



An Oracle White Paper

April 2019

Extending Oracle E-Business Suite Release 12 using Oracle APEX

Revision 3.03

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Executive Overview	2
Introduction	2
What's New	2
Prerequisites	3
Concept Overview.....	4
Oracle APEX Architecture	5
Recommended Mid-Tier Deployment Architecture	6
Recommended Database Architecture	8
Upgrading to Oracle E-Business Suite Release 12.2	10
Installing Oracle APEX	12
Oracle APEX Schemas and Workspaces	13
Accessing Oracle E-Business Suite Data	19
Updating Oracle E-Business Suite Tables	21
Downloading and Installing the Sample Code	23
Integrating with Oracle E-Business Suite	24
Creating an Oracle APEX Application	24
Oracle APEX Authentication and Authorization	34
Configuring the Oracle APEX Login Process.....	35
Defining Oracle APEX Authorizations	39
Installing the Sample Packaged Application	44
Defining Oracle E-Business Suite Profile and Form Functions	45
Defining Oracle E-Business Suite Menus and Responsibilities.....	48
Downloading and Installing the Environment Script.....	52
Running the Application Express Extension	53
Custom Tables on Oracle E-Business Suite R12.2	55
Upgrading your Application Express Extensions	56
Upgrading your APEX Extensions in a Different Database	58
Conclusion	59
Acknowledgements.....	59

Executive Overview

This paper outlines how to extend Oracle E-Business Suite functionality utilizing Oracle APEX. This paper is a collaboration between both product teams to ensure that any extensions developed following the strategy outlined in this paper are fully supported. Recommended architecture and security considerations are discussed in detail.

Introduction

Oracle E-Business Suite delivers a wide range of functionality to handle core areas of your business processing needs. However, there are situations where you want to extend your information systems beyond the range of Oracle E-Business Suite. Many times these necessary extensions are meant to handle unique industry conventions, specific customer requirements, or perhaps to offer some other competitive edge. Sometimes these change requests are simple enough, but other times more extensive customizations are needed. In these scenarios, Oracle APEX, also known as Oracle Application Express, provides an easy way to create supplemental applications that are easily integrated with your Oracle E-Business Suite and its data.

Oracle APEX is a rapid web application development tool for the Oracle database. Oracle APEX combines the qualities of a personal database (productivity, ease of use, and flexibility) with the qualities of an enterprise database (security, integrity, performance, scalability, availability, and built for the web). The browser-based interface, declarative programming framework, and simple wizards make Oracle APEX easy to learn and enable you to quickly build robust applications.

Oracle APEX offers you a quick and highly productive way to extend your Oracle E-Business Suite environment with almost no impact to your existing implementation. By moving custom behaviors out of Oracle E-Business Suite and into Oracle APEX, you can make Oracle E-Business Suite patching and upgrading much simpler.

You should be able to easily combine your Oracle APEX applications with your Oracle E-Business Suite installation using the architecture and techniques described in this white paper.

What's New

Revision 3 of this white paper includes updates to the supported product versions, new screenshots, and clearer images. This version now includes a new script, **apex_ebs_env.sql**, which accepts numerous parameters and then defines all of the required E-Business Suite components, such as a profile option, form functions, menus, responsibility, and creating a user. The existing scripts were also consolidated and simplified. Based on feedback, the custom schema for Oracle APEX given in the examples has been renamed to `XX_APEX`, instead of `APEX_EBS_EXTENSION`.

Prerequisites

The current prerequisites for the solution given in this paper are:

- Oracle E-Business Suite 12.1.3 or above
 - This paper includes specific instructions that apply to both Oracle E-Business Suite Release 12.1 and Release 12.2
 - For Oracle E-Business Suite Release 12.1, patch 12316083:R12.FND.B is required. {Go to My Oracle Support Patches & Updates: https://updates.oracle.com/Orion/PatchSearch/process_form?bug=12316083}
- Full-use Oracle Database¹, Release 11.2.0.4, or above
- [Oracle APEX](#) 5.0 or above
 - Oracle recommends installing the latest available version of Oracle APEX.
 - Screenshots in this paper utilize APEX 18.2, which may vary for other releases.
- [Oracle REST Data Services \(ORDS\)](#) 3.0 or above
 - Oracle recommends installing the latest available version of ORDS.
- [Oracle WebLogic Server \(WLS\)](#)² 12c or above
 - You can also utilize Apache Tomcat, or ORDS Standalone (Jetty) Server, instead of Oracle WebLogic Server.

Note: The minimum version of products listed above are those supported at time of writing. Please review the relevant Lifetime Support Policy documents for each product to ensure you install supported software, and know when support ends.

¹ The limited-use Oracle Database license agreement included with an Oracle E-Business Suite license does not allow for the creation of an additional schema as outlined in this paper below. Please refer to the following license agreement: <http://www.oracle.com/us/corporate/pricing/application-licensing-table-070571.pdf>. Therefore, to be fully compliant, full-use Oracle Database Enterprise Edition and Oracle Internet Application Server Enterprise Edition licenses are required.

² A separate Oracle WebLogic Server license is required, other than the Oracle WebLogic Suite for Oracle Applications, utilized for Oracle E-Business Suite Release 12.2. E-Business Suite utilizes WLS 11.2 which does not support installation of JDK 8 which is required by ORDS. Updating WLS for use by EBS is not supported.

Concept Overview

Oracle APEX offers you the ability to create reports, charts, and calendars as well as pages for data review and manipulation. To allow you to create Oracle APEX applications that reference Oracle E-Business Suite data, the desired data is exposed through database views in the APPS schema. 'SELECT' access to those views is then granted to an existing or new Oracle E-Business Suite custom schema (for this document, XX_APEX). Because direct 'INSERT', 'UPDATE', and 'DELETE' operations on Oracle E-Business Suite data is unsupported, applications that manipulate Oracle E-Business Suite data will use the public, documented and supported, Oracle E-Business Suite APIs. References to these APIs can either be coded into the Oracle APEX applications or you can define 'INSTEAD OF' triggers against the new database views which call the appropriate APIs.

The extensions built in Oracle APEX will appear seamless to your end users because you can use the same authentication (who can login) and authorization (who can see what) within your Oracle APEX applications that are used within your Oracle E-Business Suite installation. Your new applications can either be stand-alone or fully integrated. Stand-alone applications are accessed directly, but use the same login credentials used for Oracle E-Business Suite access. Fully integrated applications are registered within the Oracle E-Business Suite so they are available from within the Oracle E-Business Suite menus. Oracle APEX contains a pre-configured Authentication Scheme for Oracle Single Sign-on (OSSO) and also HTTP Header Variable for use with Oracle Access Manager (OAM).

If your Oracle E-Business Suite instance uses OSSO or OAM for authentication, setting up authentication will be very simple. If you are using custom authentication for Oracle E-Business Suite, you can still achieve seamless integration but you will need to code a function within the APPS schema that validates the user, and then add an authentication scheme to your Oracle APEX applications that calls that new function.

This document discusses the Oracle APEX architecture and provides detailed instructions for creating all the components necessary to extend Oracle E-Business Suite. In support of the instructions, scripts and sample packaged applications are available for download to ensure that any developer can quickly replicate the examples.

Oracle APEX Architecture

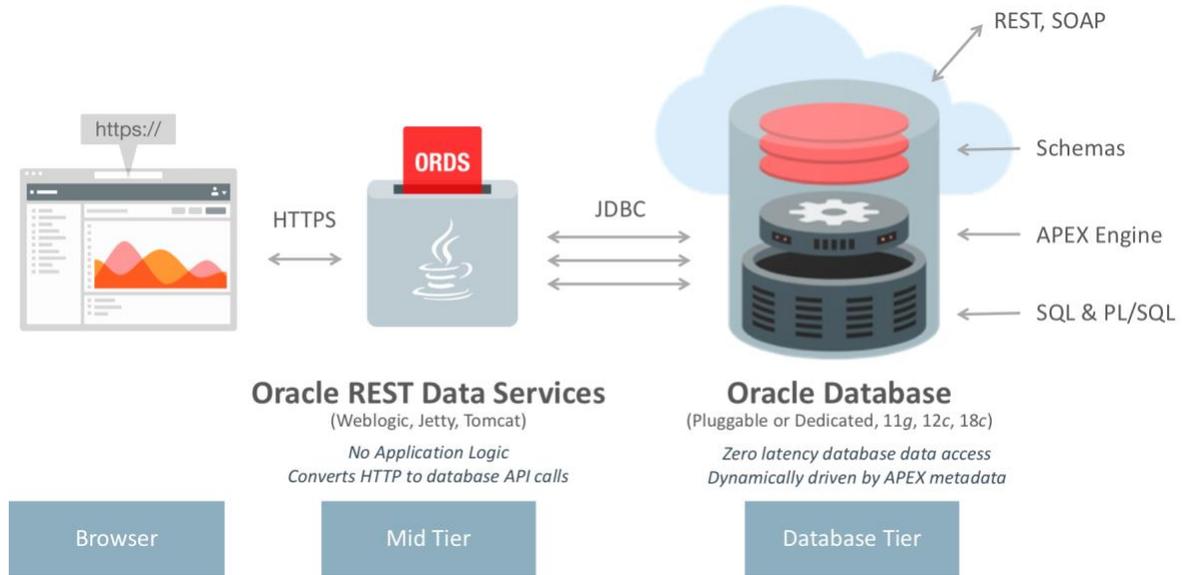


Figure 1. Oracle APEX Architecture

Oracle APEX resides completely within the Oracle Database in its own schema and can be installed on any version of the Oracle Database from 11gR2 and above. Runtime, development and deployment require no client software as access is 100% browser-based via a Web listener communicating with the Oracle Database. The application definitions are stored as meta-data within the Oracle APEX schema which is accessed to perform page rendering and processing.

The middle tier requires Oracle REST Data Services (ORDS). ORDS is a JAVA EE based solution that is certified with Oracle WebLogic Server, ORDS Standalone (Jetty Server), and Apache Tomcat. Oracle does not recommend utilizing the Embedded PL/SQL Gateway (EPG) for integrating Oracle E-Business Suite and Oracle APEX.

Note: You can not utilize the Oracle WebLogic Server 11.2 installed by Oracle E-Business Suite 12.2, as this release does not support JDK 8 required by ORDS. It is not supported to update the release of Oracle WebLogic Server utilized by Oracle E-Business Suite.

Recommended Mid-Tier Deployment Architecture

The recommended configuration is to have two separate application servers, by installing the Oracle REST Data Services within Oracle WebLogic Server, on a separate application server to the one configured for Oracle E-Business Suite. {See Figure 2 and Figure 3}. Oracle recommends installing a separate application server so that each application server can be managed independently.

