



Oracle Application Express: Developing Database Web Applications

Hands-On-Labs Guide

Unit 16: Migrating Application Development Between Environments

This exercise includes two hands-on-labs and uses the Demo Projects application.

HOL 16-1 Importing an Application: In this lab, you import an application along with the underlying database objects and seed data.

HOL 16-2 Migrating your Application Development Between Environments: In this lab, you export an application and then use SQL Workshop to export database objects and seed data.

Steps 3 through 15 in this lab are optional. You need a target APEX environment to perform these steps. To test these steps you might want to use a different Workspace. If you want to import the application in to the same Workspace, then you might have to choose a different application ID. Installing database objects and seed data will still fail, as they are already created.

HOL 16-1: Importing an Application

In this lab, you import an application in to your Workspace. Along with the application definition, you also install the supporting objects.

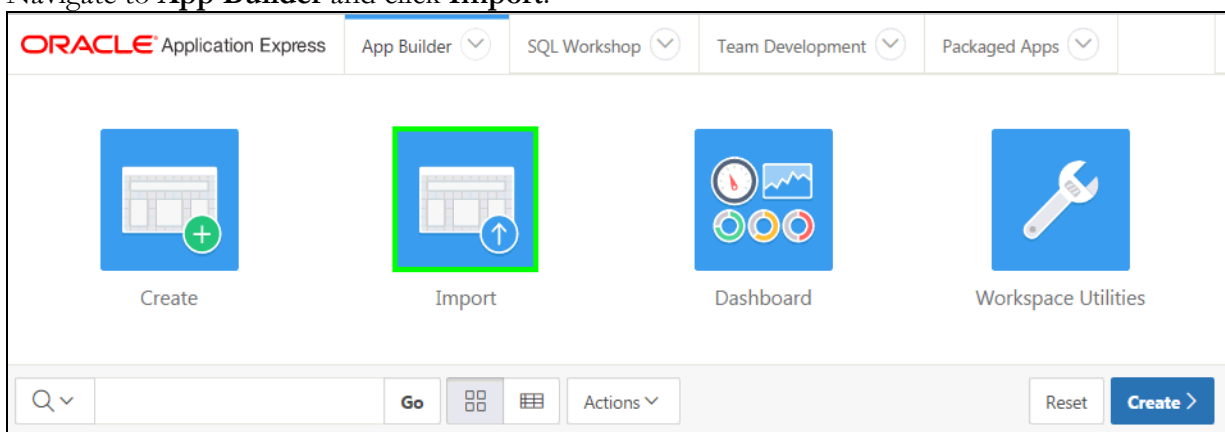
This lab uses the application export file **demo projects app export-unit 15.sql** which is the application export file for HOL 15-5. If you have not performed any of the previous hands-on labs in this course, you can use the steps in this lab to create the corresponding application. You simply need to replace the application export file name in Step 2 with the one appropriate for that particular hands-on-lab.

Note:

- The steps in this lab assume that you are importing an application within the same Workspace that you used in the previous hands-on labs. If you are importing the application in to a different Workspace (either on the same or different instance), the steps might slightly differ.
- This lab assumes that you want to reuse the same application ID from the export file.
- The underlying database objects and seed data for this application are packaged along with the application definition in a single file. After installing the application, you can view the installation scripts for the database objects and data by clicking **Supporting Objects** on the application home page.
- Installation of supporting objects and seed data may fail if you already have the database objects created in your schema. If you do not need your existing objects and seed data, then you can simply run the **drop_objects.sql** before proceeding with the steps in this lab. However, note that this script will completely delete all of your underlying database objects along with the data.

Note: If you want to take a backup of your existing application along with the database objects and seed data, perform steps 1 and 2 in HOL 16-2.

1. Navigate to **App Builder** and click **Import**.



2. Click **Choose File**.

Navigate to your working directory and double-click the **demo projects app export-unit 15.sql** file. Then, click **Next**.

Note: To import any other APEX application export in to your Workspace, you select that file in this step.

Import

Select the file you wish to import to the export repository. Once imported, you can install your file.

If the imported file is a packaged application export, the installation wizard will allow you to run the packaged installation scripts after installing the application definition.

* Import file demo project...-unit 15.sql ?

* File Type:

- Database Application, Page or Component Export ?
- Worksheet Application Export
- Plug-in
- Theme Export
- User Interface Defaults
- Team Development Feedback
- CSS Export [Deprecated]
- Image Export [Deprecated]
- File Export [Deprecated]

File Character Set ?

3. On the File Import Confirmation page, click **Next**.

4. For Install As Application, you can choose Auto Assign New Application ID. The steps might slightly differ if you select this option.

However, in this lab, you select **Reuse Application ID 103 from Export File**.

Click **Install Application**.

✔ ✔ ● **Install**

When you install an application having the same ID as an existing application in the current workspace, the existing application is deleted and then replaced by the new application. If you attempt to install an application having the same ID as an existing application in a different workspace, a benign error message displays. If you are importing a packaged Application Express application, the installation wizard will allow you to install supporting objects.

