

This behavior differs depending on your version of Oracle REST Data Services (ORDS). For ORDS 24.x and later, all GET requests that reference the `/r/` path use an ORDS REST module.

If ORDS was updated after Oracle APEX, you must run `sys.validate_apex`.

After each ORDS upgrade or if ORDS is installed **after** APEX is installed, you must run the following as `sys`:

```
-- Run as SYS
set serveroutput on
begin
    sys.validate_apex;
end;
/
```

See Also

- DBA_PROFILES in *Oracle Database Reference*

6.4 Downloading and Installing Oracle REST Data Services (ORDS)

Learn about downloading and installing Oracle REST Data Services (ORDS).

The Oracle APEX architecture requires a web server to proxy requests between a web browser and the APEX engine. Oracle REST Data Services meets the requirement but its use goes beyond that of APEX. Oracle REST Data Services simplifies the deployment process because there is no Oracle home required, as connectivity is provided using an embedded JDBC driver.

- Downloading Oracle REST Data Services
  Learn how to download Oracle REST Data Services.
- About Configuring Oracle REST Data Services Behind a Reverse Proxy or Load Balancer
  When Oracle APEX is running behind a reverse proxy or load balancer, it is important to communicate the original HTTP hostname and protocol as seen by the user's browser to the Oracle APEX engine.
- Web Server HTTP POST Request Limits
  Learn about Web Server HTTP POST request limits.

See Also

- Introduction to Oracle REST Data Services in *Oracle REST Data Services Developer's Guide*
- Installing and Configuring Oracle REST Data Services in *Oracle REST Data Services Installation and Configuration Guide*

6.4.1 Downloading Oracle REST Data Services

Learn how to download Oracle REST Data Services.

> **Tip**
>
> By default, the context root for accessing Oracle APEX through Oracle REST Data Services is `/ords`. If you wish to have a context root of `/apex` for accessing Oracle APEX, rename the `ords.war` file to `apex.war` before installing Oracle REST Data Services. See Deploying and Monitoring Oracle REST Data Services in *Oracle REST Data Services Installation and Configuration Guide*.

To download Oracle REST Data Services:

1. Download the latest release of Oracle REST Data Services from the Oracle REST Data Services download page.
2. Unzip the downloaded zip file into a directory (or folder) of your choice:
   - UNIX and Linux: unzip `ords.`*`version.number`*`.zip`
   - Windows: Double-click the file `ords.`*`version.number`*`.zip` in Windows Explorer
3. Copy the images directory, `apex/images`, from the APEX software ZIP to a location on the file system where Oracle REST Data Services is installed.

   > **Note**
   >
   > This step is optional if you use the APEX static resource of the CDN. See Managing Static Resources (Images).

4. Follow and complete all installation and configuration steps described in Installing and Configuring Oracle REST Data Services in *Oracle REST Data Services Installation and Configuration Guide*.
5. For Oracle Database multitenant architecture, ensure that you configure the connection using the service name of the specific pluggable database (PDB) you want to access. Do not use the service name of the `CDB$ROOT` unless you are configuring Oracle REST Data Services to address PDBs through the URL. See Using the Multitenant Architecture with Oracle REST Data Services in *Oracle REST Data Services Installation and Configuration Guide* for more information.

6.4.2 About Configuring Oracle REST Data Services Behind a Reverse Proxy or Load Balancer

When Oracle APEX is running behind a reverse proxy or load balancer, it is important to communicate the original HTTP hostname and protocol as seen by the user's browser to the Oracle APEX engine.

The Oracle APEX engine uses this information to generate valid URLs in HTML responses and HTTP redirects that the user's browser can successfully follow. The exact configuration steps depend on your Java EE application server. For example, for Oracle WebLogic Server, this is accomplished using Oracle WebLogic Server Proxy Plug-Ins. To learn more, see your Java EE application server documentation.

6.4.3 Web Server HTTP POST Request Limits

Learn about Web Server HTTP POST request limits.

When running Oracle REST Data Services (ORDS) in standalone mode or within a Tomcat Java Container, size limits are being imposed on POST requests which are **not** file uploads. Oracle APEX users will encounter these limits when uploading data in SQL Workshop using copy and paste or when using copy and paste while building an application from spreadsheet.

- When running Oracle REST Data Services in **Standalone Mode**, the default limit is 200 KB for ORDS 19.4.6 and earlier. It is recommended to increase the limit as follows:

  Set the Java System property `org.eclipse.jetty.server.Request.maxFormContentSize` to a higher value in bytes. You can set this property upon startup of Oracle REST Data Services. For example: `java -Dorg.eclipse.jetty.server.Request.maxFormContentSize=3000000 -jar ords.war`
- When running on Apache Tomcat, the default limit is 2 megabytes. Adjust Apache Tomcat's `maxPostSize` parameter to change that limit.

> **See Also**
>
> http://tomcat.apache.org/ for more information.

6.5 Configuring Oracle REST Data Services

Configuring Oracle REST Data Services requires that you copy the images directory, if you are using an older release validate the Oracle REST Data Services installation, configure static files support, and secure Oracle REST Data Services.

- Managing Static Resources (Images)
  Manage images and static resources used by an Oracle APEX installation.
- Validating the Oracle REST Data Services Installation
  In a new installation or upgrade of Oracle APEX and if you are using Oracle REST Data Services 21.2.1 or older, you must validate the Oracle REST Data Services installation.
- Configuring Static File Support
  For configuring static files, you must run `apex_rest_config.sql` after a new installation of Oracle APEX.
- Securing Oracle REST Data Service
  In a configuration for Oracle APEX, Oracle recommends setting the parameter `security.requestValidationFunction` to `wwv_flow_epg_include_modules.authorize`.

6.5.1 Managing Static Resources (Images)

Manage images and static resources used by an Oracle APEX installation.

The images and static resources used by an Oracle APEX installation can either be stored on the local file system accessible by Oracle REST Data Services, or referenced to the Oracle APEX static resources Content Delivery Network (CDN).

- Copying the Images Directory
  Store images and static resources used by an Oracle APEX installation on a local filesystem accessible by Oracle REST Data Services.
- Using a Static Resources CDN
  Reference images and static resources used by an Oracle APEX instance by referencing a static resources Content Delivery Network (CDN).

6.5.1.1 Copying the Images Directory

Store images and static resources used by an Oracle APEX installation on a local filesystem accessible by Oracle REST Data Services.

> **Note**
>
> Skip this step if you choose to use the Oracle APEX static resources CDN as described in Using a Static Resources CDN.

Perform this task whether you are loading a new installation or upgrading from a previous version. To host the images and static resources on a local file system, you must copy the images directory from the top level of the `apex\images` directory, for example `C:\TEMP`, to the location used by your Oracle REST Data Services installation.

During an upgrade, you overwrite your existing images directory. Before you begin the upgrade, to ensure that you can revert to the previous version, Oracle recommends that you create a copy of your existing images directory for APEX, indicating the release number of the images (for example, `images_24_1`).

6.5.1.2 Using a Static Resources CDN

Reference images and static resources used by an Oracle APEX instance by referencing a static resources Content Delivery Network (CDN).

Using the CDN may improve performance of APEX and your applications by distributing the static resources from a server that is physically located near the end user. Enabling the CDN for your APEX instance is a one-time configuration step. The CDN reference is automatically updated during patches and upgrades to point to the appropriate CDN for that version.

To use the APEX static resources CDN, connect to the database as a user with the APEX_ADMINISTRATOR_ROLE role and run the following code:

```
begin
    for c1 in (select version_no from APEX_RELEASE)
    loop
        APEX_INSTANCE_ADMIN.set_parameter(
            p_parameter => 'IMAGE_PREFIX',
            p_value => 'https://static.oracle.com/cdn/apex/' ||
                            c1.version_no || '/');
    end loop;

    commit;
end;
```

6.5.2 Validating the Oracle REST Data Services Installation

In a new installation or upgrade of Oracle APEX and if you are using Oracle REST Data Services 21.2.1 or older, you must validate the Oracle REST Data Services installation.

For validating the Oracle REST Data Services installation in a new installation or upgrade of Oracle APEX, run the following command:

```
java -jar ords.war validate [--database <dbname>]
```

> ⓘ **See Also**
>
> Repairing the Oracle REST Data Services Installation in *Oracle REST Data Services Installation and Configuration Guide*

6.5.3 Configuring Static File Support

For configuring static files, you must run `apex_rest_config.sql` after a new installation of Oracle APEX.

Oracle APEX enables application developers to include static files with their applications. Static files can be associated with a workspace, an application, a plug-in, or an application theme. When using Oracle REST Data Services as your web server, static files are served using RESTful service module built into Oracle APEX. Therefore, you must run `apex_rest_config.sql` after a new installation of Oracle APEX.

6.5.4 Securing Oracle REST Data Service

In a configuration for Oracle APEX, Oracle recommends setting the parameter `security.requestValidationFunction` to `wwv_flow_epg_include_modules.authorize`.

Set parameter `security.requestValidationFunction` to `wwv_flow_epg_include_modules.authorize` activates the allowed list of callable procedures which ships with Oracle APEX and prohibits calls to other procedures.

> ⓘ **See Also**
>
> About Configuring Oracle REST Data Services with Oracle APEX in *Oracle APEX App Builder User's Guide*

6.6 Enabling Network Services in Oracle Database

Database administrators must enable network services in Oracle Database to send outbound mail, invoke web services, or use template-based PDF report printing with external print servers in Oracle APEX.

To learn more, see Enabling Network Services in Oracle Database in *Oracle APEX Installation Guide*.

Note

Enabling network services does not apply to APEX instances running on Oracle Autonomous Database. APEX can communicate with external endpoints over the internet without additional configuration.

- When and Why Network Services Must be Enabled
  Enabling network services enables support for sending outbound mail in Oracle APEX, using REST Services, REST Enabled SQL, or other web services, and using a remote server for report printing.
- Granting Connect Privileges
- Troubleshooting an Invalid ACL Error
  Learn how to identify any invalid ACL error by running the query.

6.6.1 When and Why Network Services Must be Enabled

Enabling network services enables support for sending outbound mail in Oracle APEX, using REST Services, REST Enabled SQL, or other web services, and using a remote server for report printing.

By default, the ability to interact with network services is disabled in Oracle Database. Therefore, you must use the `DBMS_NETWORK_ACL_ADMIN` package to grant network connect privileges to the database user that owns the APEX schema (`APEX_240200`). Failing to grant these privileges results in issues with:

- Sending outbound mail in Oracle APEX.

  Users can call methods from the `APEX_MAIL` package, but issues arise when sending outbound email.
- Consuming REST services and other web services from APEX.
- Making outbound LDAP calls from APEX.
- Using a remote print server for report printing.

The granted network connect privileges apply to the entire APEX instance and enable all applications in all workspaces to perform outbound network calls. You do *not* need to grant network connect privileges to individual workspace schemas unless applications also use native database PL/SQL API such as `UTL_HTTP` and `UTL_SMTP`.

Note

When upgrading APEX, the upgrade automatically configures Network Services based on the configuration of the previous APEX version.

> **Tip**
>
> To run the examples described in this section, the compatible initialization parameter of the database must be set to at least `11.1.0.0.0`. By default, the parameter is set properly, but a database upgraded from a version prior to 11*g* may require an update. For information about changing database initialization parameters, see Specifying the Database Compatibility Level in *Oracle Multitenant Administrator's Guide*.

> **See Also**
>
> About Report Printing in *Oracle APEX App Builder User’s Guide*.

6.6.2 Granting Connect Privileges

The following example demonstrates how to grant connect privileges to any host for the the database user that owns the APEX schema (`APEX_240200`). This example assumes you connected to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role.

```
BEGIN
    DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
        host => '*',
        ace => xs$ace_type(privilege_list => xs$name_list('connect'),
                           principal_name => 
APEX_APPLICATION.g_flow_schema_owner,
                           principal_type => xs_acl.ptype_db));
END;
/
```

The following example demonstrates how to provide less privileged access to local network resources. This example enables access only to servers running on the same database host (`localhost`), such as email and report printing servers.

```
BEGIN
    DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
        host => 'localhost',
        ace => xs$ace_type(privilege_list => xs$name_list('connect'),
                           principal_name => 
APEX_APPLICATION.g_flow_schema_owner,
                           principal_type => xs_acl.ptype_db));
END;
/
```

6.6.3 Troubleshooting an Invalid ACL Error

Learn how to identify any invalid ACL error by running the query.