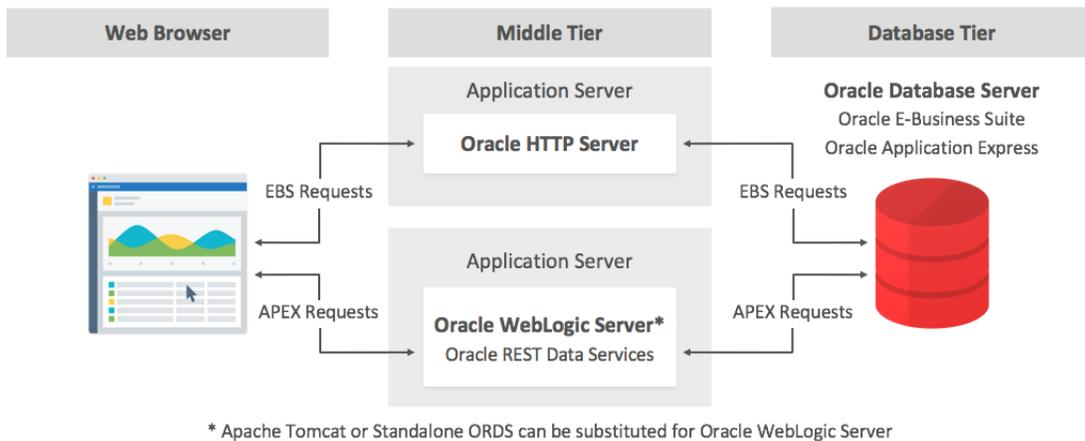


Figure 2. Recommended Middle Tier Configuration - Release 12.1

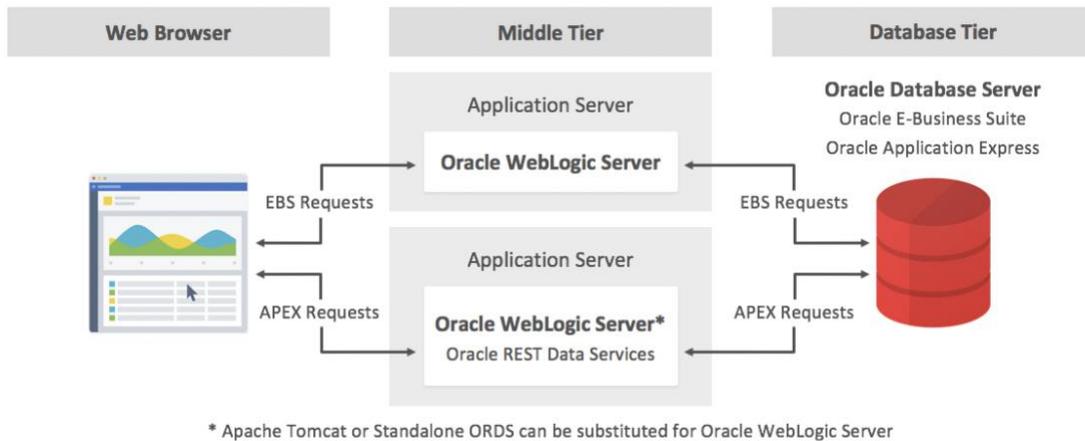
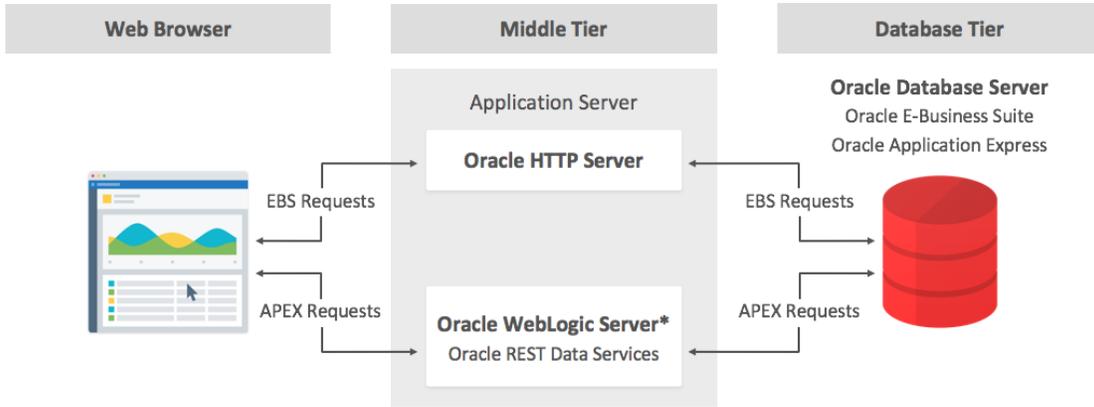


Figure 3. Recommended Middle Tier Configuration - Release 12.2

Note: You can also install Oracle REST Data Services into ORDS Standalone (Jetty Server), or Apache Tomcat, instead of Oracle WebLogic Server, and still be fully supported and in compliance.

Alternate Mid-Tier Deployment Architecture for E-Business Suite Release 12.1

For E-Business Suite Release 12.1 you can also install the Oracle WebLogic Server on the same Application Server utilized by Oracle E-Business Suite {See Figure 4}. However, it is not certified to install Oracle REST Data Services into the Oracle WebLogic Server(s) utilized by Oracle E-Business Suite Release 12.2.



* Apache Tomcat or Standalone ORDS can be substituted for Oracle WebLogic Server

Figure 4. Alternate Middle Tier Configuration - Release 12.1 ONLY

Recommended Database Architecture

Oracle recommends configuring Oracle APEX within the Oracle Database server where Oracle E-Business Suite is installed {See Figure 5}. A separate, minimally privileged, Oracle E-Business Suite custom schema (XX_APEX in this document) must be defined as the basis for building Oracle APEX applications. Defining an Oracle APEX workspace directly against the owning Oracle E-Business Suite schema (APPS) is **not** supported, as this is a breach of security protocol.

This recommendation is based on the fact that the Oracle E-Business Suite APPS schema has full access to all of the Oracle E-Business Suite database objects. If you were to associate the APPS schema with an Oracle APEX workspace, then the Oracle APEX applications in that workspace would also have full access to all of the underlying Oracle E-Business Suite database objects. These applications could then perform insert, update, and delete statements directly on the APPS database objects and potentially invalidate the APPS schema, as critical business rules, enforced by the public APIs, are bypassed. Therefore, not only is this a breach of security protocol, it is potentially catastrophic to your Oracle E-Business Suite environment, and is not supported.

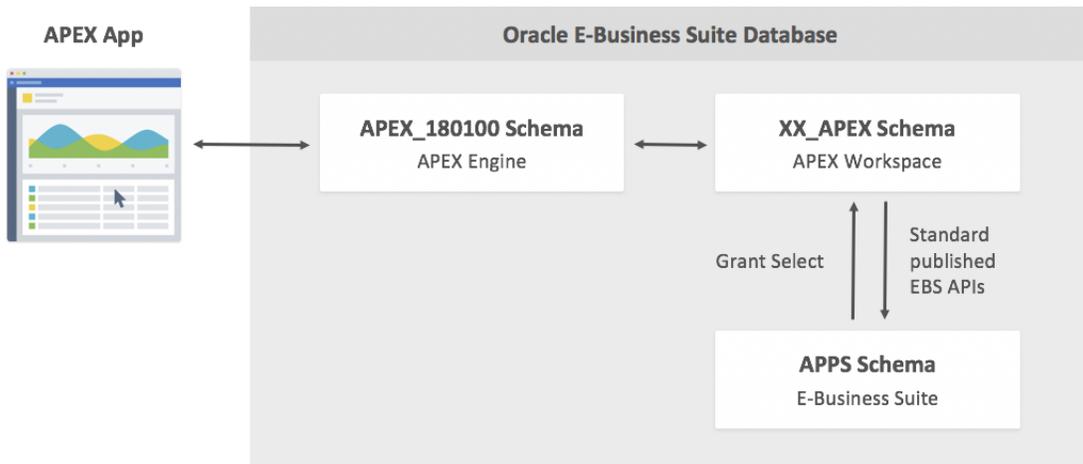


Figure 5. Recommended Database Configuration

Alternative Database Architecture using Database Links

You are permitted to configure Oracle APEX on a separate Oracle Database server from the Oracle Database server where Oracle E-Business Suite is installed {See Figure 6}. A private database link from any external Oracle Database server to the Oracle E-Business Suite Database server is recommended. This database link should be made to a separate, minimally-privileged schema (XX_APEX_LOCAL in this document), from the owning Oracle E-Business Suite schema (APPS). It is imperative that you do not define database links directly to the owning Oracle E-Business Suite schema (APPS), as this is a breach of security protocol.

If utilizing Oracle APEX 18.1 or above then you should utilize Representational State Transfer (REST), and in particular the Oracle APEX feature - REST Enabled SQL, to read data from the Oracle E-Business Suite Database.

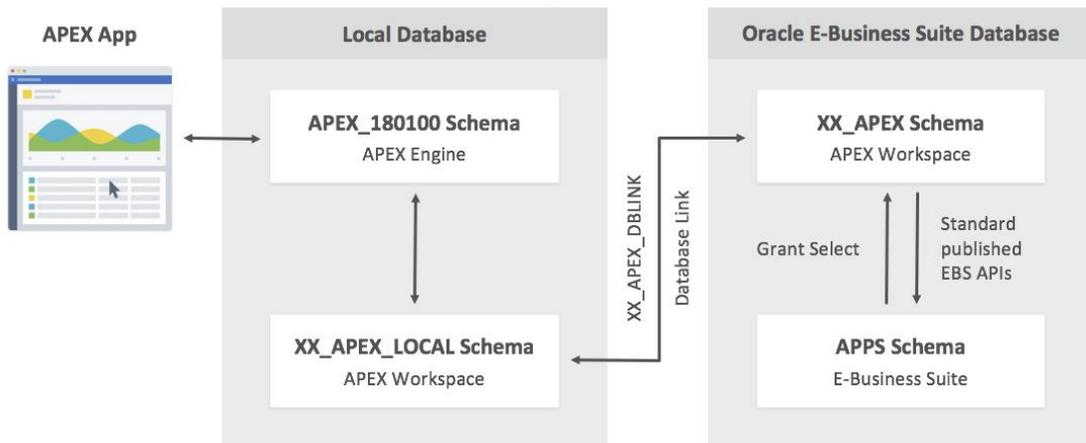


Figure 6. Alternate Database Configuration

For example, if you wish to enhance existing Oracle APEX applications, that are running on another Oracle Database, to incorporate Oracle E-Business Suite information, then you can utilize REST Enabled SQL or database links to obtain the required Oracle E-Business Suite data.

Upgrading to Oracle E-Business Suite Release 12.2

If you have previously integrated Oracle APEX with Oracle E-Business Suite Release 12.1 or earlier and are upgrading to Oracle E-Business Suite Release 12.2, then you need to plan your upgrade to take into consideration the integration of these products. Specifically, Oracle recommends the following tasks:

- Upgrade to E-Business Suite Release 12.2 following the My Oracle Support Knowledge Document 1531121.1
 - Register the custom schema(s), assigned to Application Express workspaces, within E-Business Suite Release 12.2. For further details, review Chapter 4 “Oracle E-Business Suite Online Patching” in the [Oracle E-Business Suite R12.2 Upgrade Guide](#)
- Upgrade to the latest version of Oracle APEX (currently 18.2) available on the Oracle Technology Network (<http://apex.oracle.com/download>).
- Upgrade to the latest version of Oracle REST Data Services (currently 18.4)

Before running the Oracle E-Business Suite Online Enablement process, it is critical that the Oracle E-Business Suite custom schema where Application Express is associated (XX_APEX in this document) conforms to Online Patching Standards as defined in My Oracle Support Knowledge Document 1531121.1. Otherwise, your Application Express extensions will stop working. With Oracle E-Business Suite Release 12.2 the APPS schema will be Edition-Based Redefinition (EBR) enabled to facilitate E-Business Suite Online Patching. Attempting to access editioned objects from a non-editioned schema will cause Oracle database errors. Oracle E-Business Suite Online Patching will now correctly edition the Application Express workspace custom schema(s) so that your Application Express extensions continue to function without interruption during the online upgrade process.

Note: If you attempt to enable edition-based redefinition on the Application Express Engine schema (For Application Express Release 18.2 that schema is APEX_180200) you will invalidate Application Express, making it inoperable.

Oracle recommends upgrading from any previous releases of Oracle APEX and Oracle REST Data Services to the latest available releases of each. This can be performed by following the appropriate installation guides for each product. These updates can be performed as part of the 12.2 Post Upgrade steps, before the system is released to end users. Future updates to these components could also be scheduled as part of an E-Business Suite Online Patching Cycle, after performing a cutover phase that does not bring up the middle tier processes (adop mtrestart=no). For further details on Online Patching and the adop utility, review the [Oracle E-Business Suite R12.2 Maintenance Guide](#), “The adop utility” section.