Current Workspace: **DEMO** ?

Export File Workspace: **DEMO** ?

Export File Workspace ID: **1841547862027812** ?

Export File Application ID: **103** ?

Export File Version: **2016.08.24** ?

Export File Parsing Schema: **APEX_DEMO** ?

Application Origin: **This application was exported from the current workspace.** ?

* Parsing Schema: ?

* Build Status: ?


* Install As Application: Auto Assign New Application ID ?
 Reuse Application ID 103 From Export File
 Change Application ID

> **Tasks**

Cancel
Install Application

5. Click **Replace Application**.

If you select Auto Assign New Application ID in Step 4, you do not see this warning.



Confirm Replace Application

You have requested that the existing application **103 - Demo Projects** be removed and replaced by the application to be installed: **103 - Demo Projects**. Please confirm this request.

Before replacing an application, consider creating a backup.

Cancel
Replace Application

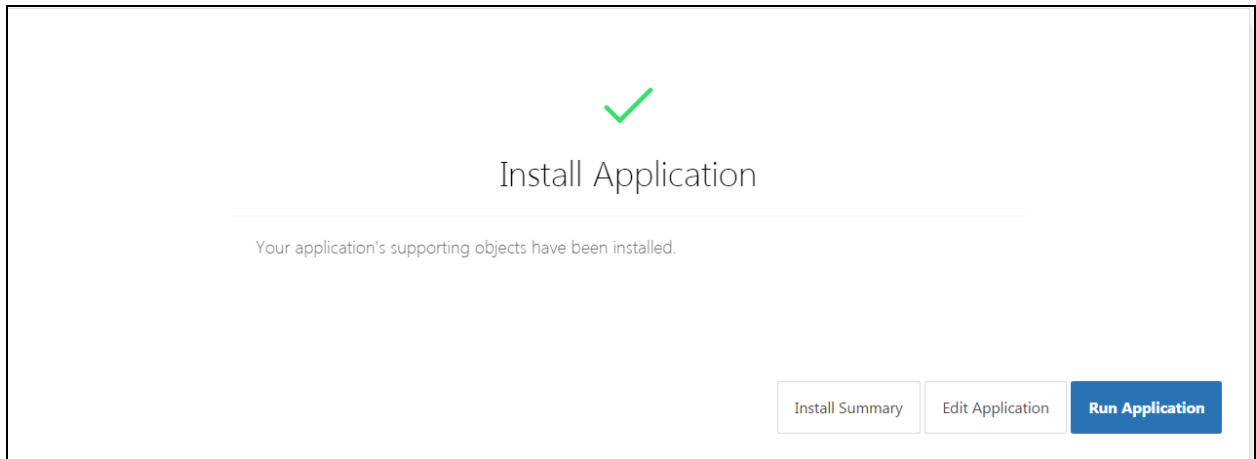
- On the Supporting Objects page, click **Next**.

The screenshot shows the 'Install Application' wizard at the 'Supporting Objects' step. A progress indicator at the top shows a blue circle under 'Supporting Objects' and a grey circle under 'Confirmation'. The main text reads: 'This application installer will guide you through the process of creating your database objects and seed data.' Below this, the following configuration is shown: Application: 103 - Demo Projects (?), Parsing Schema: APEX_DEMO (?), Free Space Required in KB: 0 (?), and Install Supporting Objects: Yes (?). A 'Tasks' link is visible on the left. At the bottom, there is a 'Cancel' button on the left and a 'Next >' button on the right, which is highlighted with a green border.

- On the Confirmation page, click **Install**.

The screenshot shows the 'Install Application' wizard at the 'Confirmation' step. The progress indicator at the top shows a green checkmark under 'Supporting Objects' and a blue circle under 'Confirmation'. The main text reads: 'Please confirm that you would like to install this application's supporting objects.' Below this, there is a 'Tasks' link on the left. At the bottom, there is a '< Cancel' button on the left and an 'Install' button on the right, which is highlighted with a green border.

- The application is successfully installed now. Click **Run Application**.



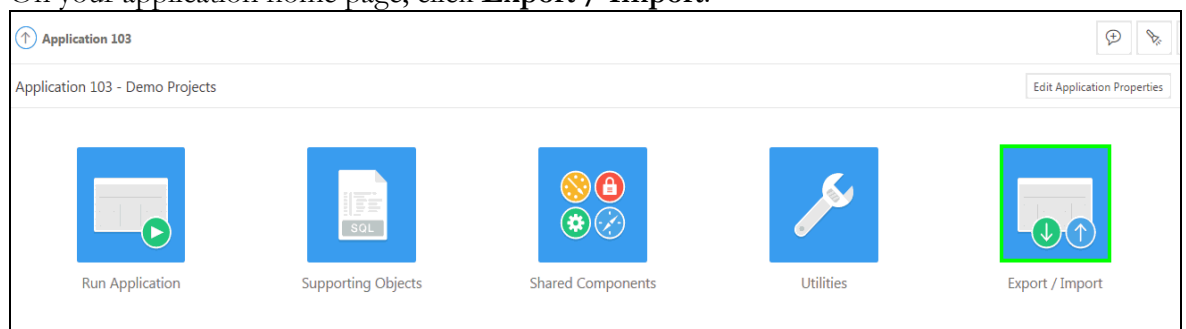
Note: The installation of database objects and seed data may succeed or fail, depending on what database objects are already created. If installation fails, click **Install Summary**, and review the errors. The errors should relate to objects already existing, such as ORA-00955: name is already used by an existing object.