If you receive an `ORA-44416: Invalid ACL` error after running the previous script, use the following query to identify the invalid ACL:

```
REM Show the dangling references to dropped users in the ACL that is assigned
REM to '*'.

SELECT ACL, PRINCIPAL
  FROM DBA_NETWORK_ACLS NACL, XDS_ACE ACE
 WHERE HOST = '*' AND LOWER_PORT IS NULL AND UPPER_PORT IS NULL AND
       NACL.ACLID = ACE.ACLID AND
       NOT EXISTS (SELECT NULL FROM ALL_USERS WHERE USERNAME = PRINCIPAL);
```

Next, run the following code to fix the ACL:

```
DECLARE
  ACL_ID   RAW(16);
  CNT      NUMBER;
BEGIN
  -- Look for the object ID of the ACL currently assigned to '*'
  SELECT ACLID INTO ACL_ID FROM DBA_NETWORK_ACLS
   WHERE HOST = '*' AND LOWER_PORT IS NULL AND UPPER_PORT IS NULL;

  -- If just some users referenced in the ACL are invalid, remove just those
  -- users in the ACL. Otherwise, drop the ACL completely.
  SELECT COUNT(PRINCIPAL) INTO CNT FROM XDS_ACE
   WHERE ACLID = ACL_ID AND
         EXISTS (SELECT NULL FROM ALL_USERS WHERE USERNAME = PRINCIPAL);

  IF (CNT > 0) THEN

    FOR R IN (SELECT PRINCIPAL FROM XDS_ACE
               WHERE ACLID = ACL_ID AND
                     NOT EXISTS (SELECT NULL FROM ALL_USERS
                                  WHERE USERNAME = PRINCIPAL)) LOOP
      UPDATE XDB.XDB$ACL
         SET OBJECT_VALUE =
               DELETEXML(OBJECT_VALUE,
                         '/ACL/ACE[PRINCIPAL="'||R.PRINCIPAL||'"]')
       WHERE OBJECT_ID = ACL_ID;
    END LOOP;

  ELSE
    DELETE FROM XDB.XDB$ACL WHERE OBJECT_ID = ACL_ID;
  END IF;

END;
/

REM commit the changes.
```

```
COMMIT;
```

Once the ACL has been fixed, you must run the first script in this section to apply the ACL to the `APEX_240200` user.

6.7 Performing Security Tasks

Oracle recommends configuring and using Secure Sockets Layer (SSL) to ensure that passwords and other sensitive data are not transmitted in clear text in HTTP requests.

Without the use of SSL, passwords could potentially be exposed, compromising security.

SSL is an industry standard protocol that uses RSA public key cryptography in conjunction with symmetric key cryptography to provide authentication, encryption, and data integrity.

> ⓘ **See Also**
>
> Configuring HTTP Protocol Attributes in *Oracle APEX Administration Guide*

6.8 Controlling the Number of Concurrent Jobs

Learn about specifying the number of concurrently running jobs.

- About Managing the Number of Concurrent Jobs
  Learn about managing maximum number of concurrently running jobs.
- Viewing the Number of JOB_QUEUE_PROCESSES
  You can view number of `JOB_QUEUE_PROCESSES` in three ways.
- Changing the Number of JOB_QUEUE_PROCESSES
  You can change the number of `JOB_QUEUE_PROCESSES` by running a SQL statement in SQLcl.

6.8.1 About Managing the Number of Concurrent Jobs

Learn about managing maximum number of concurrently running jobs.

`JOB_QUEUE_PROCESSES` determine the maximum number of concurrently running jobs. In Oracle APEX transactional support and SQL scripts require jobs. If `JOB_QUEUE_PROCESSES` is not enabled and working properly, you cannot successfully execute a script.

6.8.2 Viewing the Number of JOB_QUEUE_PROCESSES

You can view number of `JOB_QUEUE_PROCESSES` in three ways.

- Viewing JOB_QUEUE_PROCESSES in the Installation Log File
  View `JOB_QUEUE_PROCESSES` in the installation log files.
- Viewing JOB_QUEUE_PROCESSES in APEX
  View the number of `JOB_QUEUE_PROCESSES` on the About Oracle APEX page.
- Viewing JOB_QUEUE_PROCESSES from SQLcl
  View the number of `JOB_QUEUE_PROCESSES` from SQLcl.

6.8.2.1 Viewing JOB_QUEUE_PROCESSES in the Installation Log File

View `JOB_QUEUE_PROCESSES` in the installation log files.

> **See Also**
>
> Reviewing a Log of an Installation Session

6.8.2.2 Viewing JOB_QUEUE_PROCESSES in APEX

View the number of `JOB_QUEUE_PROCESSES` on the About Oracle APEX page.

To view the About Oracle APEX page:

1. Sign in to APEX.
2. Locate the Help menu at the top of the page.
3. From the Help menu, select **About**.

   The About APEX page appears.
4. Scroll down and find `JOB_QUEUE_PROCESSES` at the bottom of the page.

> **See Also**
>
> Signing In to Your Workspace

6.8.2.3 Viewing JOB_QUEUE_PROCESSES from SQLcl

View the number of `JOB_QUEUE_PROCESSES` from SQLcl.

To view the number of `JOB_QUEUE_PROCESSES` from SQLcl:

1. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

2. Run the appropriate SQL statement. For example:

   ```
   SELECT VALUE FROM v$parameter WHERE NAME = 'job_queue_processes'
   ```

6.8.3 Changing the Number of JOB_QUEUE_PROCESSES

You can change the number of `JOB_QUEUE_PROCESSES` by running a SQL statement in SQLcl.

To update the number of `JOB_QUEUE_PROCESSES`:

1. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

2. In SQLcl run the following SQL statement:

   ```
   ALTER SYSTEM SET JOB_QUEUE_PROCESSES = <number>
   ```

   For example, running the statement `ALTER SYSTEM SET JOB_QUEUE_PROCESSES = 20` sets `JOB_QUEUE_PROCESSES` to 20.

6.9 About Running APEX in Other Languages

You can install a single instance of Oracle APEX with one or more translated versions.

The APEX developer and admin interface is translated into the 9 standard languages: French, German, Italian, Japanese, Korean, Portuguese (Brazil), Simplified Chinese, Spanish, and Traditional Chinese. Developers can choose to run the APEX development environment in any of the installed languages by simply selecting the language from the App Builder sign-in or home page.

The APEX runtime engine which is used by developers to create applications is available in the following additional languages: Arabic, Brazilian Portuguese, Croatian, Czech, Danish, Dutch, Finnish, French, French - Canada, German, Greek, Hebrew, Hungarian, Icelandic Italian, Japanese, Korean, Norwegian, Polish, Portuguese (Portugal) (pt), Romanian, Russian, Serbian - Cyrillic, Serbian - Latin, Simplified Chinese, Slovak, Slovenian, Spanish, Swedish, Thai, Traditional Chinese, Turkish, Ukrainian, and Vietnamese.

A single instance of APEX can be installed with one or more of these translated versions.

In order to install other languages you must use the `apex_24.2.zip` file which contains the extra files as described in Installing Translated Versions of APEX. If you previously downloaded `apex_24.2_en.zip`, then you do not need to re-install APEX. Simply download `apex_24.2.zip` and unzip the file into the same directory where you unzipped `apex_24.2_en.zip`.

The translated version of APEX should be loaded into a database that has a character set that supports the specific language. If you attempt to install a translated version of APEX into a database that does not support the character encoding of the language, the installation may fail or the translated APEX instance may appear corrupt when run. The database character set `AL32UTF8` supports all the translated versions of APEX.

You can manually install translated versions of APEX using SQLcl. The installation files are encoded in `AL32UTF8`.

> **Note**
>
> Regardless of the target database character set, to install a translated version of APEX, you must set the character set value of the `NLS_LANG` environment variable to `AL32UTF8` before starting SQLcl.

The following examples illustrate valid `NLS_LANG` settings for loading APEX translations:

```
American_America.AL32UTF8
Japanese_Japan.AL32UTF8
```

> **See Also**
>
> Installing Translated Versions of APEX

6.10 Installing Translated Versions of APEX

Learn about installing translated versions of Oracle APEX.

- About Installing Translated Versions of APEX
  Whether you are installing for the first time or upgrading from a previous release, you must run the `load_lang.sql` script to run a translated version of Oracle APEX.
- Installing a Translated Version of APEX
  Learn how to run the appropriate language specific script to install a translated version of Oracle APEX.

> **See Also**
>
> Managing Application Globalization in *Oracle APEX App Builder User's Guide*

6.10.1 About Installing Translated Versions of APEX

Whether you are installing for the first time or upgrading from a previous release, you must run the `load_lang.sql` script to run a translated version of Oracle APEX.

The APEX developer and admin interface is translated into the 9 standard languages: French, German, Italian, Japanese, Korean, Portuguese (Brazil), Simplified Chinese, Spanish, and Traditional Chinese. Developers can choose to run the APEX development environment in any of the installed languages by simply selecting the language from the Sign In page or home page.

The APEX runtime engine which is used by developers to create applications is available in the following languages: Arabic, Brazilian Portuguese, Croatian, Czech, Danish, Dutch, Finnish, French, French - Canada, German, Greek, Hebrew, Hungarian, Icelandic, Italian, Japanese, Korean, Norwegian, Polish, Portuguese (Portugal) (pt), Romanian, Russian, Serbian - Cyrillic, Serbian - Latin, Simplified Chinese, Slovak, Slovenian, Spanish, Swedish, Thai, Traditional Chinese, Turkish, Ukrainian, and Vietnamese.

> ✓ **Tip**
>
> In fully managed Cloud databases such as Oracle Autonomous Database Serverless or Oracle APEX Application Development (APEX Service), APEX is pre-installed and pre-configured with all languages from the above list. Separate installation of the supported languages is unnecessary.

To support additional languages not covered in the above list, developers must provide their own translations. For example, if you develop a Bulgarian application and want to include report messages, such as pagination, in Bulgarian, you must translate the strings used in messages displayed in reports.

> ⓘ **See Also**
>
> Translating Messages Used Internally by APEX in *Oracle APEX App Builder User's Guide*

6.10.2 Installing a Translated Version of APEX

Learn how to run the appropriate language specific script to install a translated version of Oracle APEX.

The installation scripts are located in subdirectories identified by a language code in the unzipped distribution `apex/builder`. For example, the German version is located in `apex/builder/de` and the Japanese version is located in `apex/builder/ja`. Within each directory, there is a language loading script identified by the language code (for example, `load_de.sql` or `load_ja.sql`).

> ⓘ **Note**
>
> If you have applied a Patch Set and then install translations, you must re-execute the Patch Set to apply all fixes to the translations.

To install a translated version of APEX:

1. Set the `NLS_LANG` environment variable, making sure that the character set is `AL32UTF8`. For example:

- Bourne or Korn shell:

  ```
  NLS_LANG=American_America.AL32UTF8
  export NLS_LANG
  ```

- C shell:

  ```
  setenv NLS_LANG American_America.AL32UTF8
  ```

- For Windows based systems:

  ```
  set NLS_LANG=American_America.AL32UTF8
  ```

2. Navigate to the directory under `apex/builder` based on the language you need to install. For example for German, navigate to `apex/builder/de`. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Execute the appropriate language specific script. For example:

   ```
   @load_lang.sql
   ```

   Where `lang` is the specific language (for example, `load_de.sql` for German or `load_ja.sql` for Japanese).

6.11 Creating a Workspace and Adding APEX Users

Before you can develop or install applications, you must create a workspace, add Oracle APEX users, and sign in to your workspace.

- About Workspaces and Users
  A workspace enables multiple users to work within the same Oracle APEX installation while keeping their objects, data, and applications private.
- Signing In To Administration Services
  Sign in to Oracle APEX Administration Services using the Instance administrator account and password created or reset during the installation process.
- Creating a Workspace Manually
  Sign in to Oracle APEX Administration Services to create workspace manually.
- Creating APEX Users
  Create new users in Oracle APEX Administration Services.

- Signing In to Your Workspace
  Sign in to a workspace by navigating to the Oracle APEX Sign In page.

6.11.1 About Workspaces and Users

A workspace enables multiple users to work within the same Oracle APEX installation while keeping their objects, data, and applications private.

You access the APEX home page by logging in to a workspace using a JavaScript enabled Web browser.

Each workspace has a unique ID and name. An instance administrator can create a workspace manually within APEX Administration Services or have users submit requests. APEX Administration Services is a separate application for managing an entire APEX instance.

6.11.2 Signing In To Administration Services

Sign in to Oracle APEX Administration Services using the Instance administrator account and password created or reset during the installation process.

To manually create a workspace and user accounts, you sign in to a separate application for managing an entire Oracle APEX instance called Oracle APEX Administration Services.

To sign in to Oracle APEX Administration Services:

1. In a Web browser, navigate to the Administration Services Sign In page:

   > **Tip**
   >
   > By default, the context root for accessing Oracle APEX through Oracle REST Data Services is `/ords`. If you wish to have a context root of `/apex` for accessing Oracle APEX, rename the `ords.war` file to `apex.war` before installing Oracle REST Data Services. See Deploying and Monitoring Oracle REST Data Services in *Oracle REST Data Services Installation and Configuration Guide*.