If you were previously running Oracle REST Data Services on Oracle Glassfish, then Oracle recommends that you install Oracle REST Data Services into Oracle WebLogic Server or Apache Tomcat, as support for running Oracle REST Data Services on Oracle Glassfish is deprecated and will be completely removed in an upcoming release.

Now that you have upgraded your environment you should review the Custom Tables on Oracle E-Business Suite R12.2

Since your Oracle E-Business Suite custom schema (XX_APEX in this document), which you used to create the Oracle APEX application, is Edition-Based to comply with Oracle E-Business Suite Online Patching, there are a few differences related to your custom tables and how Oracle APEX handles DML operations.

Once your custom table is created, you need to generate the editioned view which will also create the grant and the synonym for APPS. For example, if you create a table in XX_APEX called XX_CUST_TABLE then execute the following EBS API, as the Apps user:

1. Connect to your Oracle E-Business Suite Database server
2. Start SQL*Plus and connect as APPS
3. Execute the following command:

```
exec AD_ZD_TABLE.UPGRADE('XX_APEX','XX_CUST_TABLE');
```

The custom editioned view created will have the same name as the custom table with the hast (#) sign at the end of the name. In order for the API to execute correctly it is essential that the custom schema name and the table name are in uppercase.

If creating triggers then you must create them on the editioned view and not the custom base table.

In the Oracle APEX applications, any access to custom tables should instead be referencing the corresponding editioned view. When utilizing the Create App Wizard or Create Page Wizard within Oracle APEX select one of the editioned views and complete the wizard as usual.

For existing applications, review any DML process having a custom table name and replace it with the editioned view of that table:

For example `XX_CUST_TABLE` becomes `XX_CUST_TABLE#`

On the Oracle E-Business Suite side it is business as usual! The `AD_ZD_TABLE.UPGRADE` API created a grant for APPS and a synonym on the name of the custom base table pointing to the editioned view. So using this object in views, PL/SQL packages, procedures or functions is totally transparent to the developer.

Upgrading your Application Express Extensions section below, to understand how to deploy enhancements to your Application Express applications, especially when they involve modification to existing database objects.

Installing Oracle APEX

For a fresh install, when you have not previously built Application Express extensions within your Oracle E-Business Suite environment, you need to configure Oracle APEX within the Oracle Database. Since Oracle Database 11gR1, Oracle APEX is part of the standard seed database installation. As such the Application Express engine is probably already installed but not configured within your environment. However, Oracle very highly recommends downloading the latest version of Oracle APEX (currently APEX 18.2) from the Oracle Technology Network (<http://apex.oracle.com/download>) and following the [Installation Guide – 4 Installing Application Express and Configuring Oracle REST Data Services](#).

You will also need to download the latest version of Oracle REST Data Services (currently 18.4) from Oracle Technology Network (<http://www.oracle.com/technetwork/developer-tools/rest-data-services/overview/index.html>) and follow the installation instructions for the Oracle WebLogic Server. To maintain the same context path for Application Express, `/apex`, you must rename the `ords.war` file to `apex.war` before deployment within Oracle WebLogic Server.

After completing the installation steps for both Oracle APEX (including enabling Network Services in Oracle Database 11g or above as appropriate) and Oracle REST Data Services, you need to create a workspace for your development and then create applications within that workspace. For your production instance it is highly recommended to run Oracle APEX in runtime-only mode to further harden security [See: [Installation Guide - 4.15 Converting Between Runtime and Full Development Environments](#)].

Oracle APEX Schemas and Workspaces

Workspaces are logical containers within Oracle APEX that provide functional security for Oracle APEX development. Workspaces can be associated with one or more custom database schemas. Each custom schema associated with an Oracle APEX workspace can be used to parse SQL and PL/SQL requests.

Oracle E-Business Suite uses a schema called APPS, which has full access to the complete Oracle E-Business Suite data model. As stated earlier, associating an Oracle APEX workspace directly with the APPS schema is **not** supported. Instead, create at least one separate schema in the Oracle E-Business Suite database for developing your Oracle APEX applications. In this document a custom schema XX_APEX is created.

As per Oracle E-Business Suite development standards, any “custom object created in the database should be prefixed with XX, YY, or ZZ. XX is the most popular prefix used, and is the prefix used in this paper, hence, the custom schema is called XX_APEX.

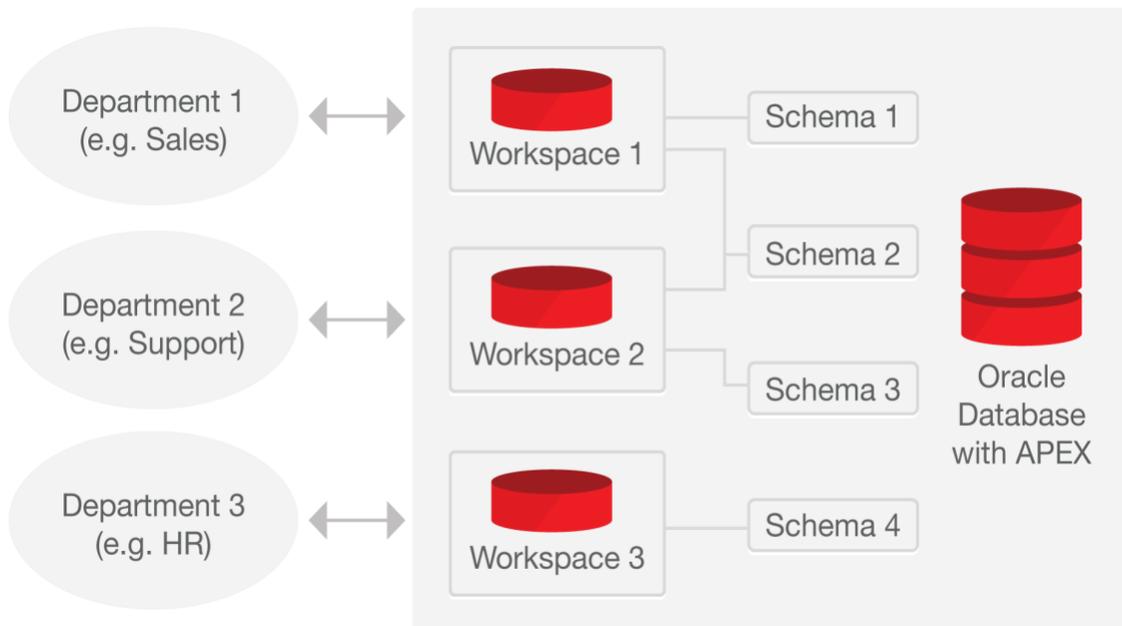


Figure 7. Example of Oracle APEX Workspaces

Defining the Oracle APEX Workspace and Schema

Based on the recommended database architecture, you should create a workspace called XX_APEX and a custom schema called XX_APEX on the database server where Oracle E-Business Suite is installed. If you already have a custom schema defined you can also associate that schema with an Oracle APEX workspace. However, if you plan to develop Application Express on a different server then skip this section and follow the steps outlined in Configuring Oracle APEX in a Different Database.

Create a workspace called XX_APEX and a custom schema called XX_APEX by performing the following steps:

1. Log into Oracle APEX Administration
(http://<EBS_Hostname>:8080/apex/apex_admin)
2. Enter *Username* **ADMIN**, *Password* (defined during installation by executing @apxchpwd), click Login
3. Navigate to Manage Workspaces > Create Workspace
4. Enter *Workspace Name* **XX_APEX**, click *Next >*
5. Enter *Schema Name* **XX_APEX**, and *Password* <<**xx_apex_password**>>, click *Next >*
6. Enter *Administrator Username* **ADMIN**, *Administrator Password* <<**APEX_Password**>>, and your email address for *Email*, click *Next >*
7. Click Create Workspace, click *Done*

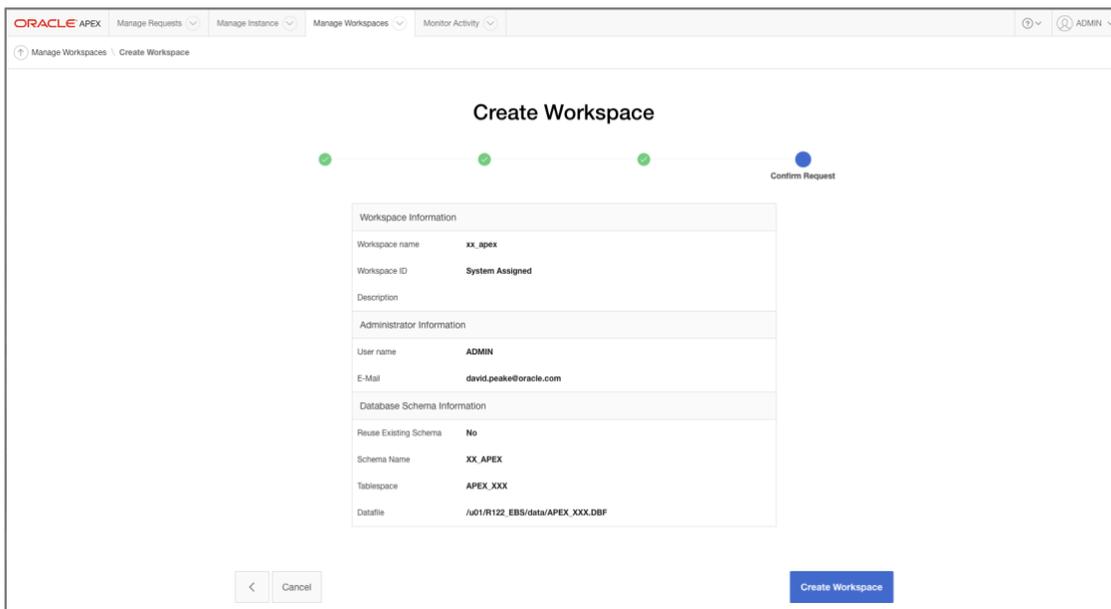


Figure 8. Oracle APEX Workspace Provisioning

The above steps will create a database schema, XX_APEX, with the correct permissions for building local database objects. However, if you are running Oracle E-Business Suite Release 12.2 or above, then you must update this schema to enable Edition-Based Redefinition (EBR). This is required as the APPS schema is enabled in order to achieve E-Business Suite Online Patching, and you will receive Oracle database errors if you do not also enable your custom schema.