HOL 16-2: Migrating your Application Development between Environments

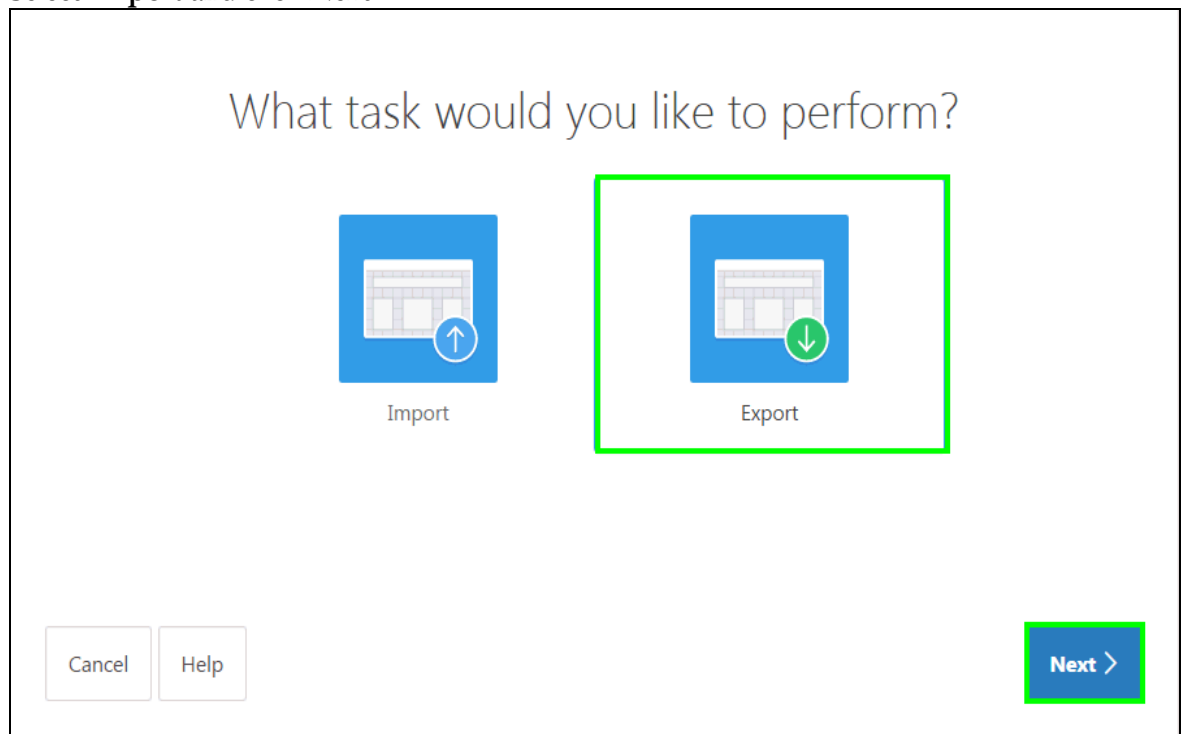
In this lab, you export an application definition, underlying database objects along with the seed data.

Steps 3 through 15 are optional. After exporting application from the current development environment, you log in to the target Application Express environment, import the application and then load the tables along with data.

1. From your current development environment, export an application. Navigate to App Builder and in the report, select the application you want to export. In this lab, you export the Demo Projects application that you created until HOL 15-5. Alternatively, you can export the application that you imported in HOL 16-1.
 - a) On your application home page, click **Export / Import**.



- b) Select **Export** and click Next.



- c) Make sure you select **Yes** for Export Private Reports and Export with Original IDs. Click **Export**.