   By default, Administration Services installs to the following location:

   ```
   http://hostname:port/ords/apex_admin
   ```

   Where:

   - `hostname` is the name of the system where Oracle REST Data Services is installed.
   - `port` is the port number assigned when configuring Oracle REST Data Services. In a default installation, this number is 8080. To learn more, see Installing and Configuring Oracle REST Data Services in *Oracle REST Data Services Installation and Configuration Guide*.
   - `ords` is the service name defined when configuring Oracle REST Data Services.

2. On the Sign In page:
   a. Username - Enter the Instance administrator account username specified in Creating or Updating Your Instance Administration Account.
   b. Password - Enter your Instance administrator account password.
   c. Click **Sign In to Administration**.

Oracle APEX Administration Services appears.

Note that, depending on your setup, you might be required to change your password when you log in for the first time.

> **See Also**
>
> Oracle APEX Administration Services in *Oracle APEX Administration Guide*

6.11.3 Creating a Workspace Manually

Sign in to Oracle APEX Administration Services to create workspace manually.

To manually create a workspace you sign in, sign in to APEX Administration Services using the `ADMIN` account and password created or reset during the installation process.

To create an APEX workspace manually:

1. Access APEX Administration Services.

   Administration Services appears. Next, create a workspace.
2. Click **Manage Workspaces**.
3. Under Workspace Actions, click **Create Workspace**.

   The Create Workspace Wizard appears.
4. For Identify Workspace, enter the following:
   a. Workspace Name - Enter a unique workspace name.
   b. Workspace ID - Leave Workspace ID blank to have the new Workspace ID automatically generated. A Workspace ID must be a positive integer greater than 100000.
   c. Workspace Description - Enter a workspace description.
   d. Click **Next**.
5. For Identify Schema, specify whether you are re-using an existing schema or creating a new one.

   If you are using an existing schema:
   a. For Re-use existing schema, select **Yes**.
   b. Select a schema from the list.
   c. Click **Next**.

   If you are creating a new schema:
   a. For Re-use existing schema, select **No**.
   b. Enter a schema name and password.
   c. Specify a space quota.
   d. Click **Next**.
6. For Identify Administrator, enter the Workspace administrator information and click **Next**.
7. Confirm your selections and click **Create Workspace**.

See Also

- Creating Workspaces in Administration Services in *Oracle APEX Administration Guide*
- Managing Existing Workspaces in *Oracle APEX Administration Guide*

6.11.4 Creating APEX Users

Create new users in Oracle APEX Administration Services.

Create new users by signing into the APEX Administration Services application using your Instance administrator password.

To create an APEX user account:

1. Sign in to APEX Administration Services.
2. Click **Manage Workspaces**.
3. Under Workspace Actions, click **Manage Developers and Users**.

   The Manage Developers and Users page appears.
4. Click **Create User**.
5. Under User Attributes, enter the appropriate information. Fields marked with an asterisk are required.

   Tip

   Most attributes in APEX include field-level Help. Attributes with field-level Help, have light gray icon that resembles a question mark (?). To view field-level Help, click the Help icon.
6. Under Account Privileges:
   a. **Workspace** - Select a workspace from the list.
   b. **Default Schema** - Specify the default schema used for this user

      When using workspaces that have more than one schema available, this schema is the default. This setting does not control security, only the user's preference.
   c. **User is an administrator** - Specify if this user should have workspace administrator privileges.

      Administrators are given access to all components. Additionally, they can manage user accounts, groups, and development services. Components may not be available if they are switched off by Instance Administrators.
   d. **User is a developer** - Specify if this user should have developer privileges.

      Developers must have access to either App Builder, SQL Workshop, or both. These components may not be available if they are switched off by the Instance Administrator.
   e. **App Builder Access** - Determines whether a developer has access to the App Builder.

f. **SQL Workshop Access** - Determines whether a developer has access to the SQL Workshop.

g. **Team Development Access** - Determines whether a developer has access to the Team Development.

h. **Set Account Availability** - Select **Locked** to prevent the account from being used. Select **Unlocked** to allow the account to be used.

   If the user has exceeded the maximum log in failures allowed, specified in Workspace Preferences, then their account will be locked automatically.

a. **Workspace** - Select a workspace in which to create the user.

b. **Default Schema** - Select the default schema for this user.

c. **Accessible Schemas (null for all)** - Enter a colon-delimited list of schemas for which this developer has permissions when using the SQL Workshop.

   The list of schemas you enter here restricts the user to a subset of the full set of schemas provisioned for the workspace and determines what schema names the user sees in SQL Workshop.

d. **User is an administrator** - Select **Yes** or **No** to specify if this user should have workspace administrator privileges.

   Administrators are given access to all components. Additionally, they can manage user accounts, groups, and development services. Components may not be available if they are switched off by an Instance Administrator.

e. **User is a developer** - Select **Yes** or **No** to specify if this user should have developer privileges.

   Developers must have access to either the App Builder, SQL Workshop, or both. Components may not be available if they are switched off by Instance Administrators.

f. **App Builder Access** - Determines whether a developer has access to App Builder

g. **SQL Workshop Access** - Determines whether a developer has access to the SQL Workshop.

h. **Team Development Access** - Determines whether a user has access to the Team Development.

i. **Account Availability** - Select **Locked** to prevent the account from being used. Select **Unlocked** to allow the account to be used.

7. Under Password:
   - **Password** - Enter a case sensitive password.
   - **Confirm Password** - Enter the password again.
   - **Require Change of Password On First Use** - Select **No** to allow the user to use the same password until it expires. Select **Yes** to require the user to change the password immediately when logging in the first time.
8. Click **Create User** or **Create and Create Another**.

See Also

Managing Users Across an Oracle APEX Instance in *Oracle APEX Administration Guide*

6.11.5 Signing In to Your Workspace

Sign in to a workspace by navigating to the Oracle APEX Sign In page.

After you create a workspace and APEX users, you can sign in to your workspace using your credentials (that is, your workspace name, user name and password).

To sign in to a workspace:

1. In a Web browser, navigate to the APEX Sign In page:

   ```
   http://hostname:port/apex/
   ```

   Where:

   - *`hostname`* is the name of the system where Oracle REST Data Services is installed.
   - *`port`* is the port number assigned when configuring Oracle REST Data Services. In a default installation, this number is 8080. To learn more, see Installing and Configuring Oracle REST Data Services in *Oracle REST Data Services Installation and Configuration Guide*.
   - *`apex`* is the service name defined when configuring Oracle REST Data Services.

   The Sign In page appears.

2. On the Sign In page, enter:

   - Workspace - Enter the name of your workspace.
   - Username - Enter your user name.
   - Password - Enter your case-sensitive password.

3. Click **Sign In**.

   Note that, depending on your setup, you might be required to change your password when you log in for the first time.

> **See Also**
>
> - Creating Workspaces in Administration Services in *Oracle APEX Administration Guide*
> - Managing Requests in *Oracle APEX Administration Guide*

6.12 Performing Post Installation Tasks for Upgrade Installations

Once you have verified that your upgrade installation was successful and all upgraded applications function properly, you should remove schemas from prior Oracle APEX installations.

- About Removing Prior APEX Installations
  Learn about removing schemas from a prior installation by verifying if a prior installation exists.

- Verifying if a Prior Installation Exists
  Run the SQL query to verify if a prior Oracle APEX installation exists.
- Removing Schemas and SYS Objects from Prior Installations
  Start SQLcl and connect to database and execute a statement to remove schemas and `SYS` objects.
- Removing Schemas from Prior Installations in a CDB
  Use `catcon.pl` to remove schemas of prior installations in a CDB.
- Fixing Invalid ACL
  Learn how to fix an invalid ACL.

See Also

Upgrading from a Previous APEX Release

6.12.1 About Removing Prior APEX Installations

Learn about removing schemas from a prior installation by verifying if a prior installation exists.

The database users associated with schemas from prior installations are privileged users and should be removed when they are no longer necessary. Removing schemas from a prior installation is a two step process. First you verify if a prior installation exists and then you remove the schemas.

6.12.2 Verifying if a Prior Installation Exists

Run the SQL query to verify if a prior Oracle APEX installation exists.

To verify if a prior installation exists:

1. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS`. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

2. Run the following query:

   ```
   SELECT username
     FROM dba_users
    WHERE (  username LIKE 'FLOWS\_______' ESCAPE '\'
          OR username LIKE 'APEX\_______' ESCAPE '\' )
      AND username NOT IN ( SELECT schema
   ```

```
    FROM dba_registry
  WHERE comp_id = 'APEX' );
```

If the results contain entries in the form `FLOWS_`*`XXXXXX`* or `APEX_`*`XXXXXX`* where *`XXXXXX`* represents six numbers, those entries are candidates for removal.

6.12.3 Removing Schemas and SYS Objects from Prior Installations

Start SQLcl and connect to database and execute a statement to remove schemas and `SYS` objects.

To remove schemas and `SYS` objects from prior installations:

1. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

2. Execute statements similar to the following example:

   ```
   DROP USER APEX_240100 CASCADE;
   DROP PACKAGE SYS.WWV_DBMS_SQL_APEX_240100;
   ```

6.12.4 Removing Schemas from Prior Installations in a CDB

Use `catcon.pl` to remove schemas of prior installations in a CDB.

To remove schemas and SYS objects from prior installations, run commands using the following example:

```
$ORACLE_HOME/perl/bin/perl -I $ORACLE_HOME/rdbms/admin $ORACLE_HOME/rdbms/
admin/catcon.pl -b drop_apex240100 -- --x'drop user APEX_240100 cascade'
$ORACLE_HOME/perl/bin/perl -I $ORACLE_HOME/rdbms/admin $ORACLE_HOME/rdbms/
admin/catcon.pl -b drop_wwv_dbms_sql -- --x'drop package
SYS.WWV_DBMS_SQL_APEX_240100'
```

6.12.5 Fixing Invalid ACL

Learn how to fix an invalid ACL.

After following the instructions in About Removing Prior APEX Installations, you may need to fix an invalid ACL if you are running Oracle Database and you enabled network services for the prior Oracle APEX schema.

To fix an invalid ACL:

1. Change your working directory to the `apex` directory where you unzipped the installation software.
2. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Execute a statement similar to the following:

   ```
   EXEC DBMS_NETWORK_ACL_ADMIN.DELETE_PRIVILEGE('power_users.xml',
   'APEX_240100');
   ```

6.13 About Performance Optimization Tasks

Learn about performance optimization.

Performance of web applications heavily depends on their size and how often a browser has to request static content like images, CSS, and JavaScript files. To improve performance, most web servers support on-the-fly HTTP response compression and provide settings that enable you to configure how long browsers can cache a file before requesting it again. The HTTP response compression is usually implemented using gzip encoding, while browser file caching is enabled by issuing Cache-Control HTTP response header.

Please see your web server documentation to learn how to enable response compression and browser file caching. For optimal performance of the Oracle APEX development environment and APEX applications, Oracle recommends enabling gzip compression of files in the virtual images directory (for example, `/i/`) and responses from the database access descriptor in addition to enabling browsers to cache files from the virtual images directory for at least 12 hours.

6.14 Converting Between Runtime and Full Development Environments

Learn about converting between runtime and full development environments.

This section describes how to convert between runtime and full development environments.

- About Runtime and Full Development Environments
  An Oracle APEX runtime environment enables users to run an application without supporting the ability to change or edit the application.
- Converting a Runtime Environment to a Full Development Environment
  Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role and run the `apxdvins.sql`.

- Converting a Full Development Environment to a Runtime Environment
  Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role and run the `apxdevrm.sql`.

6.14.1 About Runtime and Full Development Environments

An Oracle APEX runtime environment enables users to run an application without supporting the ability to change or edit the application.

A runtime environment includes only the packages necessary to run your applications, making it a more hardened environment. It does not provide a web interface for administration.

You administer an APEX runtime environment using SQLcl or SQL Developer and the `APEX_INSTANCE_ADMIN` API.

> **See Also**
>
> - About the APEX Runtime Environment
> - Installing Exported Applications into a Runtime Environment in *Oracle APEX Administration Guide*

6.14.2 Converting a Runtime Environment to a Full Development Environment

Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role and run the `apxdvins.sql`.

To convert an Oracle APEX runtime environment to a full development environment:

1. Change your working directory to the `apex` directory where you unzipped the installation software.
2. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Run `apxdvins.sql`. For example:

   ```
   @apxdvins.sql
   ```

4. Follow the instructions in Creating or Updating Your Instance Administration Account.

See Also

Working with SQLcl in *Oracle SQLcl User's Guide*

6.14.3 Converting a Full Development Environment to a Runtime Environment

Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role and run the `apxdevrm.sql`.

Note

To ensure the security and performance of your development environment, this functionality is not available in APEX instances running in Oracle Cloud.