Enable Edition-Based Redefinition by performing the following steps:

1. Connect to your Oracle E-Business Suite Database server
2. Start SQL*Plus and connect as SYS specifying the SYSDBA role
3. Alter the APEX schema:

```
alter user XX_APEX enable editions;
```

{This functionality is incorporated into the apex_ebs_setup.sql script referenced later in this paper}

If you are running Oracle E-Business Suite Release 12.2 or above then you will also need to register this schema within E-Business Suite. Information on how to register the schema is available in the Oracle E-Business R12.2 Suite Maintenance Guide, Chapter 6 “Basic DBA Tasks”, section “Registering an ORACLE Schema”. The documentation references MOS Knowledge Base - **Creating a Custom Application in Oracle E-Business Suite Release 12.2 (Doc ID 1577707.1)**

Configuring Oracle APEX in a Different Database

If you have installed Oracle APEX into a different Oracle Database to the database where E-Business Suite is installed, then Oracle recommends ensuring the database link is closed correctly, by updating the Oracle REST Data Services configuration file utilized by Application Express. To update the configuration file perform these steps:

1. Connect to the server where Oracle Rest Data Services is configured
2. Navigate to the directory that holds the appropriate defaults.xml file
{Such as /ords/config/apex}
3. Edit the defaults.xml file and add the following lines:

```
<entry key="procedure.preProcess">apex_util.close_open_db_links
</entry>
<entry key="procedure.postProcess">apex_util.close_open_db_links
</entry>
```

4. Restart Oracle Rest Data Services

You will also need to manually create the minimally privileged user, XX_APEX, within the E-Business Suite Oracle database. This custom schema will be used to access APPS objects, as it is very strongly discouraged to define a database link from another database directly to the APPS schema. It is also very strongly discouraged to define a database link from APPS to *any* other database. To create this user perform these steps:

1. Connect to your Oracle E-Business Suite Database server
2. Start SQL*Plus and connect as SYS specifying the SYSDBA role
3. Create the XX_APEX schema:

```
create user XX_APEX
identified by <<xx_apex>>
default tablespace SYSAUX;
```

4. Grant the required rights to create database objects:

```
grant create view to XX_APEX;
```

If you are running Oracle E-Business Suite Release 12.2 or above, then you must update this schema to enable Edition-Based Redefinition (EBR). This is required as the APPS schema is enabled in order to achieve E-Business Suite Online Patching. Enable Edition-Based Redefinition by performing the following steps:

1. Connect to your Oracle E-Business Suite Database server
2. Start SQL*Plus and connect as SYS specifying the SYSDBA role
3. Alter the APEX schema:

```
alter user XX_APEX enable editions;
```

You should also create a workspace called XX_APEX_LOCAL and a schema called XX_APEX_LOCAL on the local database server where Oracle APEX is configured. To create the workspace within the local installation of Application Express follow these steps:

1. Log into Oracle APEX Administration
(http://<EBS_Hostname>:8080/apex/apex_admin)
2. Enter *Username* **ADMIN**, *Password* (defined during installation by executing @apxchpwd), click *Login*
3. Navigate to *Manage Workspaces > Create Workspace*
4. Enter *Workspace Name* **XX_APEX_LOCAL**, click *Next >*
5. Enter *Schema Name* **XX_APEX_LOCAL**, and *Password* <<xx_apex_local>>, click *Next >*
6. Enter *Administrator Username* **ADMIN**, *Administrator Password* <<APEX_Password>>, and your email address for *Email*, click *Next >*

7. Click Create Workspace, click *Done*

Now that you have the two schemas, XX_APEX within the E-Business Suite Oracle Database server and XX_APEX_LOCAL in the Oracle Database where Application Express is configured, then you need to define the database link between these schemas by following these steps:

1. Connect to your local Oracle Database server (where Application Express is configured)
2. Start SQL*Plus and connect as SYS specifying the SYSDBA role
3. Change the password for the XX_APEX_LOCAL schema:

```
alter user XX_APEX_LOCAL identified by <<xx_apex_local>>;
```

4. Connect to the XX_APEX_LOCAL schema:

```
connect XX_APEX_LOCAL / <<xx_apex_local>>
```

5. Add the database link:

```
create database link XX_APEX_DBLINK
connect to XX_APEX
identified by <<xx_apex>>
using ' (DESCRIPTION= (ADDRESS= (PROTOCOL=TCP)
(HOST=<<Remote_Host>>) (PORT=1521) ) (CONNECT_DATA=
(SERVER=DEDICATED) (SERVICE_NAME=<<Remote_Service_Name>>)) ) '
```

{*Remote_Host* and *Remote_Service_Name* refer to the Oracle Database where the XX_APEX schema is defined}

{This functionality is incorporated into the apexebs_apex_dblink_setup.sql script referenced later in this paper}

Configuring Developers within Application Express

Now that the workspace has been created, development will be performed within the Application Builder. To log into the Application Builder and create developer accounts, follow these steps:

1. Log into Oracle APEX Application Builder (http://<EBS_Hostname>:8080/apex/apex)
2. Enter *Workspace* **XX_APEX / XX_APEX_EBS_LOCAL**, *Administrator Username* **ADMIN** and *Administrator Password* <<**APEX_Password**>> entered when creating the workspace, click *Login*
3. You will need to reset the password – enter *Current Password* <<**APEX_Password**>>, *New Password* <<**Your New Password**>>, *Confirm New Password* <<**Your New Password**>>, click *Apply Changes*, click *Close Window*;
4. Navigate to Administration > Manage Users and Groups
5. Depending on your requirements you can create one or more developers using the Create User or Create Multiple Users buttons (top right).



Figure 9 - Accessing Administration, Manage Users and Groups

Accessing Oracle E-Business Suite Data

In Oracle E-Business Suite R12.1 and above, multi-org views have been removed, and replaced with synonyms. The data restriction is accomplished by assigning a virtual private database (VPD) policy also known as Row Level Security (RLS) to the synonym. This policy allows the system to dynamically generate restricting conditions when queries are run against the synonym. For the clarity of this document, references to EBS tables means accessing the APPS synonym having the VPD policy for Multi-Org Access Control (MOAC) access.

Oracle APEX provides numerous wizards to rapidly build application components on existing Oracle tables or views. However, a prerequisite of many wizards within Oracle APEX is that the underlying tables or views contain primary key constraints. Given Oracle E-Business Suite tables and views do not typically include primary key constraints, it is advisable to define additional database views that incorporate primary key constraints for the Oracle Applications objects you wish to access. For security purposes, it is recommended that Oracle E-Business Suite data is accessed only through these views. The views can incorporate security, joins, etc., and prevent Oracle APEX applications from making any unauthorized changes to the underlying data.

Below is sample code for creating such a view in the APPS schema –

```
CREATE OR REPLACE VIEW xx_apex_ebs_user
( user_id
, user_name
, start_date
, end_date
, description
, email_address
, user_guid
, person_party_id
, CONSTRAINT APEX_EBS_USER_pk
PRIMARY KEY (user_id)
RELY DISABLE NOVALIDATE
)
AS
SELECT user_id
, user_name
, start_date
, end_date
, description
, email_address
, user_guid /* Used for Single-Sign On */
, person_party_id /* FK to party information */
FROM fnd_user;
```

{This functionality is incorporated into the apex_ebs_setup.sql script referenced later in this paper}

It is then necessary to grant rights to the new custom schema (XX_APEX) from the APPS schema -

```
GRANT SELECT ON xx_apex_ebs_user TO XX_APEX;  
GRANT SELECT on fnd_responsibility_vl TO XX_APEX;
```

The final step is to create a corresponding view within the XX_APEX schema -

```
CREATE OR REPLACE VIEW xx_apex_ebs_user AS  
  SELECT * from apps.xx_apex_ebs_user;
```

If you have installed Application Express into a different Oracle Database to the database where E-Business Suite is installed, then you will need to create a corresponding view within the XX_APEX_LOCAL schema that accesses the view in XX_APEX -

```
CREATE OR REPLACE VIEW xx_apex_dblink_ebs_user AS  
  SELECT * from xx_apex_ebs_user@XX_APEX_DBLINK;
```

Updating Oracle E-Business Suite Tables

Although it may be tempting to insert/update/delete records on Oracle E-Business Suite tables directly, this is unsupported and must be avoided. Please keep in mind that direct updates to Oracle E-Business Suite tables is not supported unless explicitly documented. Direct updates will bypass validation, security and business logic which could lead to data corruption or unexpected system behavior. Oracle recommends that all Oracle E-Business Suite updates are made through calls to public APIs. The Oracle E-Business Suite Integration Repository provides a searchable list of public APIs for the system.

Below is sample code for creating a package body within the APPS Schema:

```
CREATE OR REPLACE PACKAGE BODY xx_apex_sample_apis AS

FUNCTION apex_validate_login ( p_username in varchar2
                             , p_password in varchar2
                             ) RETURN BOOLEAN

IS
BEGIN
    RETURN fnd_user_pkg.validatelogin(p_username, p_password);
END apex_validate_login;

PROCEDURE apex_update_email ( p_username      in varchar2
                              , p_owner       in varchar2
                              , p_email_address in varchar2
                              )

IS
BEGIN
    wf_event.setdispatchmode('async');
    fnd_user_pkg.updateuser
    ( x_user_name      => p_username
      , x_owner        => p_owner
      , x_email_address => p_email_address);
END apex_update_email;

END xx_apex_sample_apis;
/
```

{This functionality is incorporated into the apex_cbs_setup.sql script referenced later in this paper}

When utilizing the Oracle APEX wizards on the new views to define applications, DML automatic row processes will be defined referencing the view specified. These processes can be deleted and alternate processes written which call the appropriate APIs. An alternative is to create 'INSTEAD OF' triggers on these new views which call the API to change the data in the tables. These triggers only need to be defined once against each view and will allow the standard Oracle APEX page processes that are generated as part of the page creation wizards to be utilized.

Below is sample code for creating such a trigger in the new schema. This code is not included in the sample code provided, as our examples don't update the `fnd_flex_values` records.

```
CREATE OR REPLACE TRIGGER apex_fnd_flex_values_tr
  INSTEAD OF INSERT OR UPDATE ON apex_fnd_flex_values
DECLARE
  v_storage_value VARCHAR2(32000);
BEGIN
  IF INSERTING THEN
    fnd_flex_val_api.create_independent_vset_value
      ( p_flex_value_set_name => :NEW.flex_value_set_name
        , p_flex_value         => :NEW.flex_value
        , p_description        => :NEW.description
        , x_storage_value      => v_storage_value
      );
  ELSIF UPDATING THEN
    fnd_flex_val_api.update_independent_vset_value
      ( p_flex_value_set_name => :NEW.flex_value_set_name
        , p_flex_value         => :NEW.flex_value
        , p_description        => :NEW.description
        , x_storage_value      => v_storage_value
      );
  ENDIF;
END;
/
```

Downloading and Installing the Sample Code

The code samples given above are available for download from the Oracle Technology Network (OTN). To run these scripts, perform the following steps:

1. Download the Ssetup script file from OTN –
<http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/329274.sql>
2. Save the file as **apex_ebs_setup.sql** and upload to your Oracle E-Business Suite Database server
3. Connect to your Oracle E-Business Suite Database server
4. Start SQL*Plus and connect as SYS specifying the SYSDBA role
5. Run the script, providing the appropriate criteria:
@apex_ebs_setup

If you have installed Application Express into a different Oracle Database to the database where E-Business Suite is installed, then perform the following steps:

1. Download the DB Link Setup script file from OTN –
<http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/2282591.sql>
2. Save the file as **apex_ebs_dblink_setup.sql** and upload to your Oracle E-Business Suite Database server
3. Connect to your local Oracle Database server (where Application Express is configured)
4. Start SQL*Plus and connect as the APEX Workspace schema [APEX_EBS_LOCAL]
5. Run the script, providing the appropriate criteria:
@apex_ebs_dblink_setup

Integrating with Oracle E-Business Suite

In order to connect from Oracle E-Business Suite to your new Oracle APEX Applications, a secure mechanism is required to hand control from Oracle E-Business Suite to Oracle APEX. The following actions need to be performed:

1. Create an Oracle APEX application
2. Configure the Oracle APEX login process
3. Define Oracle APEX authorizations
4. Define Oracle E-Business Suite profile option and form functions
5. Link E-Business Suite functions to menus and responsibility

Creating an Oracle APEX Application

Developers define applications within Oracle APEX as a collection of pages. Pages generally have one or more regions, buttons, items, computations, processes, branches, dynamic actions, tabs, lists, and breadcrumbs. Each application within an Oracle APEX instance has a unique application ID. For end-users to access an Oracle APEX application directly they enter a URL of the form:

http://<APEX_Host>:8080/apex/f?p=<Application_Id>:<Home_Page>

To create an Oracle APEX application, perform the following steps:

1. Navigate to the *Application Builder*
2. Click *Create*, select *Application Type* **New Application**

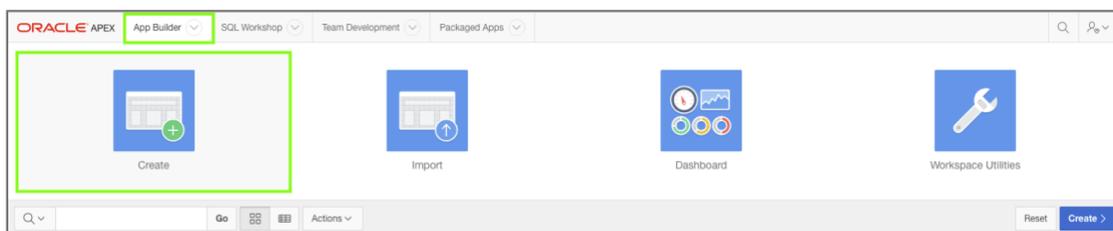


Figure 10. Creating an Oracle APEX Application

Note: All of the screenshots and steps provided are utilizing Oracle APEX 18.1. If you are using APEX 5.0, APEX 5.1, or a later release of Oracle APEX then the screenshots and steps may vary.