Export Application

Choose Application

* Application: 103 Demo Projects

Export Application

Selected Application: Demo Projects

Page Count: 16

Owner: APEX_DEMO

File Format: UNIX

Owner Override:

Build Status Override: Run and Build Application

Debugging: No

As of: minutes ago (~ 5 min delay)

File Character Set: Unicode UTF-8

Export Preferences

Export Supporting Object Definitions: Yes

Export Public Reports: Yes No

Export Private Reports: Yes No

Export Report Subscriptions: Yes No

Export Developer Comments: Yes No

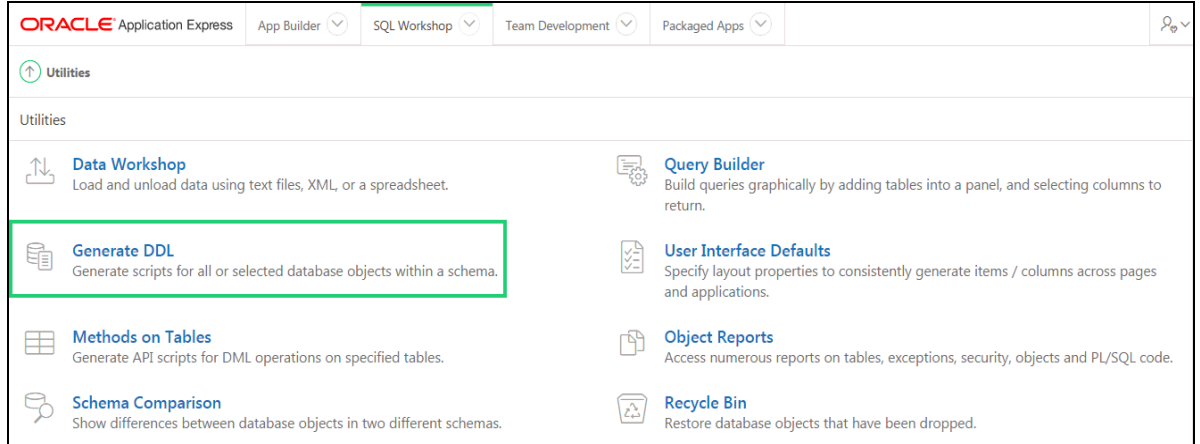
Export Translations: Yes No

Export with Original IDs: Yes No

- d) The application export file is saved in your local directory as a .sql file. You might want to rename the file.
Note: Depending on the browser, if you see a Save dialog, click **Save**.

2. Now, you export database objects and data. Perform the following steps:

- a) In your Workspace, click **SQL Workshop > Utilities**.
Under Utilities, select **Generate DDL**.



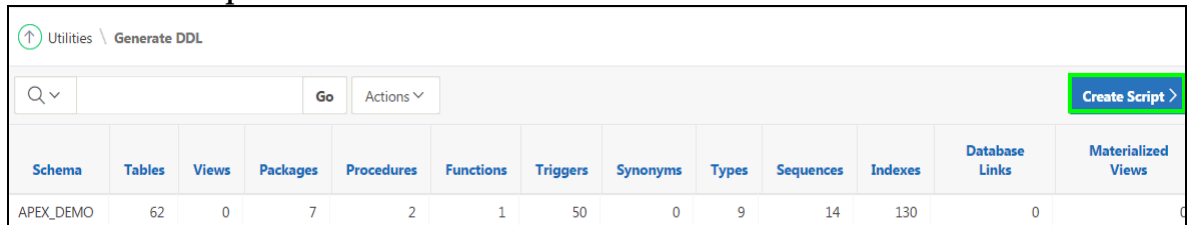
ORACLE Application Express App Builder SQL Workshop Team Development Packaged Apps

Utilities

Utilities

- Data Workshop**
Load and unload data using text files, XML, or a spreadsheet.
- Generate DDL**
Generate scripts for all or selected database objects within a schema.
- Methods on Tables**
Generate API scripts for DML operations on specified tables.
- Schema Comparison**
Show differences between database objects in two different schemas.
- Query Builder**
Build queries graphically by adding tables into a panel, and selecting columns to return.
- User Interface Defaults**
Specify layout properties to consistently generate items / columns across pages and applications.
- Object Reports**
Access numerous reports on tables, exceptions, security, objects and PL/SQL code.
- Recycle Bin**
Restore database objects that have been dropped.

- b) Click **Create Script**.