To convert an APEX full development environment to a runtime environment:

1. Change your working directory to the `apex` directory where you unzipped the installation software.
2. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Run `apxdevrm.sql`. For example:

   ```
   @apxdevrm.sql
   ```

See Also

Working with SQLcl in *Oracle SQLcl User's Guide*

A

Automating the APEX Installation Process

Automate the process of installing and configuring an Oracle APEX instance.

- About apxsilentins.sql
  Run the `apxsilentins.sql` script to automate the installation and configuration of an Oracle APEX instance.
- Running apxsilentins.sql
  Run the `apxsilentins.sql` script.

A.1 About apxsilentins.sql

Run the `apxsilentins.sql` script to automate the installation and configuration of an Oracle APEX instance.

Traditionally you run the run the `apexins.sql` script to install Oracle APEX and then perform a multiple other steps to configure the `APEX_PUBLIC_USER` account. The `apxsilentins.sql` script simplifies the installation and configuation process. `apxsilentins.sql` accepts additional parameters so that passwords can be passed for following database users associated with the Oracle APEX schema: `APEX_PUBLIC_USER`, `APEX_LISTENER`, `APEX_REST_PUBLIC_USER` and the Oracle APEX Instance Administration user, `ADMIN`. You can also use these passwords for the configuration of middle tiers and other processes. `apxsilentins.sql` also completes other installation steps such as creating and setting the password for the Instance Administration user, `ADMIN`, configuring a network ACL, and configuring Oracle REST Data Services.

Running the `apxsilentins.sql` script, removes the need for completing the following topics:

- Installing APEX
- Creating or Updating Your Instance Administration Account
- Configuring the APEX_PUBLIC_USER Account
- Enabling Network Services in Oracle Database
- Configuring Static File Support (`apex_rest_config.sql`)

A.2 Running apxsilentins.sql

Run the `apxsilentins.sql` script.

To run `apxsilentins.sql`:

1. Change your working directory to `apex`.
2. Start SQLcl and connect as user `SYS` to the database where Oracle APEX is installed. You will need to specify the `SYSDBA` role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

3. Disable any existing password complexity rules for the default profile.
4. Run `apxsilentins.sql` passing the following eight arguments in the order shown:

   ```
   @apxsilentins.sql tablespace_apex tablespace_files tablespace_temp images
       password_apex_pub_user password_apex_listener
   password_apex_rest_pub_user
       password_internal_admin
   ```

   Where:

   - *`tablespace_apex`* is the name of the tablespace for the APEX platform schema.
   - *`tablespace_files`* is the name of the tablespace for the APEX files schema.
   - *`tablespace_temp`* is the name of the temporary tablespace or tablespace group.
   - *`images`* is the virtual directory for APEX images. To support future APEX upgrades, define the virtual image directory as `/i/`.
   - `password_apex_pub_user` is the password for the `APEX_PUBLIC_USER` database account.
   - `password_apex_listener` is the password for the `APEX_LISTENER` database account.
   - `password_apex_rest_pub_user` is the password for the `APEX_REST_PUBLIC_USER` database account.
   - `password_internal_admin` is the password for the Instance Administration `ADMIN` APEX account. This password must meet the following requirements:
     - Contain at least 6 characters.
     - Contain at least one numeric character (0123456789).
     - Contain at least one punctuation character (!"#$%&()``*+,-/:;?_).
     - Contain at least one uppercase alphabetic character.

For example:

```
@apxsilentins.sql SYSAUX SYSAUX TEMP /i/ Passw0rd!1 Passw0rd!2 Passw0rd!3
Passw0rd!4
```

Once `apxsilentins.sql` completes, you follow the steps in Downloading and Installing Oracle REST Data Services (ORDS) and Configuring Oracle REST Data Services (except for "Configuring Static File Support").

Use the passwords you supplied to `apxsilentins.sql` when completing these steps. Then, move on to Creating a Workspace and Adding APEX Users.

B

Maximizing Uptime During an APEX Upgrade

Learn how to maximize uptime during an Oracle APEX upgrade.

Previously, APEX could only be upgraded by completely disabling application usage for an extended length of time. The following is an overview of the additional steps you can take to keep your applications usable for end users during most portions of an APEX upgrade.

This advanced procedure is an alternative to the following the topics in Downloading and Installing APEX.

To upgrade the instance, administrators typically run these phases in one step by executing one of the following:

- For full development environment:

  ```
  @apexins.sql tablespace_apex tablespace_files tablespace_temp images
  ```

- For runtime-only environment:

  ```
  @apxrtins.sql tablespace_apex tablespace_files tablespace_temp images
  ```

Where:

- *`tablespace_apex`* is the name of the tablespace for the APEX platform schema.
- *`tablespace_files`* is the name of the tablespace for the APEX files schema.
- *`tablespace_temp`* is the name of the temporary tablespace or tablespace group.
- *`images`* is the virtual directory for APEX images.

The upgrade of an APEX instance runs in four phases:

1. Create database schemas and database objects (tables, packages).
2. Migrate application metadata.
3. Migrate data that runtime applications modify and switch to the new version.
4. Migrate additional log and summary data.

Phases 1 and 4 do not disable end users using the instance. Phase 2 only affects developers who modify applications, and new background processes for Page Processes do not run. Phase 3 affects all access to APEX.

Oracle now also provides alternative upgrade scripts to run the phases independently. Administrators can use these scripts instead of `apexins.sql` and `apxrtins.sql`, to reduce the effective downtime of an APEX instance from potentially hours to just a few minutes (depending on hardware performance).

> **Note**
>
> This feature is not supported when APEX is installed in `CDB$ROOT`.

Administrators must sequentially execute the following scripts to start phases 1, 2 and 3, respectively. At the end of phase 3, a scheduler job automatically starts to execute phase 4.

To reduce downtime during an APEX upgrade:

1. Execute phase 1 script: Development and runtime usage is not affected.
    - For full development environment:

      ```
      @apexins1.sql tablespace_apex tablespace_files tablespace_temp images
      ```

    - For runtime-only environment:

      ```
      @apxrtins1.sql tablespace_apex tablespace_files tablespace_temp images
      ```

   Example:

   ```
   @apexins1.sql sysaux sysaux temp /i/
   ```

2. Execute phase 2 script: Development is disabled, but runtime usage is not affected.
    - For full development environment:

      ```
      @apexins2.sql tablespace_apex tablespace_files tablespace_temp images
      ```

    - For runtime-only environment:

      ```
      @apxrtins2.sql tablespace_apex tablespace_files tablespace_temp images
      ```

   Example:

   ```
   @apexins2.sql sysaux sysaux temp /i/
   ```

3. Disable web access for the web server, Oracle REST Data Services.
4. Execute phase 3 script: APEX can not be used.
    - For full development environment:

      ```
      @apexins3.sql tablespace_apex tablespace_files tablespace_temp images
      ```

    - For runtime-only environment:

      ```
      @apxrtins3.sql tablespace_apex tablespace_files tablespace_temp images
      ```

   Example: `@apexins3.sql sysaux sysaux temp /i/`
5. Install images of the new APEX version in your web server. Administrators can do this while phase 3 is running or even earlier, if the new version's images directory is different to the previous APEX version's (for example: */i212/* for the new version vs. */i211/* for the old version).

   For details refer to the installation instructions for Oracle REST Data Services.
6. Re-enable web access for the web server and restart Oracle REST Data Services.

After web access is restarted, developers and users can access the instance again, while phase 4 finishes in the background.

See Also

Installing and Configuring APEX and Oracle REST Data Services

C

APEX Installation Troubleshooting

Learn about troubleshooting Oracle APEX Installation.

This section contains information on troubleshooting.

- Reviewing a Log of an Installation Session
  The `apexins.sql` script creates a log file in the `apex` directory using the naming convention `installYYYY-MM-DD_HH24-MI-SS.log`.
- Verifying the Validity of an APEX Installation
  Verify the validity of an Oracle APEX installation by running a query.
- Cleaning Up After a Failed Installation
  Learn about best practices for troubleshooting and cleaning up after a failed installation.
- About Images Displaying Incorrectly in APEX
  Learn about troubleshooting if images in Oracle APEX do not display correctly.
- About Page Protection Violation
  A page protection violation may be caused by manual alteration of protected page items.

> **See Also**
>
> Upgrading from a Previous APEX Release

C.1 Reviewing a Log of an Installation Session

The `apexins.sql` script creates a log file in the `apex` directory using the naming convention `installYYYY-MM-DD_HH24-MI-SS.log`.

In a successful installation, the log file contains the following text:

```
Thank you for installing Oracle APEX.
```

```
Oracle APEX is installed in the APEX_240200 schema.
```

If the log file contains a few errors, it does not mean that your installation failed. Note that acceptable errors are noted as such in the log file.

C.2 Verifying the Validity of an APEX Installation

Verify the validity of an Oracle APEX installation by running a query.

You can verify the validity of an APEX installation by running the following query:

```
SELECT STATUS FROM DBA_REGISTRY
WHERE COMP_ID = 'APEX';
```

If the result is `VALID`, you can assume the installation was successful.

C.3 Cleaning Up After a Failed Installation

Learn about best practices for troubleshooting and cleaning up after a failed installation.

In a successful installation the following banner displays near the end of the installation:

```
Thank you for installing Oracle APEX.
Oracle APEX is installed in the APEX_240200 schema.
```

To reinstall, you must either drop the APEX database schemas, or run a script to completely remove APEX from the database, depending upon the installation type.

- Reverting to a Previous Release After a Failed Upgrade Installation
  Learn about reverting to Oracle APEX to a previous release in the case of a failed upgrade installation.
- Removing APEX from the Database
  Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role and execute the `SQL> @apxremov.sql` command.

C.3.1 Reverting to a Previous Release After a Failed Upgrade Installation

Learn about reverting to Oracle APEX to a previous release in the case of a failed upgrade installation.

In the case of a failed upgrade installation, you may want to revert Oracle APEX to a previous release and then remove the schemas associated with the current release.

- Verifying If You Have a Previous Release of APEX
  Run a query to verify if you have previous release of Oracle APEX.
- Reverting the Images Directory
  If you altered your images directory, revert it back to the release you want to revert to. You must point the text alias `/i/` back to images directory for the release you want to revert to.
- Reverting to a Previous Release
  Learn how to revert to a previous release Oracle APEX.
- Removing the APEX Release Schema
  After you revert to the prior release, remove the Oracle APEX schema.

C.3.1.1 Verifying If You Have a Previous Release of APEX

Run a query to verify if you have previous release of Oracle APEX.

To verify whether you have a previous release of APEX:

1. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role. For example:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

2. Execute the following command in SQLcl:

   ```
   select username from dba_users
       where regexp_like(username,'(FLOWS|APEX)_\d{6}')
           and username <> (select table_owner from all_synonyms
           where synonym_name = 'WWV_FLOW'
           and owner = 'PUBLIC')
   ```

   If the query above returns any rows, the database contains a previous release of APEX.

C.3.1.2 Reverting the Images Directory

If you altered your images directory, revert it back to the release you want to revert to. You must point the text alias `/i/` back to images directory for the release you want to revert to.

See Also

Copying the Images Directory

C.3.1.3 Reverting to a Previous Release

Learn how to revert to a previous release Oracle APEX.