3. In the Create Application Wizard, enter *Name* **APEX EBS Demo**
4. Click *Add Page*, and select *Form*
5. In the Create Form Page dialog, enter Page Name **Update Email – Without Responsibility**
6. {Optional} Click *Set Icon*, and select an appropriate icon
7. For Table select `XX_APEX_EBS_USER` and click *Add Page*.

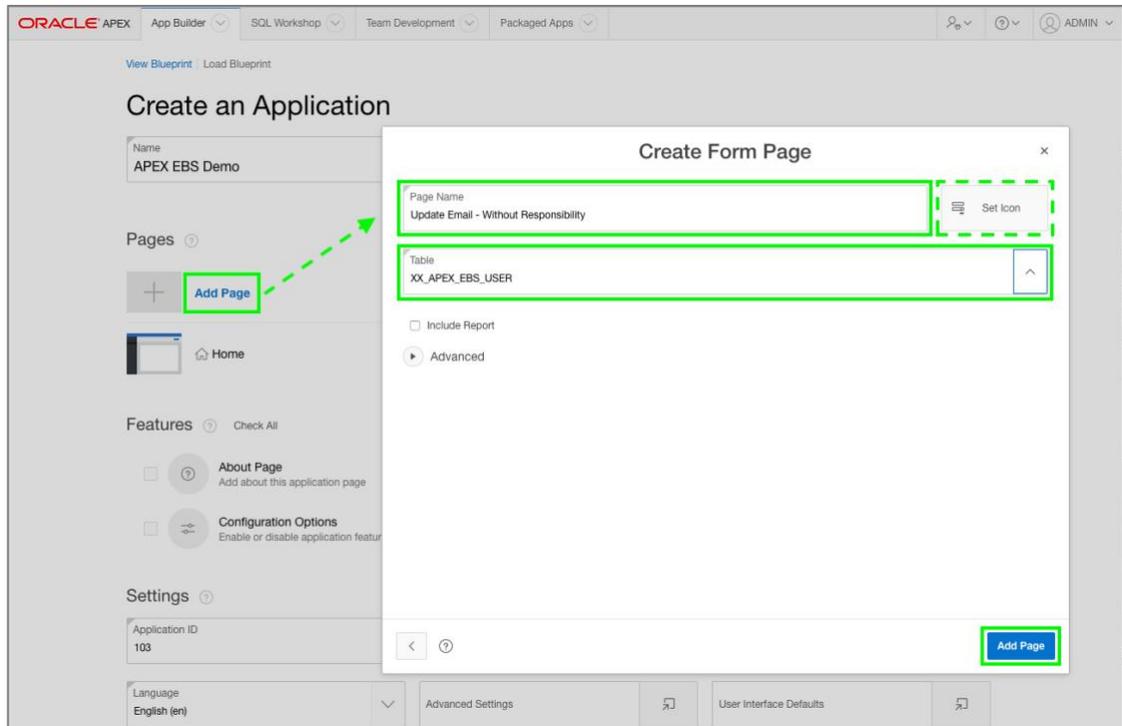


Figure 11. Adding a Page in Create Application Wizard

8. Click *Add Page*, and select *Form*
9. In the Create Form Page dialog, enter Page Name **Update Email – Using Responsibility**
10. {Optional} Click *Set Icon*, and select an appropriate icon
11. For Table select `XX_APEX_EBS_USER` and click *Add Page*.
12. Click *Create Application*.

Above you created an application with two form pages based on the `XX_APEX_EBS_USER` view. Now you will need to update these pages to work correctly with the view, and for the second page to correctly utilize the EBS responsibility.

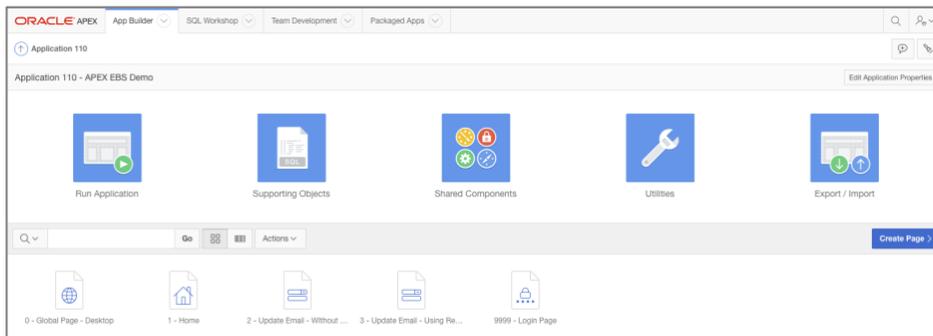


Figure 12. New application displaying pages generated

By default, the Create Application Wizard created a *Fetch Row from XX_APEX_EBS_USER* process to retrieve the data from the associated table or view. However, you need to update the generated process to use the Oracle APEX user name:

1. From the Application Home Page, click 2 – Update Email – Without Responsibility, which will bring up Page Designer and display Page 2.
2. In the left pane, within the Rendering tab, expand the *Pre-Rendering* node.
3. Click the *Fetch Row from XX_APEX_EBS_USER* process
4. In the right pane, within the Property Editor, find the *Primary Key Column* attribute
5. Replace the value **ROWID (Invalid)** with **USER_NAME**
6. In the right pane, within the Property Editor, find the *Primary Key Item* attribute
7. Replace the value **P2_ROWID** with **APP_USER**
8. At the top of the page, click *Save*

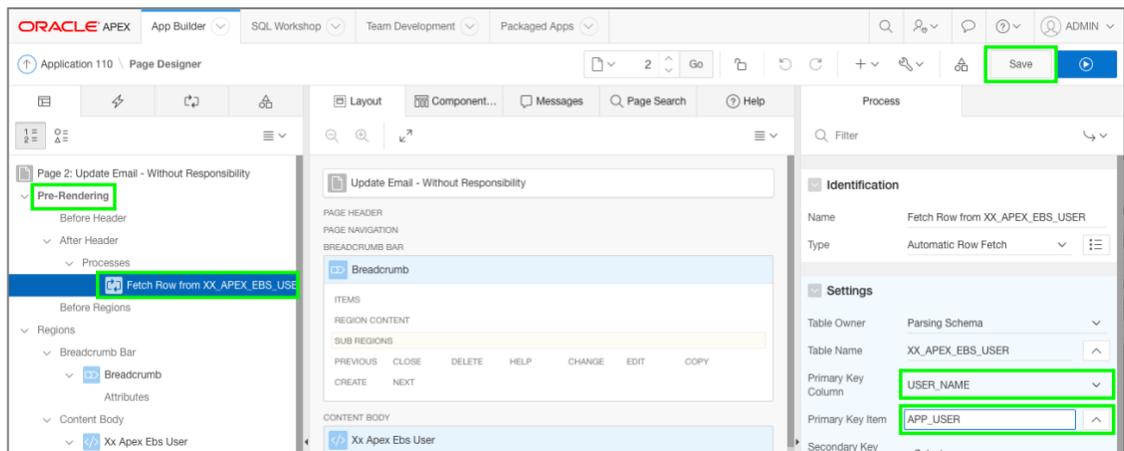


Figure 13. Updating the Pre- Rendering process

The items on the page were generated as text items by default. For this example, we only want the user to be able to update the Email Address, so we need to update the page and change item properties:

1. From the Application Builder, within Page Designer – Ensure you are on Page 2.
2. In the left pane, within the Rendering tab, select Xx Apex Ebs User region.
3. In the right pane, within the Property Editor, update the *Title* attribute, to APEX EBS User
4. In the left pane, within the Rendering tab, under APEX EBS User region, click P2_USER_ID
5. Hold the SHIFT key and click P2_DESCRIPTION
6. Hold the Control (CTRL) key and click P2_PERSON_PARTY_ID
7. In the right pane, within the Property Editor, find the *Identification* > *Type* attribute
8. Use the select list to update the type to **Display Only**.
9. Click *Save*

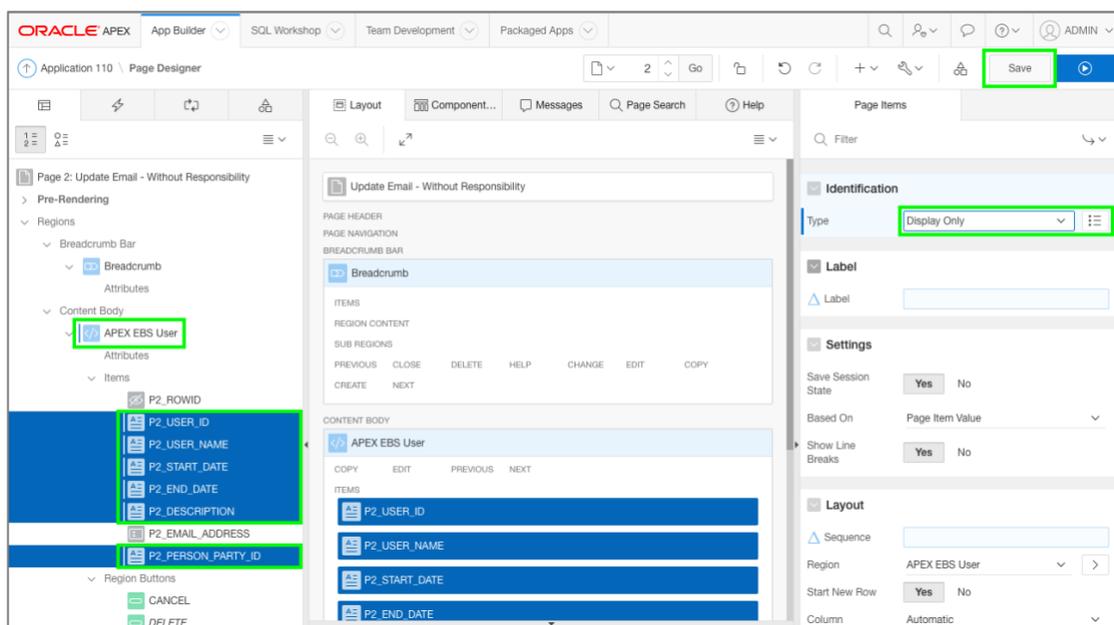


Figure 14. Updating Items to Display Only

The wizard also created a process called *Process Row of XX_APEX_EBS_USER* which is an Automatic Row Processing (DML) process to perform inserts, updates and deletes on the specified table or view. However, you need to replace the process with a call to the *apex_update_email* procedure defined earlier:

1. From the Application Builder, within Page Designer – Ensure you are on Page 2.
2. In the left pane, click the Processing tab (3rd tab).
3. Right-click the *Process Row of XX_APEX_EBS_USER* process and select *Delete*.