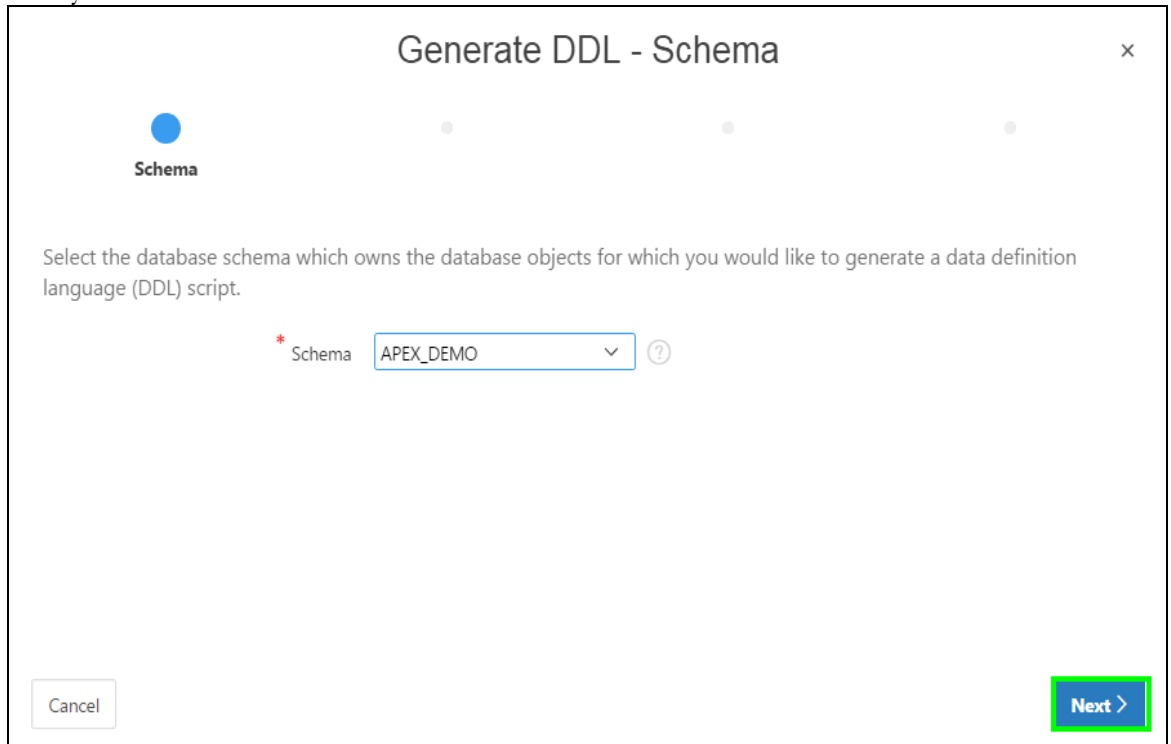
Utilities \ Generate DDL

Q Go Actions

Create Script >

Schema	Tables	Views	Packages	Procedures	Functions	Triggers	Synonyms	Types	Sequences	Indexes	Database Links	Materialized Views
APEX_DEMO	62	0	7	2	1	50	0	9	14	130	0	

- c) Verify the Schema and click **Next**.



Generate DDL - Schema

Schema

Select the database schema which owns the database objects for which you would like to generate a data definition language (DDL) script.

* Schema APEX_DEMO

Cancel **Next >**

- d) For Output, select **Save As Script File**.
For Object Type, select **Table** and click **Next**.

The screenshot shows the 'Generate DDL - Object Type' dialog box. The title bar reads 'Generate DDL - Object Type'. Below the title bar, there are four progress indicators, with the second one (labeled 'Object Type') being active. The main text says: 'Select the object types for which you would like to generate DDL. Clicking **Generate DDL** generates DDL for the selected object types. To select object names for selected object types, click **Next**.'

The 'Output' section has two radio buttons: 'Display Inline' (unselected) and 'Save As Script File' (selected). There is a help icon (?) next to 'Save As Script File'.

The 'Check All' section has a checkbox (unselected) and a help icon (?) next to it.

The 'Object Type' section has a grid of checkboxes for various object types: Function, Index, Package, Procedure, Sequence, Synonym, **Table** (checked and highlighted with a green box), Trigger, View, Database Link, Type, and Materialized View. There is a help icon (?) to the right of the grid.

The 'File Character Set' section has a dropdown menu set to 'Unicode UTF-8' and a help icon (?) next to it.

At the bottom, there are two buttons on the left: '<' and 'Cancel'. On the right, there are two buttons: 'Generate DDL' and 'Next >', with the 'Next >' button highlighted in green.

- e) Select all of the relevant tables for this application.
Select the following tables:
- **DEMO_PROJECTS**
 - **DEMO_PROJ_COMMENTS**
 - **DEMO_PROJ_CONSTRAINTS**
 - **DEMO_PROJ_MILESTONES**
 - **DEMO_PROJ_STATUS**
 - **DEMO_PROJ_TASKS**
 - **DEMO_PROJ_TASK_LINKS**
 - **DEMO_PROJ_TASK_TODOS**
 - **DEMO_PROJ_TEAM_MEMBERS**

Click Generate **DDL**.