- Reverting to Release 1.5
- Reverting to Release 1.6
- Reverting to Release 2.0
- Reverting to Release 2.2
- Reverting to Release 3.0
- Reverting to Release 3.1
- Reverting to Release 3.2
- Reverting to Release 4.0
- Reverting to Release 4.1
- Reverting to Release 4.2 in a non-CDB or PDB with Local APEX
- Reverting to Release 4.2 in a CDB
- Reverting to Release 5.0 in a non-CDB or PDB with Local APEX
- Reverting to Release 5.0 in a CDB
- Reverting to Release 5.1 in a non-CDB or PDB with Local APEX
- Reverting to Release 5.1 in a CDB
- Reverting to Release 18.1 in a non-CDB or PDB with Local APEX

- Reverting to Release 18.1 in a CDB
- Reverting to Release 18.2 in a non-CDB or PDB with Local APEX
- Reverting to Release 18.2 in a CDB
- Reverting to Release 19.1 in a non-CDB or PDB with Local APEX
- Reverting to Release 19.1 in a CDB
- Reverting to Release 19.2 in a non-CDB or PDB with Local APEX
- Reverting to Release 19.2 in a CDB
- Reverting to Release 20.1 in a non-CDB or PDB with Local APEX
- Reverting to Release 20.1 in a CDB
- Reverting to Release 20.2 in a non-CDB or PDB with Local APEX
- Reverting to Release 20.2 in a CDB
- Reverting to Release 21.1 in a non-CDB or PDB with Local APEX
- Reverting to Release 21.1 in a CDB
- Reverting to Release 21.2 in a non-CDB or PDB with Local APEX
- Reverting to Release 21.2 in a CDB
- Reverting to Release 22.1 in a non-CDB or PDB with Local APEX
- Reverting to Release 22.1 in a CDB
- Reverting to Release 22.2 in a non-CDB or PDB with Local APEX
- Reverting to Release 22.2 in a CDB
- Reverting to Release 23.1 in a non-CDB or PDB with Local APEX
- Reverting to Release 23.1 in a CDB
- Reverting to Release 23.2 in a non-CDB or PDB with Local APEX
- Reverting to Release 23.2 in a CDB
- Reverting to Release 24.1 in a non-CDB or PDB with Local APEX
- Reverting to Release 24.1 in a CDB
- Re-enabling the REST Administration Interface After Downgrading

C.3.1.3.1 Reverting to Release 1.5

To revert to Oracle APEX release 1.5:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

3. Execute the following:

   ```
   ALTER SESSION SET CURRENT_SCHEMA = FLOWS_010500;
   exec
   flows_010500.wwv_flow_upgrade.switch_schemas('APEX_240200','FLOWS_010500');
   ```

4. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.2 Reverting to Release 1.6

To revert to Oracle APEX release 1.6:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Execute the following:

   ```
   ALTER SESSION SET CURRENT_SCHEMA = FLOWS_010600;
   exec
   flows_010600.wwv_flow_upgrade.switch_schemas('APEX_240200','FLOWS_010600');
   ```

4. Depending upon the release you are reverting to, execute the appropriate command in SQLcl.
5. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.3 Reverting to Release 2.0

To revert to Oracle APEX release 2.0:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Execute the following:

   ```
   ALTER SESSION SET CURRENT_SCHEMA = FLOWS_020000;
   exec
   flows_020000.wwv_flow_upgrade.switch_schemas('APEX_240200','FLOWS_020000');
   ```

4. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.4 Reverting to Release 2.2

To revert to Oracle APEX release 2.2:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Execute the following:

```
ALTER SESSION SET CURRENT_SCHEMA = FLOWS_020200;
exec
flows_020200.wwv_flow_upgrade.switch_schemas('APEX_240200','FLOWS_020200');
```

4. Remove the APEX release schema. See Removing the APEX Release Schema.

> **See Also**
>
> Removing the APEX Release Schema

C.3.1.3.5 Reverting to Release 3.0

To revert to Oracle APEX release 3.0:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 3.0 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Execute the following:

```
set define '^'

@apexvalidate x x FLOWS_030000
ALTER SESSION SET CURRENT_SCHEMA = FLOWS_030000;
exec
flows_030000.wwv_flow_upgrade.switch_schemas('APEX_240200','FLOWS_030000');
ALTER SESSION SET CURRENT_SCHEMA = SYS;
declare
    l_apex_version varchar2(30);
begin
    sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
    l_apex_version := flows_030000.wwv_flows_release;
    dbms_registry.downgrading('APEX','Oracle Application
Express','validate_apex','FLOWS_030000');
    dbms_registry.downgraded('APEX',l_apex_version);
    validate_apex;
```

```
end;
/
```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

> **See Also**
>
> Reverting the Images Directory

C.3.1.3.6 Reverting to Release 3.1

To revert to Oracle APEX release 3.1:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex/core` in the 3.1 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Execute the following commands:

   ```
   @wwv_flow_val.plb
   @wwv_dbms_sql.sql
   @wwv_dbms_sql.plb
   ```

5. Change your working directory to `apex` in the 3.1 source.
6. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

7. Execute the following:

```
set define '^'

@apexvalidate x x FLOWS_030100
ALTER SESSION SET CURRENT_SCHEMA = FLOWS_030100;
exec
flows_030100.wwv_flow_upgrade.switch_schemas('APEX_240200','FLOWS_030100');
ALTER SESSION SET CURRENT_SCHEMA = SYS;
declare
    l_apex_version varchar2(30);
begin
    sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
    l_apex_version := flows_030100.wwv_flows_release;
    dbms_registry.downgrading('APEX','Oracle Application
Express','validate_apex','FLOWS_030100');
    dbms_registry.downgraded('APEX',l_apex_version);
    validate_apex;
end;
/
```

8. Remove the APEX release schema. See Removing the APEX Release Schema.

> **See Also**
>
> Reverting the Images Directory

C.3.1.3.7 Reverting to Release 3.2

To revert to Oracle APEX release 3.2:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex/core` in the 3.2 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Execute the following commands:

   ```
   @wwv_flow_val.plb
   @wwv_dbms_sql.sql
   @wwv_dbms_sql.plb
   ```

5. Start SQLcl and connect to the database where APEX is installed as SYS specifying the SYSDBA role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

6. Execute the following:

   ```
   set define '^'

   @apexvalidate x x APEX_030200
   ALTER SESSION SET CURRENT_SCHEMA = APEX_030200;
   exec
   apex_030200.wwv_flow_upgrade.switch_schemas('APEX_240200','APEX_030200');
   ALTER SESSION SET CURRENT_SCHEMA = SYS;
   declare
       l_apex_version varchar2(30);
   begin
       sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
       l_apex_version := apex_030200.wwv_flows_release;
       dbms_registry.downgrading('APEX','Oracle Application
   Express','validate_apex','APEX_030200')
       dbms_registry.downgraded('APEX',l_apex_version);
       validate_apex;
   end;
   /
   ```

7. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.8 Reverting to Release 4.0

To revert to Oracle APEX release 4.0:

1. If you altered your images directory, revert it back to the release you want to revert to.

2. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
    - On Windows:

      ```
      SYSTEM_DRIVE:\ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

    - On UNIX and Linux:

      ```
      $ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

3. Execute the following commands:

    ```
    @wwv_flow_val.sql
    @wwv_flow_val.plb
    @wwv_dbms_sql.sql
    @wwv_dbms_sql.plb
    ```

4. Change your working directory to `apex` in the 4.0 source.
5. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
    - On Windows:

      ```
      SYSTEM_DRIVE:\ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

    - On UNIX and Linux:

      ```
      $ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

6. Execute the following:

    ```
    set define '^'

    @apexvalidate x x APEX_040000
    ALTER SESSION SET CURRENT_SCHEMA = APEX_040000;
    exec
    apex_040000.wwv_flow_upgrade.switch_schemas('APEX_240200','APEX_040000');
    ALTER SESSION SET CURRENT_SCHEMA = SYS;
    declare
        l_apex_version varchar2(30);
    begin
        sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
        l_apex_version := apex_040000.wwv_flows_release;
        dbms_registry.downgrading('APEX','Oracle Application
    Express','validate_apex','APEX_040000');
        dbms_registry.downgraded('APEX',l_apex_version);
    ```

```
    validate_apex;
end;
/
```

7. Remove the APEX release schema. See Removing the APEX Release Schema.

> **See Also**
>
> Reverting the Images Directory

C.3.1.3.9 Reverting to Release 4.1

To revert to Oracle APEX release 4.1:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex/core` in the 4.1 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
    - On Windows:

      ```
      SYSTEM_DRIVE:\ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

    - On UNIX and Linux:

      ```
      $ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

4. Execute the following commands:

   ```
   @wwv_flow_val.sql
   @wwv_flow_val.plb
   @wwv_dbms_sql.sql
   @wwv_dbms_sql.plb
   ```

5. Change your working directory to `apex` in the 4.1 source.
6. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
    - On Windows:

      ```
      SYSTEM_DRIVE:\ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

  - On UNIX and Linux:

    ```
    $ sql /nolog
    SQL> CONNECT SYS as SYSDBA
    Enter password: SYS_password
    ```

7. Execute the following:

   ```
   set define '^'

   @apexvalidate x x APEX_040100
   ALTER SESSION SET CURRENT_SCHEMA = APEX_040100;
   exec
   apex_040100.wwv_flow_upgrade.switch_schemas('APEX_240200','APEX_040100');
   ALTER SESSION SET CURRENT_SCHEMA = SYS;
   declare
       l_apex_version varchar2(30);
   begin
       sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
       l_apex_version := apex_040100.wwv_flows_release;
       dbms_registry.downgrading('APEX','Oracle Application
   Express','validate_apex','APEX_040100');
       dbms_registry.downgraded('APEX',l_apex_version);
       validate_apex;
   end;
   /
   ```

8. Remove the APEX release schema. See Removing the APEX Release Schema.

> **See Also**
>
> Reverting the Images Directory

C.3.1.3.10 Reverting to Release 4.2 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX release 4.2 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex/core` in the 4.2 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

4. Execute the following commands:

   ```
   alter session set current_schema = SYS;

   @core_sys_views.sql

   grant select on sys.wwv_flow_gv$session to APEX_040200;

   @wwv_flow_val.sql
   @wwv_flow_val.plb
   @wwv_dbms_sql.sql
   grant execute on wwv_dbms_sql to APEX_040200;
   @wwv_dbms_sql.plb

   begin
       dbms_utility.compile_schema('APEX_040200');
   end;
   /
   ```

5. Change your working directory to `apex` in the 4.2 source.
6. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

7. Execute the following:

   ```
   set define '^'

   @apexvalidate x x APEX_040200
   ALTER SESSION SET CURRENT_SCHEMA = APEX_040200;
   exec
   apex_040200.wwv_flow_upgrade.switch_schemas('APEX_240200','APEX_040200');
   ALTER SESSION SET CURRENT_SCHEMA = SYS;
   declare
       l_apex_version varchar2(30);
   begin
       sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
       l_apex_version := apex_040200.wwv_flows_release;
   ```

```
    dbms_registry.downgrading('APEX','Oracle Application 
Express','validate_apex','APEX_040200');
    dbms_registry.downgraded('APEX',l_apex_version);
    validate_apex;
end;
/
```

8. Remove the APEX release schema. See Removing the APEX Release Schema.

> **See Also**
>
> Reverting the Images Directory

C.3.1.3.11 Reverting to Release 4.2 in a CDB

To revert to Oracle APEX release 4.2 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex/core` in the 4.2 source.
3. Create a new text file in that directory named `apx42dgrd1.sql` consisting of the following:

```
alter session set current_schema = SYS;

@core_sys_views.sql

grant select on sys.wwv_flow_gv$session to APEX_040200;

@wwv_flow_val.sql
@wwv_flow_val.plb
@wwv_dbms_sql.sql
grant execute on wwv_dbms_sql to APEX_040200;
@wwv_dbms_sql.plb

begin
    dbms_utility.compile_schema('APEX_040200');
end;
/
```

4. Create a second new text file in that directory named `apx42dgrd.sql` consisting of the following:

```
set define '^'

whenever sqlerror exit

column :xe_home new_value OH_HOME NOPRINT
variable xe_home varchar2(255)

set serverout on
begin
-- get oracle_home
    sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
    if length(:xe_home) = 0 then
```

```
        sys.dbms_output.put_line(lpad('-',80,'-'));
        raise_application_error (
            -20001,
            'Oracle Home environment variable not set' );
    end if;
end;
/
whenever sqlerror continue

set termout off
select :xe_home from sys.dual;
set termout on

host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
catcon.pl -b apx42dgrd apx42dgrd1.sql
```

5. Start SQLcl and connect to `CDB$ROOT` of the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

6. Execute the following commands:

   ```
   @apx42dgrd.sql
   ```

7. Change your working directory to `apex` in the 4.2 source.
8. Create a new text file in that directory name `apx42dgrd1.sql` with the following contents:

   ```
   set define '^'

   ALTER SESSION SET CURRENT_SCHEMA = SYS;

   @apexvalidate x x APEX_040200

   ALTER SESSION SET CURRENT_SCHEMA = APEX_040200;
   exec
   apex_040200.wwv_flow_upgrade.switch_schemas('APEX_240200','APEX_040200');
   ALTER SESSION SET CURRENT_SCHEMA = SYS;
   declare
       l_apex_version varchar2(30);
   begin
       sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
       l_apex_version := apex_040200.wwv_flows_release;
       dbms_registry.downgrading('APEX','Oracle Application
   Express','validate_apex','APEX_040200');
   ```

```
    dbms_registry.downgraded('APEX',l_apex_version);
    validate_apex;
end;
/
```

9. Create a second new text file in that directory named `apx42dgrd.sql` consisting of the following:

   ```
   set define '^'

   whenever sqlerror exit

   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)

   set serverout on
   begin
   -- get oracle_home
       sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
       if length(:xe_home) = 0 then
           sys.dbms_output.put_line(lpad('-',80,'-'));
           raise_application_error (
               -20001,
               'Oracle Home environment variable not set' );
       end if;
   end;
   /
   whenever sqlerror continue

   set termout off
   select :xe_home from sys.dual;
   set termout on

   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b apx42dgrd apx42dgrd1.sql
   ```

10. Start SQLcl and connect to `CDB$ROOT` of the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
    - On Windows:

      ```
      SYSTEM_DRIVE:\ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

    - On UNIX and Linux:

      ```
      $ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

11. Execute the following:

    ```
    @apx42dgrd.sql
    ```

12. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.12 Reverting to Release 5.0 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX release 5.0 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex/core` in the 5.0 source.
3. Start SQLcl and connect to the database where APEX is installed as SYS specifying the SYSDBA role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Execute the following commands:

   ```
   alter session set current_schema = SYS;

   @wwv_flow_val.sql
   @wwv_flow_val.plb

   begin
        dbms_utility.compile_schema('APEX_050000');
   end;
   /

   set define '^'
   @validate_apex x x APEX_050000

   begin
     for i in ( select owner, trigger_name
                  from sys.dba_triggers
                 where owner        =    'APEX_050000'
                   and trigger_name like 'WWV_FLOW_UPGRADE_%'
                 order by 1 )
     loop
       sys.dbms_output.put_line('Dropping trigger '||i.owner||'.'||
   i.trigger_name);
       execute immediate 'drop trigger '||i.owner||'.'||i.trigger_name;
     end loop;
   end;
   /
   ```

```
ALTER SESSION SET CURRENT_SCHEMA = APEX_050000;
exec
apex_050000.wwv_flow_upgrade.switch_schemas('APEX_240200','APEX_050000');
ALTER SESSION SET CURRENT_SCHEMA = SYS;
drop context APEX$SESSION;
create context APEX$SESSION using APEX_050000.WWV_FLOW_SESSION_CONTEXT;
declare
     l_apex_version varchar2(30);
     l_schemas sys.dbms_registry.schema_list_t;
begin
     sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
     l_apex_version := apex_050000.wwv_flows_release;
     dbms_registry.downgrading('APEX','Oracle Application
Express','validate_apex','APEX_050000');
     dbms_registry.downgraded('APEX',l_apex_version);
     select username
       bulk collect into l_schemas
       from all_users
      where username in
('FLOWS_FILES','APEX_PUBLIC_USER','APEX_LISTENER','APEX_REST_PUBLIC_USER','
APEX_INSTANCE_ADMIN_USER')
      order by 1;
     sys.dbms_registry.update_schema_list('APEX', l_schemas);
     validate_apex;
end;
/
```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.13 Reverting to Release 5.0 in a CDB

To revert to Oracle APEX release 5.0 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex/core` in the 5.0 source.
3. Create a new text file in that directory named `apx50dgrd1.sql` consisting of the following:

```
alter session set current_schema = SYS;

@wwv_flow_val.sql
@wwv_flow_val.plb

begin
     dbms_utility.compile_schema('APEX_050000');
end;
/

set define '^'
```

```
@validate_apex x x APEX_050000

begin
  for i in ( select owner, trigger_name
               from sys.dba_triggers
              where owner        =    'APEX_050000'
                and trigger_name like 'WWV_FLOW_UPGRADE_%'
              order by 1 )
  loop
    sys.dbms_output.put_line('Dropping trigger '||i.owner||'.'||
i.trigger_name);
    execute immediate 'drop trigger '||i.owner||'.'||i.trigger_name;
  end loop;
end;
/

ALTER SESSION SET CURRENT_SCHEMA = APEX_050000;
exec
apex_050000.wwv_flow_upgrade.switch_schemas('APEX_240200','APEX_050000');
ALTER SESSION SET CURRENT_SCHEMA = SYS;
drop context APEX$SESSION;
create context APEX$SESSION using APEX_050000.WWV_FLOW_SESSION_CONTEXT;
declare
     l_apex_version varchar2(30);
     l_schemas sys.dbms_registry.schema_list_t;
begin
     sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
     l_apex_version := apex_050000.wwv_flows_release;
     dbms_registry.downgrading('APEX','Oracle Application
Express','validate_apex','APEX_050000');
     dbms_registry.downgraded('APEX',l_apex_version);
     select username
       bulk collect into l_schemas
       from all_users
      where username in
('FLOWS_FILES','APEX_PUBLIC_USER','APEX_LISTENER','APEX_REST_PUBLIC_USER','
APEX_INSTANCE_ADMIN_USER')
      order by 1;
     sys.dbms_registry.update_schema_list('APEX', l_schemas);
     validate_apex;
end;
/
```

4. Create a second new text file in that directory named `apx50dgrd.sql` consisting of the following:

```
set define '^'
whenever sqlerror exit
column :xe_home new_value OH_HOME NOPRINT
variable xe_home varchar2(255)
set serverout on
begin
-- get oracle_home
sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
if length(:xe_home) = 0 then
    sys.dbms_output.put_line(lpad('-',80,'-'));
    raise_application_error (-20001,'Oracle Home environment variable not
set' );
end if;
end;
/
whenever sqlerror continue
set termout off
select :xe_home from sys.dual;
set termout on
host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
catcon.pl -b apx50dgrd apx50dgrd1.sql
```

5. Start SQLcl and connect to `CDB$ROOT` of the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

6. Execute the following commands:

   ```
   @apx50dgrd.sql
   ```

7. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.14 Reverting to Release 5.1 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX release 5.1 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex/core` in the 5.1 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

4. Execute the following commands:

   ```
   alter session set current_schema = SYS;

   @wwv_flow_val.sql
   @wwv_flow_val.plb

   begin
        dbms_utility.compile_schema('APEX_050100');
   end;
   /

   set define '^'
   @validate_apex x x APEX_050100

   begin
     for i in ( select owner, trigger_name
                  from sys.dba_triggers
                 where owner        =    'APEX_050100'
                   and trigger_name like 'WWV_FLOW_UPGRADE_%'
                 order by 1 )
     loop
       sys.dbms_output.put_line('Dropping trigger '||i.owner||'.'||
   i.trigger_name);
       execute immediate 'drop trigger '||i.owner||'.'||i.trigger_name;
     end loop;
   end;
   /

   ALTER SESSION SET CURRENT_SCHEMA = APEX_050100;
   exec
   apex_050100.wwv_flow_upgrade.switch_schemas('APEX_240200','APEX_050100');
   ALTER SESSION SET CURRENT_SCHEMA = SYS;
   drop context APEX$SESSION;
   create context APEX$SESSION using APEX_050100.WWV_FLOW_SESSION_CONTEXT;
   declare
        l_apex_version varchar2(30);
        l_schemas sys.dbms_registry.schema_list_t;
   begin
        sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
        l_apex_version := apex_050100.wwv_flows_release;
        dbms_registry.downgrading('APEX','Oracle Application
   Express','validate_apex','APEX_050100');
        dbms_registry.downgraded('APEX',l_apex_version);
        select username
          bulk collect into l_schemas
          from all_users
         where username in
   ('FLOWS_FILES','APEX_PUBLIC_USER','APEX_LISTENER','APEX_REST_PUBLIC_USER','
   APEX_INSTANCE_ADMIN_USER')
   ```

```
         order by 1;
        sys.dbms_registry.update_schema_list('APEX', l_schemas);
        validate_apex;
    end;
    /
```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.15 Reverting to Release 5.1 in a CDB

To revert to Oracle APEX release 5.1 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex/core` in the 5.1 source.
3. Create a new text file in that directory named `apx51dgrd1.sql` consisting of the following:

```
alter session set current_schema = SYS;

@wwv_flow_val.sql
@wwv_flow_val.plb

begin
     dbms_utility.compile_schema('APEX_050100');
end;
/

set define '^'
@validate_apex x x APEX_050100

begin
  for i in ( select owner, trigger_name
               from sys.dba_triggers
              where owner        =    'APEX_050100'
                and trigger_name like 'WWV_FLOW_UPGRADE_%'
              order by 1 )
  loop
    sys.dbms_output.put_line('Dropping trigger '||i.owner||'.'||
i.trigger_name);
    execute immediate 'drop trigger '||i.owner||'.'||i.trigger_name;
  end loop;
end;
/

ALTER SESSION SET CURRENT_SCHEMA = APEX_050100;
exec
apex_050100.wwv_flow_upgrade.switch_schemas('APEX_240200','APEX_050100');
ALTER SESSION SET CURRENT_SCHEMA = SYS;
drop context APEX$SESSION;
```

```
create context APEX$SESSION using APEX_050100.WWV_FLOW_SESSION_CONTEXT;
declare
     l_apex_version varchar2(30);
     l_schemas sys.dbms_registry.schema_list_t;
begin
     sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
     l_apex_version := apex_050100.wwv_flows_release;
     dbms_registry.downgrading('APEX','Oracle Application
Express','validate_apex','APEX_050100');
     dbms_registry.downgraded('APEX',l_apex_version);
     select username
       bulk collect into l_schemas
       from all_users
      where username in
('FLOWS_FILES','APEX_PUBLIC_USER','APEX_LISTENER','APEX_REST_PUBLIC_USER','
APEX_INSTANCE_ADMIN_USER')
      order by 1;
     sys.dbms_registry.update_schema_list('APEX', l_schemas);
     validate_apex;
end;
/
```

4. Create a second new text file in that directory named `apx51dgrd.sql` consisting of the following:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)
   set serverout on
   begin
   -- get oracle_home
   sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
   if length(:xe_home) = 0 then
       sys.dbms_output.put_line(lpad('-',80,'-'));
       raise_application_error (-20001,'Oracle Home environment variable not
   set' );
   end if;
   end;
   /
   whenever sqlerror continue
   set termout off
   select :xe_home from sys.dual;
   set termout on
   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b apx51dgrd apx51dgrd1.sql
   ```

5. Start SQLcl and connect to `CDB$ROOT` of the database where APEX is installed as `SYS` specifying the `SYSDBA` role:

   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

6. Execute the following commands:

   ```
   @apx51dgrd.sql
   ```

7. Remove the APEX release schema. See Removing the APEX Release Schema.

> **See Also**
>
> Reverting the Images Directory

C.3.1.3.16 Reverting to Release 18.1 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX release 18.1 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 18.1 source.
3. Create a new text file in that directory named `apxdwngrd.sql` consisting of the following:

   ```
   set define '^'
   set concat on
   set concat .
   set verify off

   set termout off
   column foo new_val LOG
   select 'apxdwngrd_' ||to_char(sysdate, 'YYYY-MM-DD_HH24-MI-SS') || '.log'
   as foo
     from sys.dual;
   set termout on
   spool ^LOG

   @@core/scripts/set_appun.sql

   whenever sqlerror exit
   set serveroutput on size unlimited

   declare
       l_cnt    number := 0;
   begin
       select count(*) into l_cnt from sys.dba_users where username =
   '^APPUN';
       if l_cnt = 0 then
           dbms_output.put_line('^APPUN not found in this database.');
           raise program_error;
       end if;
   end;
   /
   ```

```
whenever sqlerror continue


prompt ...Create validate procedure in SYS schema and start registration
@@core/validate_apex.sql x x ^APPUN

grant inherit any privileges to ^APPUN;

prompt Installing SYS views

@@core/sys_core_views.sql

@@core/wwv_flow_val.sql
grant execute on sys.wwv_flow_val to ^APPUN.;

@@core/wwv_flow_val.plb

ALTER SESSION SET CURRENT_SCHEMA = ^APPUN;

exec sys.dbms_session.modify_package_state(sys.dbms_session.reinitialize);

begin
    ^APPUN..wwv_flow_upgrade.remove_jobs();
    ^APPUN..wwv_flow_upgrade.create_jobs('^APPUN');
    ^APPUN..wwv_flow_upgrade.create_public_synonyms('^APPUN');
    ^APPUN..wwv_flow_upgrade.grant_public_synonyms('^APPUN');
    ^APPUN..wwv_flow_upgrade.flows_files_objects_remove('^APPUN');
    ^APPUN..wwv_flow_upgrade.flows_files_objects_create('^APPUN');
end;
/

ALTER SESSION SET CURRENT_SCHEMA = SYS;

drop context APEX$SESSION;
create context APEX$SESSION using ^APPUN..WWV_FLOW_SESSION_CONTEXT;

alter package sys.wwv_dbms_sql_^APPUN. compile;
alter package sys.wwv_dbms_sql_^APPUN. compile body;

exec sys.dbms_session.modify_package_state(sys.dbms_session.reinitialize);

set serveroutput on size unlimited

declare
     l_apex_version varchar2(30);
     l_schemas sys.dbms_registry.schema_list_t;
begin
    sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
    execute immediate 'drop package ^APPUN..WWV_FLOW_DB_VERSION';
    l_apex_version := ^APPUN..wwv_flows_release;
    sys.dbms_registry.loading('APEX','Oracle Application
Express','validate_apex', '^APPUN');
    select username
      bulk collect into l_schemas
      from sys.all_users
     where username in
```