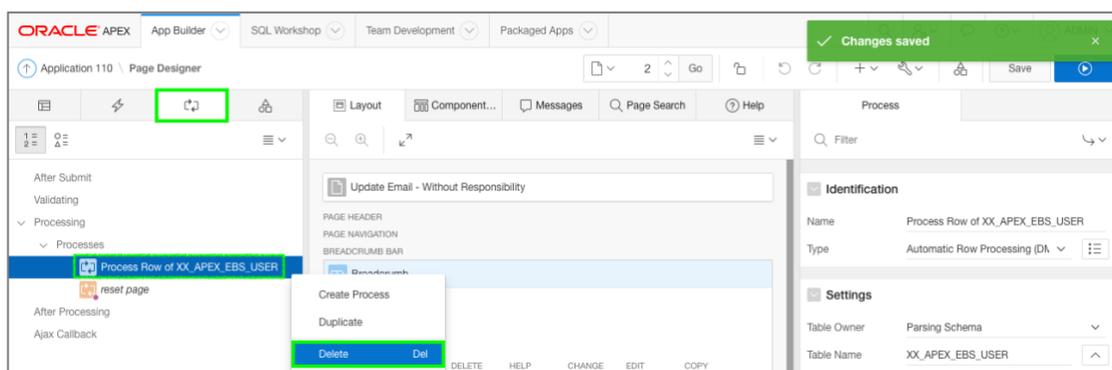


Figure 15. Deleting the generated process

4. Right-click on *Processes* region, click *Create Process*
5. In the left pane, within the Processing tab, drag the 'New' process above the 'reset page' process
6. In the right pane, within the Property Editor, enter the following:
 - Identification > Name – enter **Update Email**
 - Source > PL/SQL Code - enter:


```
apps.xx_apex_sample_apis.apex_update_email
( p_username      => :APP_USER
, p_owner         => :APP_USER
, p_email_address => :P2_EMAIL_ADDRESS
);
```
 - Success Message > Success Message – enter **Email updated successfully**
 - Error > Error Message – enter **Email not updated**
 - Server-side Condition > When Button Pressed – select *SAVE*
7. In the left pane, within the Rendering tab, drag the *Update Email* process above the *reset page* process.
8. Click *Save*.

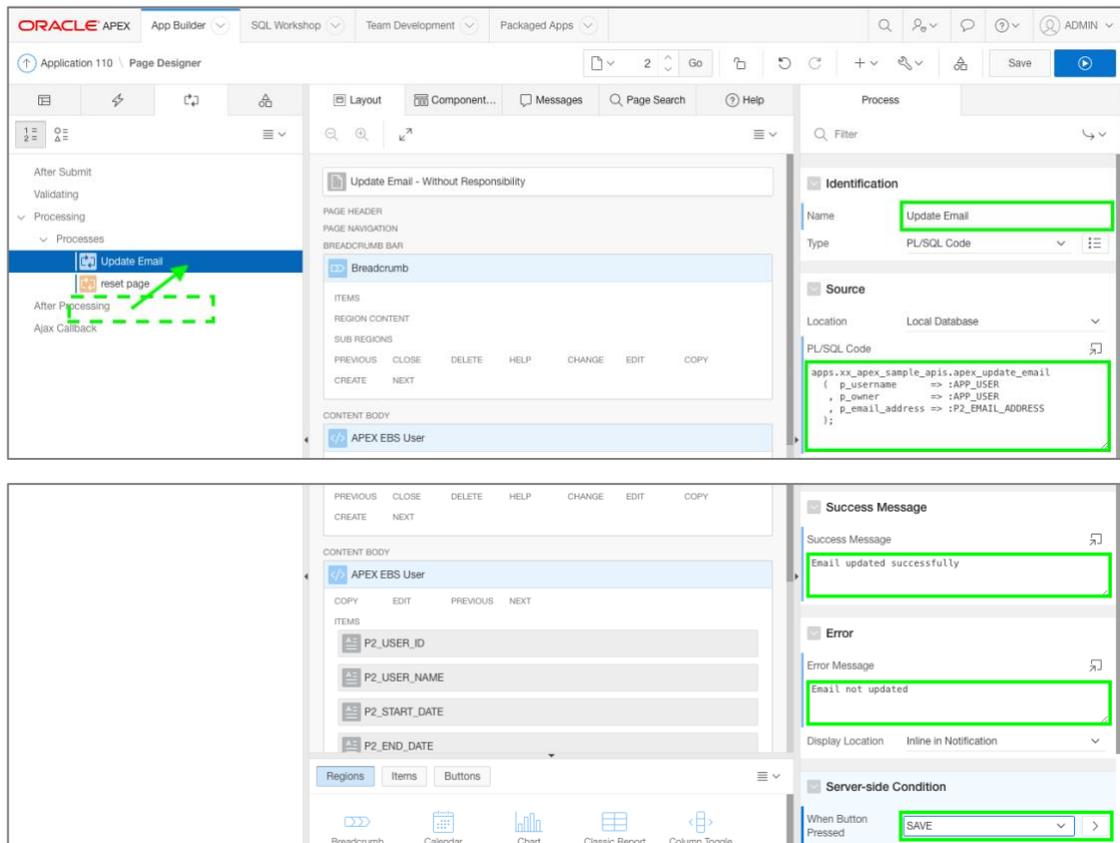


Figure 16. Adding a new process

You also need to perform the same operations on Page 3 as you performed on Page 2, namely:

- Update the Pre-Rendering process [Page 25]
- Update Items to Display Only [Page 26]
- Deleting the current process and creating Update Email process [Page 27]

To show the Oracle E-Business Suite responsibility that is going to be passed to Page 3 you must add a new region, two items, and a process.

To create the region and page items use the following steps:

1. From the Application Builder, within Page Designer – Ensure you are on Page 3.
2. In the left pane, click on the Rendering tab (1st tab).
3. Right-click on *Content Body*, and select *Create Region*.

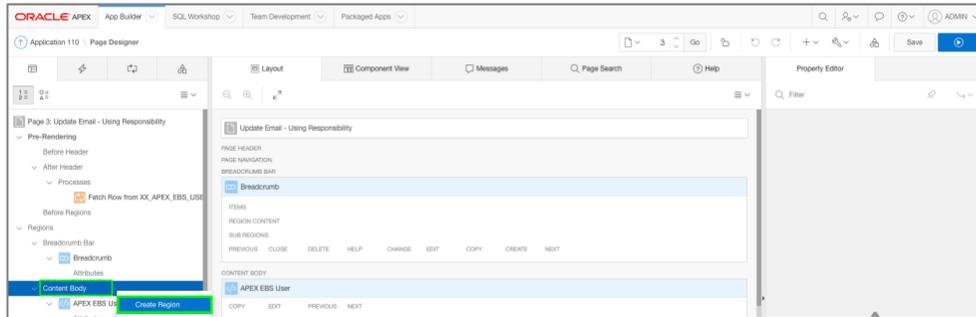


Figure 17. Creating a region on Page 3

4. In the right pane, within the Property Editor, enter the following: Identification > Title – enter **Responsibility Details**
5. In the left pane, within the Rendering tab, right-click the *Responsibility Details* region, and select *Create Page Item*.
6. In the right pane, within the Property Editor, enter the following:
 - Identification > Title – enter **P3_RESPONSIBILITY_NAME**
 - Identification > Type – select *Display Only*
 - Source > Type – select *Null*

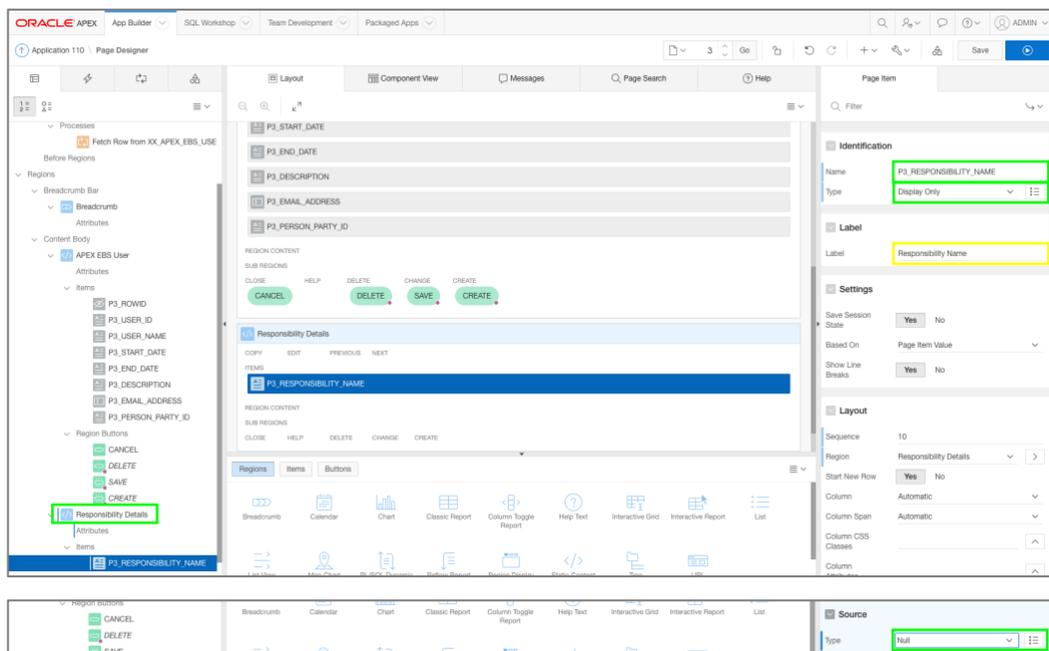


Figure 18. Creating a page item

7. In the left pane, within the Rendering tab, right-click the *P3_RESPONSIBILITY_NAME* item, and select *Duplicate*.
8. In the right pane, within the Property Editor, enter the following:
 - Identification > Title – enter **P3_RESP_DESCRIPTION**
 - Label > Label – enter **Description**

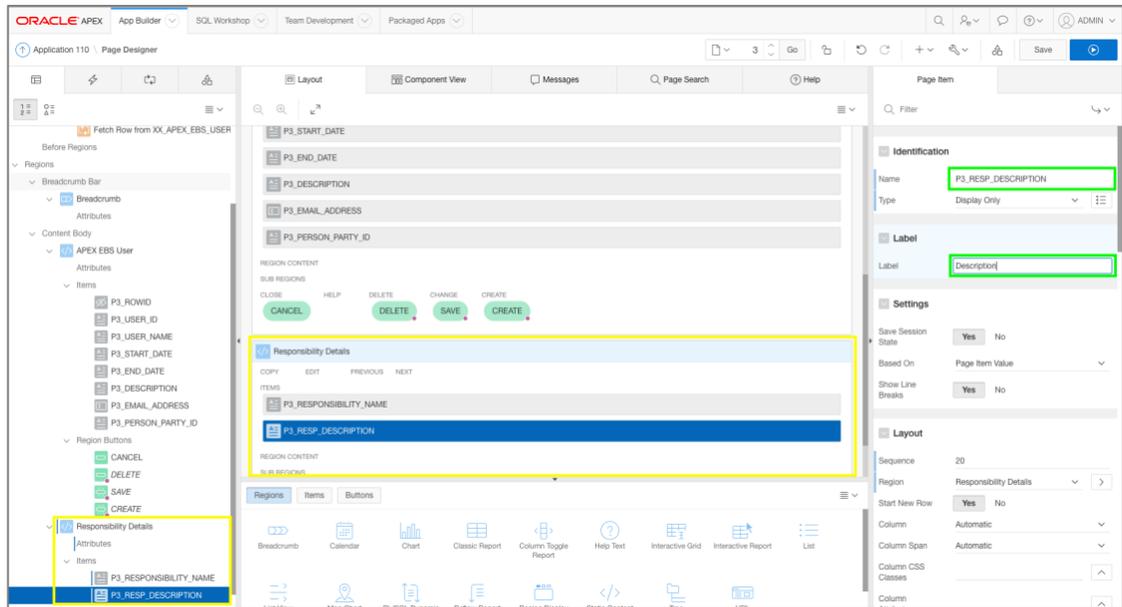


Figure 19. New region and items created on Page 3

To create the process required to retrieve the responsibility details use the following:

1. In the left pane, within the Rendering tab, expand Pre-Rendering
2. Right-click on *Processes*, and select *Create Process*.
3. In the right pane, within the Property Editor, enter the following:
 - Identification > Name - enter **Fetch Responsibility**
 - Source > PL/SQL Code - enter:

```

begin
  for c1 in (select responsibility_name
            ,      description
            from apps.fnd_responsibility_vl
            where application_id      = :EBS_APP_ID
            and  responsibility_id = :EBS_RESP_ID
            ) loop
    :P3_RESPONSIBILITY_NAME := c1.responsibility_name;
    :P3_RESP_DESCRIPTION    := c1.description;
  end loop;
exception
  when others then
    null;
end;
    
```

4. Click *Save*.

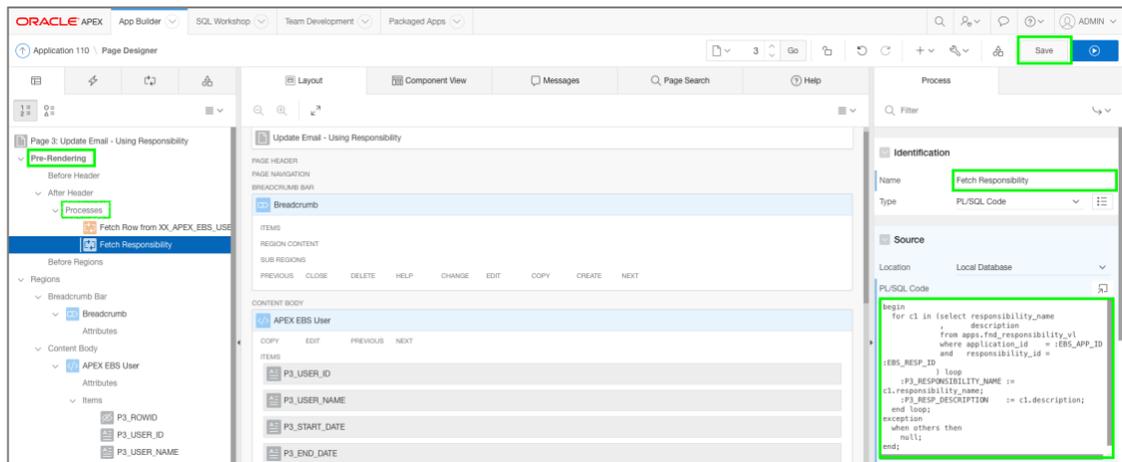


Figure 20. Create process to populate responsibility

At this stage the application is using default Oracle APEX user credentials. Run the application and enter the user name and password you defined when creating the XX_APEX Workspace earlier:

1. From page Designer, click *Application xxx*, in the breadcrumb (top left), to navigate back to the Application Home Page.
2. Click *Run Application*.
3. Enter *Username ADMIN*, *Password APEX_Password1*, click *Sign In*
4. Navigate using the icons and tabs

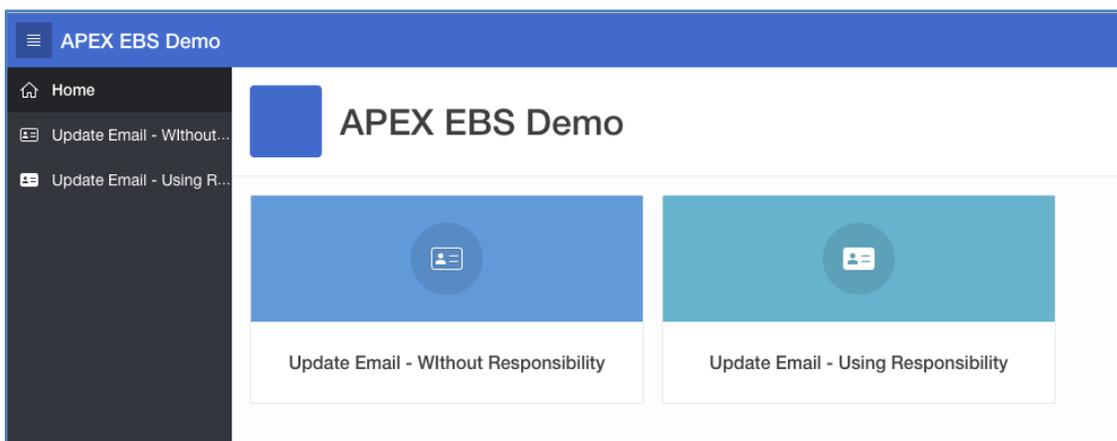


Figure 21. Running the Oracle APEX application

When clicking on Pages 2 or 3 you should receive an **ORA_01403: no data found error**. This is expected behavior as the Application User is currently the “Admin” user you defined within Oracle APEX, and not an EBS user that would be found in the XX_APEX_EBS_USER view. In other words, you are currently using Oracle APEX credentials instead of Oracle E-Business Suite credentials. Later in the document we will define authentication schemes that allow signing in using the E-Business Suite credentials. After defining the correct authentication user data will be displayed.

Oracle APEX Authentication and Authorization

Oracle APEX provides “out-of-the-box” mechanisms to handle both authentication (Is the user a valid Oracle E-Business Suite user?) and authorization (What privileges / functions does the user have within the application while using this responsibility?).

Authentication schemes check the user’s user name and password credentials before the user is allowed to access the application. Authorization schemes, on the other hand, control display and user access to pages, regions, items, buttons, and processes within an application.

You can also define plug-in authentications and plug-in authorizations. The benefit of developing such plug-ins is to ensure consistency across different applications. [See: Advanced Programming Techniques – Implementing Plug-Ins

<https://docs.oracle.com/database/apex-18.1/HTMDB/implementing-plug-ins.htm#HTMDB27001>].

User Authentication

There are several pre-configured authentication schemes defined within Oracle APEX that can be utilized, specifically Oracle Single Sign-On (OSSO), HTTP Header Variable for use with Oracle Access Manager (OAM), or custom authentication schemes [See: Managing Application Security – Establishing User Identity Through Authentication <https://docs.oracle.com/database/apex-18.1/HTMDB/establishing-user-identity-through-authentication.htm#HTMDB12003>].

Once you have configured OAM, OSSO or a custom authentication, you will be able to log into your Oracle APEX applications using any valid OAM, OSSO or Oracle E-Business Suite user name and password.

Configuring the Oracle APEX Login Process

If using custom authentication, users will not automatically log into Oracle APEX. However, if you are using Oracle Identity Manager or Oracle Single Sign-On then users will not be required to enter their user credentials when Oracle APEX is invoked from the Oracle E-Business Suite menu.

Configuring Oracle Access Manager

Many Oracle E-Business Suite installations use Oracle Access Manager (OAM) or Oracle Single Sign-On (OSSO) to provide a centralized method of authenticating users. The Oracle standard for single user authentication with Oracle E-Business Suite Release 12 is Oracle Access Manager.

If you are using OAM then follow the Integrating Oracle APEX with Oracle Access Manager white paper available from the Oracle Technology Network (OTN): <http://www.oracle.com/technetwork/developer-tools/apex/learnmore/apex-oam-integration-1375333.pdf>. You will need to integrate Oracle Access Manager with Application Express separately from integration with E-Business Suite, but they can both use the same components. Once they are both configured then they can utilize the same user authentication.

Configuring Single Sign-On

If you are using OSSO with your Oracle E-Business Suite, you will have to identify the Oracle APEX engine as a partner application within the OSSO infrastructure. Once you have configured the OSSO infrastructure, you can simply create an Oracle APEX authentication scheme using the name given to the OSSO entry to integrate user authentication with your Oracle E-Business Suite environment. An OSSO-based authentication scheme will use the OSSO login page for users of your Oracle APEX application. This method has the downside that a new session will be established each time Oracle APEX is called from an Oracle E-Business Suite function, thereby losing any session information previously established. Application developers should keep this in mind when developing applications.

Before configuring Single Sign-On you will need the following:

- Download the appropriate OSSO SDK (Software Development Kit)
- Details regarding the OSSO Server – Site Id, Site Token, Encryption Key, Single Sign-On URL, Single Sign-Off URL
- A copy of **custom_auth_sso_902.sql** and **custom_auth_sso_902.plb** from the Oracle APEX installation – **apex/core** directory

To configure Oracle APEX as a partner application within OSSO perform the following steps:

1. Copy the SSO SDK to the Oracle E-Business Suite Database Server
2. Unzip the SSO SDK file
3. Copy **custom_auth_sso_902.sql** and **custom_auth_sso_902.plb** to the *packages* directory (created when the SSO SDK file was unzipped)
4. Connect to your Oracle E-Business Suite Database server and navigate to the *packages* directory
5. Run SQL*Plus, connect as SYS AS SYSDBA – **sqlplus / AS SYSDBA**
(All following statements will be run from the SQL prompt; Replace all passwords with a case sensitive password which is not the same as the user being altered)

```

alter user APEX_180100 unlock identified by <APEX_180100_Password>;

alter user APEX_PUBLIC_USER unlock identified by
<APEX_PUBLIC_USER_Password>;

alter user XX_APEX identified by <XX_APEX_Password>;

connect XX_APEX/<XX_APEX_Password>

@loadsdk.sql

@regapp.sql (The Listener Token will be HTML_DB:<EBS_Hostname>:8080 ,
specify the Partner App Name XX_APEX_SSO; Use the values determined
above for Site ID, Site Token, etc.)

grant execute on wwsec_sso_enabler_private to APEX_180100;

connect APEX_180100/<APEX_180100_Password>

@custom_auth_sso_902.sql

@custom_auth_sso_902.plb

grant execute on wwv_flow_custom_auth_sso to APEX_PUBLIC_USER;

connect APEX_PUBLIC_USER/<APEX_PUBLIC_USER_Password>

create public synonym wwv_flow_custom_auth_sso for
apex_180100.wwv_flow_custom_auth_sso

exit

```

Once you have configured the Single Sign-On you will now need to define the authentication scheme within Oracle APEX. Follow these steps to create your SSO authentication:

1. Log into Oracle APEX Application Builder
(**http://<EBS_Hostname>:8080/apex/apex**)
2. From the Application Builder for the APEX EBS Demo application – go to Shared Components

3. Click *Authentication Schemes*, click *Create*, select **Based on a pre-configured scheme from the gallery**, click *Next >*
4. Select Oracle Application Server Single Sign-On (My Application as Partner App), click *Next >*
5. Enter *Partner Application Name* **APEX_EBS_EXTENSION_SSO**, click *Next >*
6. Enter *Name* **EBS SSO Login**, click *Create Scheme*
7. Click *Change Current Tab*, select Available Authentication Schemes **EBS SSO LOGIN**, click *Next >*, click *Make Current*
8. Click *Run Application*

If correctly configured, when you run the application, you will be redirected to the OSSO Login Page and can enter your Single Sign-on credentials to log into Oracle APEX.

Configuring Custom Authentication

If your Oracle E-Business Suite installation does not use Oracle Access Manager or Oracle Single Sign-On, you can create a custom authentication scheme which calls a PL/SQL function within Oracle APEX to validate the identity of the user, based on the password sent.

Below is sample code for creating a function to determine if the user is valid –

```
CREATE OR REPLACE FUNCTION xx_apex_validate_login
  ( p_username IN VARCHAR2
    , p_password IN VARCHAR2
  ) RETURN BOOLEAN
AS
BEGIN
  RETURN fnd_user_pkg.validate_login(p_username, p_password);
END apex_validate_login;
```

Note: This specific code is included in the **apex_ebs_setup.sql** file you downloaded and installed into the APPS schema earlier in this paper. This code is included within that file in *Package xx_apex_sample_apis, Function apex_validate_login*. This package function is used below when defining a custom authentication.

Once you have created this function in your APPS schema you need to define the authentication scheme within Oracle APEX. Follow these steps to create your custom authentication:

1. Log into Oracle APEX Application Builder
(http://<EBS_Hostname>:8080/apex/apex)

2. From the Application Builder for the APEX EBS Demo application – go to Shared Components
3. Click *Authentication Schemes*, click *Create*, select **Based on a pre-configured scheme from the gallery**, click *Next >*
4. Enter *Name* **EBS Custom Login**, select *Schema Type* **Custom**
5. For settings enter the following:
 - Authentication Function Name – enter **apps.xx_apex_sample_apis.apex_validate_login**
6. Click *Create Authentication Scheme*.