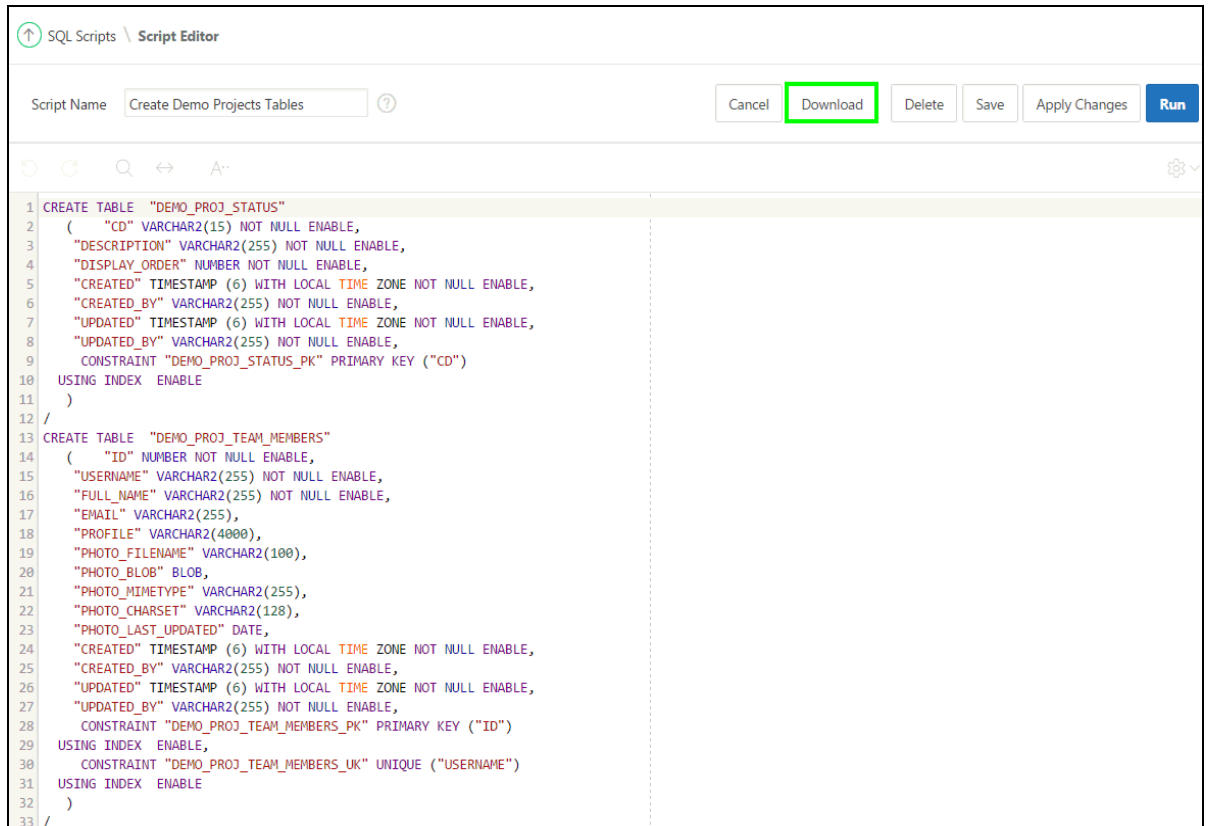
The screenshot shows a dialog box titled "Generate DDL - Object Name". It contains a table with the following data:

		Object Name
<input type="checkbox"/>		DEMO_PRODUCT_INFO
<input checked="" type="checkbox"/>		DEMO_PROJECTS
<input checked="" type="checkbox"/>		DEMO_PROJ_COMMENTS
<input checked="" type="checkbox"/>		DEMO_PROJ_CONSTRAINTS
<input checked="" type="checkbox"/>		DEMO_PROJ_MILESTONES
<input checked="" type="checkbox"/>		DEMO_PROJ_STATUS
<input checked="" type="checkbox"/>		DEMO_PROJ_TASKS
<input checked="" type="checkbox"/>		DEMO_PROJ_TASK_LINKS
<input checked="" type="checkbox"/>		DEMO_PROJ_TASK_TODOS
<input checked="" type="checkbox"/>		DEMO_PROJ_TEAM_MEMBERS
<input type="checkbox"/>		DEMO_STATES

At the bottom left, there are buttons for "<" and "Cancel". At the bottom right, there is a "Generate DDL" button highlighted with a green border.

- f) For Script Name, enter a meaningful name, for example **Create Demo Project Tables**. Optionally enter a description. Click **Create Script**.
- g) The DDL is now saved as a script under SQL Scripts. Click the **Edit** icon (pencil) on the recently created script.

h) Click **Download**.