```
('FLOWS_FILES','APEX_PUBLIC_USER','APEX_LISTENER','APEX_REST_PUBLIC_USER','
APEX_INSTANCE_ADMIN_USER')
     order by 1;
     sys.dbms_registry.update_schema_list('APEX', l_schemas);
     sys.dbms_registry.loaded('APEX',l_apex_version);
     commit;
     sys.validate_apex;
end;
/
```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.17 Reverting to Release 18.1 in a CDB

To revert to Oracle APEX release 18.1 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 18.1 source.
3. Create a new text file in that directory named `apxdwngrd.sql` consisting of the following:

   ```
   set define '^'
   set concat on
   set concat .
   set verify off

   set termout off
   column foo new_val LOG
   select 'apxdwngrd_' ||to_char(sysdate, 'YYYY-MM-DD_HH24-MI-SS') || '.log'
   as foo
     from sys.dual;
   ```

```
set termout on
spool ^LOG

@@core/scripts/set_appun.sql

whenever sqlerror exit
set serveroutput on size unlimited

declare
    l_cnt    number := 0;
begin
    select count(*) into l_cnt from sys.dba_users where username =
'^APPUN';
    if l_cnt = 0 then
        dbms_output.put_line('^APPUN not found in this database.');
        raise program_error;
    end if;
end;
/
whenever sqlerror continue


prompt ...Create validate procedure in SYS schema and start registration
@@core/validate_apex.sql x x ^APPUN

grant inherit any privileges to ^APPUN;

prompt Installing SYS views

@@core/sys_core_views.sql

@@core/wwv_flow_val.sql
grant execute on sys.wwv_flow_val to ^APPUN.;

@@core/wwv_flow_val.plb

ALTER SESSION SET CURRENT_SCHEMA = ^APPUN;

exec sys.dbms_session.modify_package_state(sys.dbms_session.reinitialize);

begin
    ^APPUN..wwv_flow_upgrade.remove_jobs();
    ^APPUN..wwv_flow_upgrade.create_jobs('^APPUN');
    ^APPUN..wwv_flow_upgrade.create_public_synonyms('^APPUN');
    ^APPUN..wwv_flow_upgrade.grant_public_synonyms('^APPUN');
    ^APPUN..wwv_flow_upgrade.flows_files_objects_remove('^APPUN');
    ^APPUN..wwv_flow_upgrade.flows_files_objects_create('^APPUN');
end;
/

ALTER SESSION SET CURRENT_SCHEMA = SYS;

drop context APEX$SESSION;
create context APEX$SESSION using ^APPUN..WWV_FLOW_SESSION_CONTEXT;

alter package sys.wwv_dbms_sql_^APPUN. compile;
alter package sys.wwv_dbms_sql_^APPUN. compile body;

exec sys.dbms_session.modify_package_state(sys.dbms_session.reinitialize);

set serveroutput on size unlimited

declare
     l_apex_version varchar2(30);
     l_schemas sys.dbms_registry.schema_list_t;
begin
    sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
    execute immediate 'drop package ^APPUN..WWV_FLOW_DB_VERSION';
    l_apex_version := ^APPUN..wwv_flows_release;
    sys.dbms_registry.loading('APEX','Oracle Application 
Express','validate_apex', '^APPUN');
    select username
      bulk collect into l_schemas
      from sys.all_users
     where username in 
('FLOWS_FILES','APEX_PUBLIC_USER','APEX_LISTENER','APEX_REST_PUBLIC_USER','
APEX_INSTANCE_ADMIN_USER')
     order by 1;
     sys.dbms_registry.update_schema_list('APEX', l_schemas);
     sys.dbms_registry.loaded('APEX',l_apex_version);
     commit;
     sys.validate_apex;
end;
/
```

4. Create a second new text file in that directory named `apxdwngrd_cdb.sql` consisting of the following:

```
set define '^'
whenever sqlerror exit
column :xe_home new_value OH_HOME NOPRINT
variable xe_home varchar2(255)
set serverout on
begin
-- get oracle_home
sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
if length(:xe_home) = 0 then
    sys.dbms_output.put_line(lpad('-',80,'-'));
    raise_application_error (-20001,'Oracle Home environment variable not 
set' );
end if;
end;
/
whenever sqlerror continue
set termout off
select :xe_home from sys.dual;
set termout on
host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
catcon.pl -b apx181dgrd apxdwngrd.sql
```

5. Start SQLcl and connect to `CDB$ROOT` of the database where APEX is installed as `SYS` specifying the `SYSDBA` role:

- On Windows:

  ```
  SYSTEM_DRIVE:\ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

6. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

7. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.18 Reverting to Release 18.2 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX release 18.2 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 18.2 source.
3. Create a new text file in that directory named `apxdwngrd.sql` consisting of the following:

   ```
   set define '^'
   set concat on
   set concat .
   set verify off

   set termout off
   column foo new_val LOG
   select 'apxdwngrd_' ||to_char(sysdate, 'YYYY-MM-DD_HH24-MI-SS') || '.log'
   as foo
     from sys.dual;
   set termout on
   spool ^LOG

   @@core/scripts/set_appun.sql

   whenever sqlerror exit
   set serveroutput on size unlimited

   declare
       l_cnt    number := 0;
   begin
       select count(*) into l_cnt from sys.dba_users where username =
   ```

```
'^APPUN';
    if l_cnt = 0 then
        dbms_output.put_line('^APPUN not found in this database.');
        raise program_error;
    end if;
end;
/
whenever sqlerror continue


prompt ...Create validate procedure in SYS schema and start registration
@@core/validate_apex.sql x x ^APPUN

grant inherit any privileges to ^APPUN;

prompt Installing SYS views

@@core/sys_core_views.sql

@@core/wwv_flow_val.sql
grant execute on sys.wwv_flow_val to ^APPUN.;

@@core/wwv_flow_val.plb

ALTER SESSION SET CURRENT_SCHEMA = ^APPUN;

exec sys.dbms_session.modify_package_state(sys.dbms_session.reinitialize);

begin
    ^APPUN..wwv_flow_upgrade.remove_jobs();
    ^APPUN..wwv_flow_upgrade.create_jobs('^APPUN');
    ^APPUN..wwv_flow_upgrade.create_public_synonyms('^APPUN');
    ^APPUN..wwv_flow_upgrade.grant_public_synonyms('^APPUN');
    ^APPUN..wwv_flow_upgrade.flows_files_objects_remove('^APPUN');
    ^APPUN..wwv_flow_upgrade.flows_files_objects_create('^APPUN');
end;
/

ALTER SESSION SET CURRENT_SCHEMA = SYS;

drop context APEX$SESSION;
create context APEX$SESSION using ^APPUN..WWV_FLOW_SESSION_CONTEXT;

alter package sys.wwv_dbms_sql_^APPUN. compile;
alter package sys.wwv_dbms_sql_^APPUN. compile body;

exec sys.dbms_session.modify_package_state(sys.dbms_session.reinitialize);

set serveroutput on size unlimited

declare
     l_apex_version varchar2(30);
     l_schemas sys.dbms_registry.schema_list_t;
begin
    sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
    execute immediate 'drop package ^APPUN..WWV_FLOW_DB_VERSION';
    l_apex_version := ^APPUN..wwv_flows_release;
    sys.dbms_registry.loading('APEX','Oracle Application 
Express','validate_apex', '^APPUN');
    select username
      bulk collect into l_schemas
      from sys.all_users
     where username in 
('FLOWS_FILES','APEX_PUBLIC_USER','APEX_LISTENER','APEX_REST_PUBLIC_USER','
APEX_INSTANCE_ADMIN_USER')
     order by 1;
     sys.dbms_registry.update_schema_list('APEX', l_schemas);
     sys.dbms_registry.loaded('APEX',l_apex_version);
     commit;
     sys.validate_apex;
end;
/
```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

> **See Also**
>
> Reverting the Images Directory

C.3.1.3.19 Reverting to Release 18.2 in a CDB

To revert to Oracle APEX release 18.2 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 18.2 source.
3. Create a new text file in that directory named `apxdwngrd.sql` consisting of the following:

   ```
   set define '^'
   set concat on
   set concat .
   ```

```
set verify off

set termout off
column foo new_val LOG
select 'apxdwngrd_' ||to_char(sysdate, 'YYYY-MM-DD_HH24-MI-SS') || '.log'
as foo
  from sys.dual;
set termout on
spool ^LOG

@@core/scripts/set_appun.sql

whenever sqlerror exit
set serveroutput on size unlimited

declare
    l_cnt    number := 0;
begin
    select count(*) into l_cnt from sys.dba_users where username =
'^APPUN';
    if l_cnt = 0 then
        dbms_output.put_line('^APPUN not found in this database.');
        raise program_error;
    end if;
end;
/
whenever sqlerror continue


prompt ...Create validate procedure in SYS schema and start registration
@@core/validate_apex.sql x x ^APPUN

grant inherit any privileges to ^APPUN;

prompt Installing SYS views

@@core/sys_core_views.sql

@@core/wwv_flow_val.sql
grant execute on sys.wwv_flow_val to ^APPUN.;

@@core/wwv_flow_val.plb

ALTER SESSION SET CURRENT_SCHEMA = ^APPUN;

exec sys.dbms_session.modify_package_state(sys.dbms_session.reinitialize);

begin
    ^APPUN..wwv_flow_upgrade.remove_jobs();
    ^APPUN..wwv_flow_upgrade.create_jobs('^APPUN');
    ^APPUN..wwv_flow_upgrade.create_public_synonyms('^APPUN');
    ^APPUN..wwv_flow_upgrade.grant_public_synonyms('^APPUN');
    ^APPUN..wwv_flow_upgrade.flows_files_objects_remove('^APPUN');
    ^APPUN..wwv_flow_upgrade.flows_files_objects_create('^APPUN');
end;
/
```

```
ALTER SESSION SET CURRENT_SCHEMA = SYS;

drop context APEX$SESSION;
create context APEX$SESSION using ^APPUN..WWV_FLOW_SESSION_CONTEXT;

alter package sys.wwv_dbms_sql_^APPUN. compile;
alter package sys.wwv_dbms_sql_^APPUN. compile body;

exec sys.dbms_session.modify_package_state(sys.dbms_session.reinitialize);

set serveroutput on size unlimited

declare
     l_apex_version varchar2(30);
     l_schemas sys.dbms_registry.schema_list_t;
begin
    sys.dbms_registry.set_session_namespace (namespace => 'DBTOOLS');
    execute immediate 'drop package ^APPUN..WWV_FLOW_DB_VERSION';
    l_apex_version := ^APPUN..wwv_flows_release;
    sys.dbms_registry.loading('APEX','Oracle Application
    select username
      bulk collect into l_schemas
      from sys.all_users
     where username in
('FLOWS_FILES','APEX_PUBLIC_USER','APEX_LISTENER','APEX_REST_PUBLIC_USER','
APEX_INSTANCE_ADMIN_USER')
     order by 1;
     sys.dbms_registry.update_schema_list('APEX', l_schemas);
     sys.dbms_registry.loaded('APEX',l_apex_version);
     commit;
     sys.validate_apex;
end;
/
```

4. Create a second new text file in that directory named `apxdwngrd_cdb.sql` consisting of the following:

```
set define '^'
whenever sqlerror exit
column :xe_home new_value OH_HOME NOPRINT
variable xe_home varchar2(255)
set serverout on
begin
-- get oracle_home
sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
if length(:xe_home) = 0 then
    sys.dbms_output.put_line(lpad('-',80,'-'));
    raise_application_error (-20001,'Oracle Home environment variable not
set' );
end if;
end;
/
whenever sqlerror continue
set termout off
select :xe_home from sys.dual;
```

```
set termout on
host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
catcon.pl -b apx182dgrd apxdwngrd.sql
```

5. Start SQLcl and connect to `CDB$ROOT` of the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

6. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

7. Remove the APEX release schema. See Removing the APEX Release Schema.

See Also

Reverting the Images Directory

C.3.1.3.20 Reverting to Release 19.1 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX release 19.1 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 19.1 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.21 Reverting to Release 19.1 in a CDB

To revert to Oracle APEX release 19.1 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 19.1 source.
3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)
   set serverout on
   begin
   -- get oracle_home
   sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
   if length(:xe_home) = 0 then
       sys.dbms_output.put_line(lpad('-',80,'-'));
       raise_application_error (-20001,'Oracle Home environment variable not
   set' );
   end if;
   end;
   /
   whenever sqlerror continue
   set termout off
   select :xe_home from sys.dual;
   set termout on
   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b apx191dgrd apxdwngrd.sql
   ```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.22 Reverting to Release 19.2 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX release 19.2 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 19.2 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.23 Reverting to Release 19.2 in a CDB

To revert to Oracle APEX release 19.2 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 19.2 source.
3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)
   set serverout on
   begin
   -- get oracle_home
   sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
   if length(:xe_home) = 0 then
       sys.dbms_output.put_line(lpad('-',80,'-'));
       raise_application_error (-20001,'Oracle Home environment variable not
   set' );
   end if;
   end;
   /
   whenever sqlerror continue
   set termout off
   ```

```
select :xe_home from sys.dual;
set termout on
host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
catcon.pl -b apx192dgrd apxdwngrd.sql
```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.24 Reverting to Release 20.1 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX release 20.1 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 20.1 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.25 Reverting to Release 20.1 in a CDB

To revert to Oracle APEX release 20.1 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 20.1 source.
3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)
   set serverout on
   begin
   -- get oracle_home
   sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
   if length(:xe_home) = 0 then
       sys.dbms_output.put_line(lpad('-',80,'-'));
       raise_application_error (-20001,'Oracle Home environment variable not
   set' );
   end if;
   end;
   /
   whenever sqlerror continue
   set termout off
   select :xe_home from sys.dual;
   set termout on
   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b apx201dgrd apxdwngrd.sql
   ```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.26 Reverting to Release 20.2 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX release 20.2 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 20.2 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.27 Reverting to Release 20.2 in a CDB

To revert to Oracle APEX release 20.2 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 20.2 source.
3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)
   set serverout on
   begin
   -- get oracle_home
   sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
   if length(:xe_home) = 0 then
       sys.dbms_output.put_line(lpad('-',80,'-'));
       raise_application_error (-20001,'Oracle Home environment variable not
   set' );
   end if;
   end;
   /
   whenever sqlerror continue
   set termout off
   select :xe_home from sys.dual;
   set termout on
   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b apx202dgrd apxdwngrd.sql
   ```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.28 Reverting to Release 21.1 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX 21.1 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 21.1 source.
3. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.29 Reverting to Release 21.1 in a CDB

To revert to Oracle APEX release 21.1 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 21.1 source.