The screenshot shows the Oracle SQL Script Editor interface. At the top, the 'Script Name' field contains 'Create Demo Projects Tables'. The 'Download' button is highlighted with a green border. Below the script name, there are buttons for 'Cancel', 'Download', 'Delete', 'Save', 'Apply Changes', and 'Run'. The main area of the editor contains SQL code for creating two tables: 'DEMO_PROJ_STATUS' and 'DEMO_PROJ_TEAM_MEMBERS'. The code is as follows:

```

1 CREATE TABLE "DEMO_PROJ_STATUS"
2 ( "CD" VARCHAR2(15) NOT NULL ENABLE,
3  "DESCRIPTION" VARCHAR2(255) NOT NULL ENABLE,
4  "DISPLAY_ORDER" NUMBER NOT NULL ENABLE,
5  "CREATED" TIMESTAMP (6) WITH LOCAL TIME ZONE NOT NULL ENABLE,
6  "CREATED_BY" VARCHAR2(255) NOT NULL ENABLE,
7  "UPDATED" TIMESTAMP (6) WITH LOCAL TIME ZONE NOT NULL ENABLE,
8  "UPDATED_BY" VARCHAR2(255) NOT NULL ENABLE,
9  CONSTRAINT "DEMO_PROJ_STATUS_PK" PRIMARY KEY ("CD")
10 USING INDEX ENABLE
11 )
12 /
13 CREATE TABLE "DEMO_PROJ_TEAM_MEMBERS"
14 ( "ID" NUMBER NOT NULL ENABLE,
15  "USERNAME" VARCHAR2(255) NOT NULL ENABLE,
16  "FULL_NAME" VARCHAR2(255) NOT NULL ENABLE,
17  "EMAIL" VARCHAR2(255),
18  "PROFILE" VARCHAR2(4000),
19  "PHOTO_FILENAME" VARCHAR2(100),
20  "PHOTO_BLOB" BLOB,
21  "PHOTO_MIMETYPE" VARCHAR2(255),
22  "PHOTO_CHARSET" VARCHAR2(128),
23  "PHOTO_LAST_UPDATED" DATE,
24  "CREATED" TIMESTAMP (6) WITH LOCAL TIME ZONE NOT NULL ENABLE,
25  "CREATED_BY" VARCHAR2(255) NOT NULL ENABLE,
26  "UPDATED" TIMESTAMP (6) WITH LOCAL TIME ZONE NOT NULL ENABLE,
27  "UPDATED_BY" VARCHAR2(255) NOT NULL ENABLE,
28  CONSTRAINT "DEMO_PROJ_TEAM_MEMBERS_PK" PRIMARY KEY ("ID")
29 USING INDEX ENABLE,
30 CONSTRAINT "DEMO_PROJ_TEAM_MEMBERS_UK" UNIQUE ("USERNAME")
31 USING INDEX ENABLE
32 )
33 /

```

i) Click **Save**.

j) Now, create a script to include trigger definitions and then download the script. Repeat steps a through i and input the following:

- Select Trigger for Object Type
- Enter **Create Demo Projects Triggers** for script name.
- Select the following triggers:
 - ✓ **biu_demo_proj_status**
 - ✓ **biu_demo_proj_team_members**
 - ✓ **biu_demo_projects**
 - ✓ **biu_demo_proj_milestones**
 - ✓ **biu_demo_proj_tasks**
 - ✓ **biu_demo_proj_task_todos**
 - ✓ **biu_demo_proj_task_links**
 - ✓ **biu_demo_proj_comments**

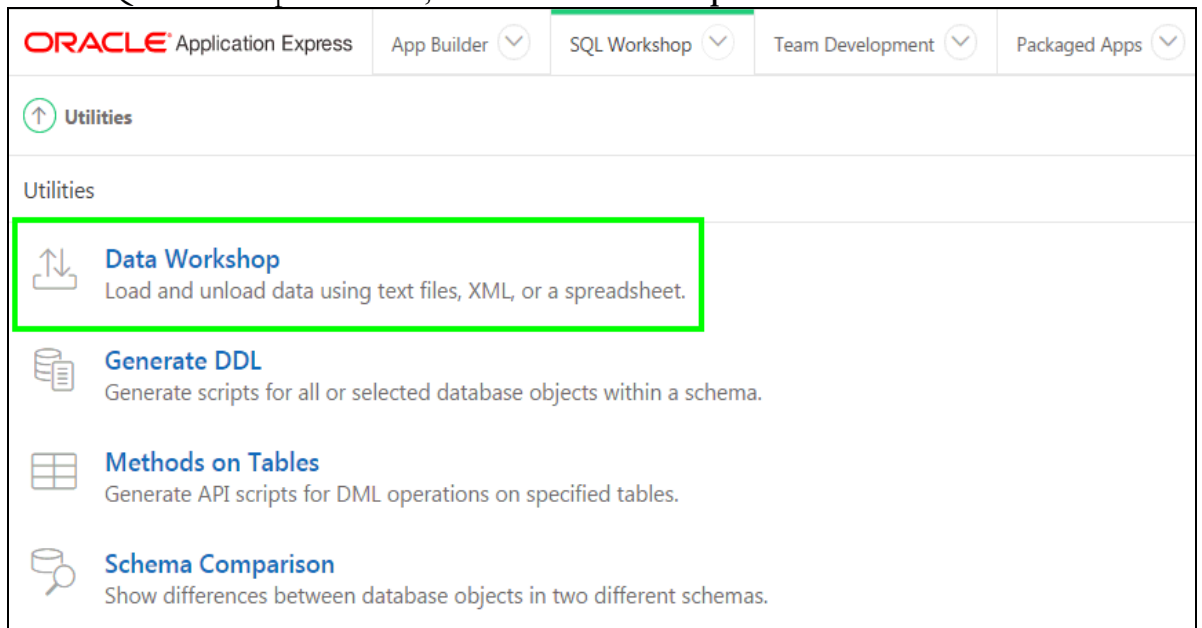
Script Name ? Cancel Download Delete Save Apply Changes Run

```

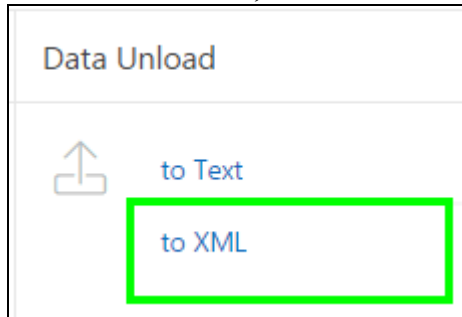
1 CREATE OR REPLACE EDITIONABLE TRIGGER "BIU_DEMO_PROJECTS"
2   before insert or update on demo_projects
3   for each row
4 begin
5   if :new.id is null then
6     :new.id := to_number(sys_guid(), 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX');
7   end if;
8
9   if inserting then
10    :new.created := localtimestamp;
11    :new.created_by := nvl(wv_flow.g_user,user);
12  end if;
13  :new.updated := localtimestamp;
14  :new.updated_by := nvl(wv_flow.g_user,user);
15 end;
16
17 /
18 ALTER TRIGGER "BIU_DEMO_PROJECTS" ENABLE
    
```

k) Now that you have both the DDL scripts created and downloaded, you need to unload the table data.

Under SQL Workshop > Utilities, select **Data Workshop**.



l) Under Data Unload, click to **XML**.



- m) On the Unload to XML – Columns page, select the application table for **Table**. See step e above for the Demo Projects table list.
Use the Ctrl key to select all of the columns in a table and click **Unload Data**.

The screenshot shows a dialog box titled "Unload to XML - Columns". It contains the following fields and options:

- Table Owner:** APEX_DEMO
- Table:** DEMO_PROJECTS
- Columns:** A list of columns is displayed, with the following items selected (highlighted in blue): ID, PROJECT_LEAD, NAME, DESCRIPTION, STATUS_CD, COMPLETED_DATE, CREATED, CREATED_BY, UPDATED, and UPDATED_BY.

At the bottom of the dialog, there are two buttons: "Cancel" and "Unload Data" (which is highlighted in green).

- n) Enter a meaningful name for the file and click **Save**.
- o) Repeat steps m and n for each of the Demo Projects application tables.
3. You have exported the application definition, database objects and data from your current development environment. Now first navigate to the target APEX environment and install the application definition.
Log into your target Application Express development environment and perform the following steps:
- Navigate to **App Builder** and click **Import**
 - Click **Choose File**.
Navigate to your working directory and double-click the application export file. Then, click **Next**.
 - On the File Import Confirmation page, click **Next**.
 - For Install As Application, specify your choice for Application ID. The default is Auto Assign New Application ID.
 - Click **Install Application**. The application is successfully installed now.
4. Log into your target Application Express development environment.

5. Use SQL Workshop to load and run the script file, for creating the table and trigger definitions
 - a) Click SQL Workshop.
 - b) Click SQL Scripts.
6. Upload the script to create the tables first.
 - a) Click **Upload**.
 - b) For File, click **Choose File**.
 - c) In the operating system File Browser, navigate to the subdirectory where you saved the table script file. Locate **Create Demo Project Tables.sql**, and double-click the file.
 - d) Click **Upload**.
7. Click the **Run** icon to the right of the script you uploaded.
Click Run Now.
8. Click the View Results icon for the script you just ran.
9. To create triggers, repeat steps 5 through 8 and select the **Create Demo Projects Triggers.sql**.
10. Currently the tables you created do not have any data. Use the XML files you created to populate the tables.

Note: The order in which the tables are populated is crucial, to ensure referential integrity does not prevent records loading.

For example, loading any records into DEMO_PROJECTS before loading the records into DEMO_PROJ_TEAM_MEMBERS will fail, as the ASSIGNEE column in DEMO_PROJECTS must correspond to an existing record in DEMO_PROJ_TEAM_MEMBERS.

In the Application Express main toolbar, click the SQL Workshop Down Arrow, select Utilities and then select **Data Workshop**.

11. Under Data Load, locate **XML Data**. Click **XML Data**.
12. For Table, select **DEMO_PROJ_TEAM_MEMBERS**.
For File, click **Choose File**, locate the file for DEMO_TEAM_MEMBERS, and double-click the file.
13. Click **Load Data**.
14. Load data for all of the Demo Projects tables. Repeat steps 10 through 13 for each of the table.
15. Now, you can run and review the application.



Oracle is committed to developing practices and products that help protect the environment

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. Licensed under the Creative Commons Attribution 4.0 International License as shown at <https://creativecommons.org/licenses/by/4.0/legalcode>.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document.

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

Hardware and Software, Engineered to Work Together