3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

```
set define '^'
whenever sqlerror exit
column :xe_home new_value OH_HOME NOPRINT
variable xe_home varchar2(255)
set serverout on
begin
-- get oracle_home
sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
if length(:xe_home) = 0 then
    sys.dbms_output.put_line(lpad('-',80,'-'));
    raise_application_error (-20001,'Oracle Home environment variable not
set' );
end if;
end;
/
whenever sqlerror continue
set termout off
select :xe_home from sys.dual;
set termout on
host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
catcon.pl -b apx211dgrd apxdwngrd.sql
```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.30 Reverting to Release 21.2 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX 21.2 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 21.2 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:

- On Windows:

  ```
  SYSTEM_DRIVE:\ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.31 Reverting to Release 21.2 in a CDB

To revert to Oracle APEX release 21.2 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 21.2 source.
3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)
   set serverout on
   begin
   -- get oracle_home
   sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
   if length(:xe_home) = 0 then
       sys.dbms_output.put_line(lpad('-',80,'-'));
       raise_application_error (-20001,'Oracle Home environment variable not
   set' );
   end if;
   end;
   /
   whenever sqlerror continue
   set termout off
   select :xe_home from sys.dual;
   set termout on
   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b apx212dgrd apxdwngrd.sql
   ```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:

- On Windows:

  ```
  SYSTEM_DRIVE:\ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.32 Reverting to Release 22.1 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX 22.1 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 22.1 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.33 Reverting to Release 22.1 in a CDB

To revert to Oracle APEX release 22.1 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 22.1 source.

3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)
   set serverout on
   begin
   -- get oracle_home
   sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
   if length(:xe_home) = 0 then
       sys.dbms_output.put_line(lpad('-',80,'-'));
       raise_application_error (-20001,'Oracle Home environment variable not
   set' );
   end if;
   end;
   /
   whenever sqlerror continue
   set termout off
   select :xe_home from sys.dual;
   set termout on
   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b apx212dgrd apxdwngrd.sql
   ```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.34 Reverting to Release 22.2 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX 22.2 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 22.2 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:

- On Windows:

  ```
  SYSTEM_DRIVE:\ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.35 Reverting to Release 22.2 in a CDB

To revert to Oracle APEX release 22.2 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 22.2 source.
3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)
   set serverout on
   begin
   -- get oracle_home
   sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
   if length(:xe_home) = 0 then
       sys.dbms_output.put_line(lpad('-',80,'-'));
       raise_application_error (-20001,'Oracle Home environment variable not
   set' );
   end if;
   end;
   /
   whenever sqlerror continue
   set termout off
   select :xe_home from sys.dual;
   set termout on
   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b apx222dgrd apxdwngrd.sql
   ```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:

  - On Windows:

    ```
    SYSTEM_DRIVE:\ sql /nolog
    SQL> CONNECT SYS as SYSDBA
    Enter password: SYS_password
    ```

  - On UNIX and Linux:

    ```
    $ sql /nolog
    SQL> CONNECT SYS as SYSDBA
    Enter password: SYS_password
    ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.36 Reverting to Release 23.1 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX 23.1 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 23.1 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.37 Reverting to Release 23.1 in a CDB

To revert to Oracle APEX release 23.1 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 23.1 source.

3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)
   set serverout on
   begin
   -- get oracle_home
   sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
   if length(:xe_home) = 0 then
       sys.dbms_output.put_line(lpad('-',80,'-'));
       raise_application_error (-20001,'Oracle Home environment variable not
   set' );
   end if;
   end;
   /
   whenever sqlerror continue
   set termout off
   select :xe_home from sys.dual;
   set termout on
   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b apx231dgrd apxdwngrd.sql
   ```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.38 Reverting to Release 23.2 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX 23.2 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 23.2 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:

- On Windows:

  ```
  SYSTEM_DRIVE:\ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.39 Reverting to Release 23.2 in a CDB

To revert to Oracle APEX release 23.2 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 23.2 source.
3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)
   set serverout on
   begin
   -- get oracle_home
   sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
   if length(:xe_home) = 0 then
       sys.dbms_output.put_line(lpad('-',80,'-'));
       raise_application_error (-20001,'Oracle Home environment variable not
   set' );
   end if;
   end;
   /
   whenever sqlerror continue
   set termout off
   select :xe_home from sys.dual;
   set termout on
   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b apx232dgrd apxdwngrd.sql
   ```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:

- On Windows:

  ```
  SYSTEM_DRIVE:\ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.40 Reverting to Release 24.1 in a non-CDB or PDB with Local APEX

To revert to Oracle APEX 24.1 in a non-CDB or PDB with a locally installed APEX:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 24.1 source.
3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

4. Run the `apxdwngrd.sql` script:

   ```
   SQL> @apxdwngrd.sql
   ```

5. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.41 Reverting to Release 24.1 in a CDB

To revert to Oracle APEX release 24.1 in a CDB:

1. If you altered your images directory, revert it back to the release you want to revert to.
2. Change your working directory to `apex` in the 24.1 source.

3. Create a script in the apex directory called `apxdwngrd_cdb.sql` with the following contents:

```
set define '^'
whenever sqlerror exit
column :xe_home new_value OH_HOME NOPRINT
variable xe_home varchar2(255)
set serverout on
begin
-- get oracle_home
sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
if length(:xe_home) = 0 then
    sys.dbms_output.put_line(lpad('-',80,'-'));
    raise_application_error (-20001,'Oracle Home environment variable not
set' );
end if;
end;
/
whenever sqlerror continue
set termout off
select :xe_home from sys.dual;
set termout on
host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
catcon.pl -b apx241dgrd apxdwngrd.sql
```

4. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

5. Run the `apxdwngrd_cdb.sql` script:

   ```
   SQL> @apxdwngrd_cdb.sql
   ```

6. Remove the APEX release schema. See Removing the APEX Release Schema.

C.3.1.3.42 Re-enabling the REST Administration Interface After Downgrading

If the REST Administration Interface was used before the upgrade attempt, you must re-create the `APEX_INSTANCE_ADMIN_USER`. If the REST Administration Interface was not used, skip this step.

To re-create the `APEX_INSTANCE_ADMIN_USER`:

1. Change your working directory to `apex` in the, *XX.X* release source (where *XX.X* is the release number you reverted to).

2. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role:
    - On Windows:

      ```
      SYSTEM_DRIVE:\ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

    - On UNIX and Linux:

      ```
      $ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

3. If the no authentication was used, run the following:

   ```
   create user apex_instance_admin_user no authentication
   ```

4. If the authentication was used, run the following:

   ```
   create user apex_instance_admin_user identified by <random-password>
   password expire
   ```

C.3.1.4 Removing the APEX Release Schema

After you revert to the prior release, remove the Oracle APEX schema.

- Removing the APEX Release 24.2 Schema from a Non-CDB
  Start SQLcl and connect to the database and execute `DROP USER APEX_240200 CASCADE;` command.
- Removing the APEX Release 24.2 Schema from a CDB
  Create text files, start SQLcl and connect to the database execute `@remove_apx242_usr.sql`.

C.3.1.4.1 Removing the APEX Release 24.2 Schema from a Non-CDB

Start SQLcl and connect to the database and execute `DROP USER APEX_240200 CASCADE;` command.

To remove the release 24.2 schema from a non-CDB:

1. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
    - On Windows:

      ```
      SYSTEM_DRIVE:\ sql /nolog
      SQL> CONNECT SYS as SYSDBA
      Enter password: SYS_password
      ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

2. Execute the following command:

   ```
   DROP USER APEX_240200 CASCADE;
   ```

   Once you have removed the APEX 24.2 schema, you can now attempt the upgrade again.

C.3.1.4.2 Removing the APEX Release 24.2 Schema from a CDB

Create text files, start SQLcl and connect to the database execute `@remove_apx242_usr.sql`.

To remove the release 24.2 schema from a CDB:

1. Create a new text file named `remove_apx242_usr1.sql` with the following contents:

   ```
   alter session set current_schema = SYS;
   drop user APEX_240200 cascade;
   ```

2. Create a second new text file named `remove_apx242_usr.sql` with the following contents:

   ```
   set define '^'
   whenever sqlerror exit
   column :xe_home new_value OH_HOME NOPRINT
   variable xe_home varchar2(255)

   set serverout on
   begin
   -- get oracle_home
       sys.dbms_system.get_env('ORACLE_HOME',:xe_home);
       if length(:xe_home) = 0 then
           sys.dbms_output.put_line(lpad('-',80,'-'));
           raise_application_error (
               -20001,
               'Oracle Home environment variable not set' );
       end if;
   end;
   /
   whenever sqlerror continue

   set termout off
   select :xe_home from sys.dual;
   set termout on
   host ^OH_HOME/perl/bin/perl -I ^OH_HOME/rdbms/admin ^OH_HOME/rdbms/admin/
   catcon.pl -b
         remove_apx242_usr remove_apx242_usr.sql
   ```

3. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:

- On Windows:

  ```
  SYSTEM_DRIVE:\ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

- On UNIX and Linux:

  ```
  $ sql /nolog
  SQL> CONNECT SYS as SYSDBA
  Enter password: SYS_password
  ```

4. Execute the following command:

   ```
   @remove_apx242_usr.sql
   ```

   Once you have removed the APEX 24.2 schema, you can now attempt the upgrade again.

C.3.2 Removing APEX from the Database

Start SQLcl and connect to the database where Oracle APEX is installed as `SYS` specifying the `SYSDBA` role and execute the `SQL> @apxremov.sql` command.

This section describes how to remove the APEX schema, synonyms, and users from the database without deleting the database.

> **Note**
>
> Do NOT follow these steps if you have upgraded your database from a prior release, and still want to use the prior release of APEX. For information about reverting to a prior release, see Reverting to a Previous Release. If you are not sure whether you have completed a new installation or an upgrade installation, review Cleaning Up After a Failed Installation to verify if a previous release of APEX exists in the database.

To remove APEX from the database:

1. Change your working directory to the `apex` directory where you unzipped the APEX software.
2. Start SQLcl and connect to the database where APEX is installed as `SYS` specifying the `SYSDBA` role:
   - On Windows:

     ```
     SYSTEM_DRIVE:\ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

   - On UNIX and Linux:

     ```
     $ sql /nolog
     SQL> CONNECT SYS as SYSDBA
     Enter password: SYS_password
     ```

3. Execute the following command:

   ```
   SQL> @apxremov.sql
   ```

4. After successfully removing APEX using `apxremov.sql`, you must exit your current SQLcl session and reconnect before attempting another install using `apexins.sql`.

C.4 About Images Displaying Incorrectly in APEX

Learn about troubleshooting if images in Oracle APEX do not display correctly.

If images in APEX do not display correctly, you may have more than one definition of the `/i/` alias. To address this issue:

- If possible, rename the first instance of `/i/` to a different alias name.
- Alternatively, copy the images from the directory where APEX was downloaded or the images copied for Oracle REST Data Services (ORDS) to the directory defined by the first `/i/` alias.

C.5 About Page Protection Violation

A page protection violation may be caused by manual alteration of protected page items.

If this error occurs after installation when trying to log into Oracle APEX, then stop and start Oracle REST Data Services. If you are unsure of what caused this error, contact the application administrator for assistance.

Index