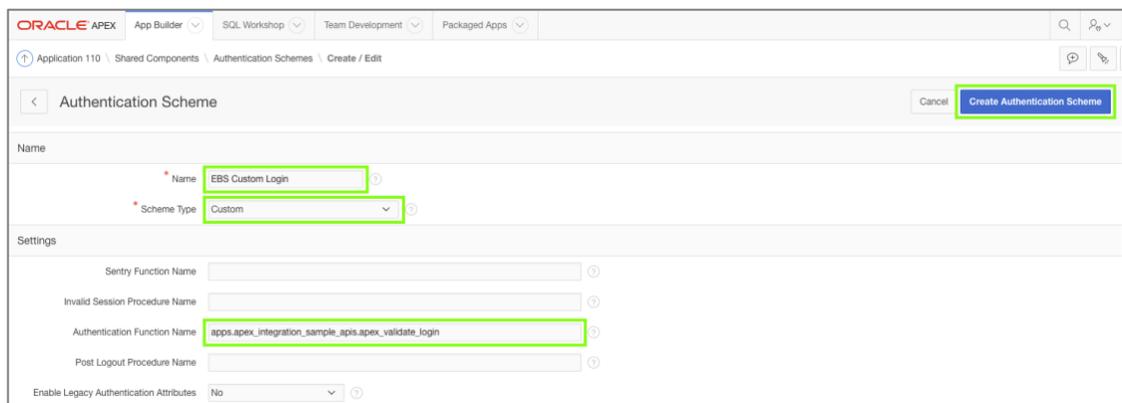


Figure 22. Creating a custom authentication

Run the application to ensure it is working correctly:

1. Navigate to the browser / tab where the runtime application is running.
2. Click the username 'Admin' (top right), and then click *Sign Out*.
3. On the login page, for Username enter **sysadmin**, and for Password enter the EBS sysadmin password.
4. Navigate to Pages 2 and 3 you should have zero errors and see details displayed.

Defining Oracle APEX Authorizations

Authentication schemes only check that the user has a valid Oracle E-Business Suite login, not that the user is authorized to use an application. Oracle APEX provides the ability to define authorization schemes that can be used on every component within an application (for example pages, regions, buttons, items, validations, etc.) to restrict access [See: Managing Application Security – Providing Security Through Authorization <https://docs.oracle.com/database/apex-18.1/HTMDB/providing-security-through-authorization.htm#HTMDB12004>].

Oracle E-Business Suite includes the concept of responsibilities, which based on the underlying menus, determines what capabilities and data users are allowed to access. The recommended practice is to create Oracle APEX authorization schemes that mimic the necessary Oracle E-Business Suite responsibilities to define cohesive access plans.

It is very strongly recommended that authorizations are applied to all non-public Oracle APEX pages (and those limited by Oracle E-Business Suite responsibilities) rather than just to menu items. Oracle recommends defining an authorization scheme at the application level referencing the Oracle E-Business Suite form function which called the APEX application. Session state protection should also be defined for the application to prevent users from accessing pages by manipulating the URL [See: Managing Application Security – Understanding Developer Security Best Practices <https://docs.oracle.com/database/apex-18.1/HTMDB/understanding-developer-security-best-practices.htm#HTMDB25974>].

Using Oracle E-Business Suite Responsibilities with Integrated Access

By modifying the function call used to invoke Oracle APEX from Oracle E-Business Suite you can pass the parameters required to test the responsibility directly. By defining an Oracle APEX authorization scheme which first performs an APPS_INITIALIZE and then returns the result from FND_FUNCTION.TEST you can use this authorization to prevent unauthorized access to pages and processes to users without the correct responsibility.

To correctly test Oracle E-Business Suite responsibilities it is necessary to send parameters across to Oracle APEX so they can be used when calling APPS_INITIALIZE from Oracle APEX. The three parameters are: Responsibility Id, Application Id, and Security Group Id.

In order to call APPS procedures and functions from another schema it is necessary to define a package with DEFINER authorization such that the PL/SQL is run with the privileges of the APPS schema rather than the privileges of the Oracle APEX schema.

Below is sample code for creating such a DEFINER Package in the APPS schema –

```
CREATE OR REPLACE PACKAGE xx_apex_global AUTHID DEFINER AS
  PROCEDURE apps_initialize(
    user_id IN NUMBER,
    resp_id IN NUMBER,
    resp_appl_id IN NUMBER,
    security_group_id IN NUMBER DEFAULT 0,
    server_id IN NUMBER DEFAULT -1);

  FUNCTION function_test(function_name IN VARCHAR2) RETURN BOOLEAN;
end xx_apex_global;
/

CREATE OR REPLACE PACKAGE BODY xx_apex_global AS
  PROCEDURE apps_initialize(
    user_id IN NUMBER,
    resp_id IN NUMBER,
    resp_appl_id IN NUMBER,
    security_group_id IN NUMBER DEFAULT 0,
    server_id IN NUMBER DEFAULT -1) IS
  BEGIN
    fnd_global.apps_initialize(user_id, resp_id, resp_appl_id,
                              security_group_id, server_id);
  END;

  FUNCTION function_test(function_name IN VARCHAR2) RETURN BOOLEAN IS
  BEGIN
    RETURN fnd_function.test(function_name);
  END;
END xx_apex_global;
/
```

Note: This specific code is included in the **apex_ebs_setup.sql** file you downloaded and installed into the APPS schema earlier in this paper.

You need to create three Oracle APEX Application Items to hold the three Oracle E-Business Suite parameters - Responsibility Id, Application Id, and Security Group Id, that will be passed by the Oracle E-Business Suite function.

Follow these steps to add Application Items into Oracle APEX:

1. Log into Oracle APEX Application Builder
(http://<EBS_Hostname>:8080/apex/apex)
2. From the Application Builder for the APEX EBS Demo application – go to Shared Components
3. Click Application Items, click Create >, enter Name **EBS_RESP_ID**, for Session State Protection select *Unrestricted*, click Create Application Item
4. Click *Create* >, enter Name **EBS_APP_ID**, select *Unrestricted*, click *Create Application Item*
5. Click *Create* >, enter Name **EBS_SEC_GROUP**, select *Unrestricted*, click *Create Application Item*

Below is sample code for defining an Oracle APEX authorization –

```

DECLARE
  l_user_id  NUMBER DEFAULT 0;
  l_resp_id  NUMBER DEFAULT 0;
  l_app_id   NUMBER DEFAULT 0;
  l_sec_group NUMBER DEFAULT 0;
BEGIN
  for c1 in (select user_id from XX_APEX_EBS_USER
             where user_name = :APP_USER
             ) loop
    L_user_id := c1.user_id;
  end loop;
  begin
    select nvl(to_number(:EBS_RESP_ID),0) into l_resp_id from dual;
    select nvl(to_number(:EBS_APP_ID),0) into l_app_id from dual;
    select nvl(to_number(:EBS_SEC_GROUP),0) into l_sec_group from dual;
  exception
  when others then
    l_resp_id := 0;
    l_app_id := 0;
    l_sec_group := 0;
  end;
  -- Call Initialize with the User (based on APEX log in) and the
  -- Responsibility Id, Account Id, and Security Group if passed from
EBS
  if l_resp_id <> 0 then
    apps.xx_apex_global.apps_initialize( l_user_id, l_resp_id
                                         , l_app_id, l_sec_group);
  else
    RETURN FALSE;
  end if;

  -- Check if User has permission on EBS Function
  RETURN apps.xx_apex_global.function_test('XX_APEX_DEMO_2');
END;

```

Follow these instructions to create the Oracle APEX authorization scheme:

1. From the Application Builder for the APEX EBS Demo application – go to Shared Components
2. Click Authorization Schemes, click Create >, select Create Authorization Scheme **From Scratch**, click Next >
3. Enter *Name* **EBS_RESPONSIBILITY**, select *Schema Type* **PL/SQL Function Returning Boolean**, copy the SQL code above into PL/SQL Function Body, enter *Identify error message* **Access Denied – You do not have access to the required function based on your Oracle E-Business Suite responsibility!**, click *Create Authorization Scheme*

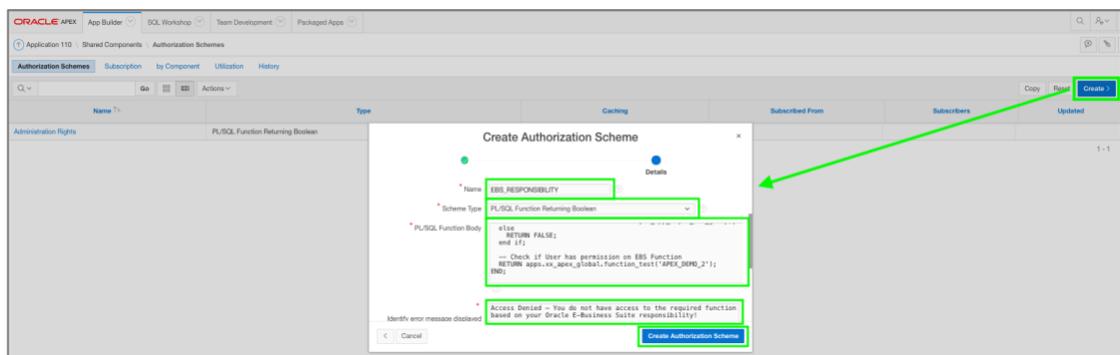


Figure 23. Defining Oracle APEX Authorization Scheme

Follow these instructions to restrict Page 3 using the APEX Authorization:

1. From the Application Builder for the APEX EBS Demo application - Edit Page 3
2. In the right pane, within the Property Editor, enter the following:
 - Security > Authorization Scheme – select *EBS_RESPONSIBILITY*
 - Security > Page Access Protection – select *Unrestricted*
3. Click *Save*.
4. Run the application
5. Click Using Responsibilities – Access is denied due to the authorization.

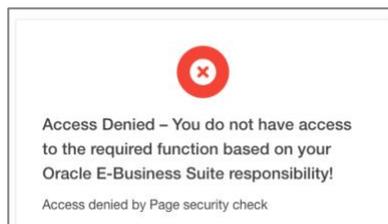


Figure 24. Authorization error message

Follow these instructions to allow deep linking to the application when called from E-Business Suite:

1. From the Application Builder for the APEX EBS Demo application, click Shared Components.
2. In Security (center pane), click *Security Attributes*.
3. Under Session Management, for Deep Linking select *Enabled*.
4. Click *Apply Changes*.

Installing the Sample Packaged Application

To quickly configure Oracle APEX within your Oracle E-Business Suite environment, a sample Oracle APEX application is available which includes working examples for updating the user's email address based on the steps performed previously in this paper.

To download from Oracle Technology Network (OTN) and install the sample application perform the following steps:

1. Download the Oracle APEX Sample Application script file from OTN – <http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/332107.sql>
2. Save the file as **apex_ebs_application_rev3.sql** on your local desktop
3. Log into Oracle APEX Application Builder
(**http://<EBS_Hostname>:8080/apex**)
4. From the Application Builder – click *Import*
5. Select *Import File* **apex_ebs_application_rev3.sql**, click *Next >*, click *Next >*, click *Install*
6. Click Run Application

Note: This application will produce errors at runtime unless the **apex_ebs_setup.sql** script has been run in the SYS schema as outlined earlier in this paper.

Sample Packaged Application in a Different Database

If you have installed Oracle APEX into a different Oracle Database to the database where E-Business Suite is installed, then the application above needs to be modified to access the appropriate view, namely `XX_APEX_DBLINK_EBS_USER`, and all calls to APPS need to be prefixed with the database link name, namely `XX_APEX_DBLINK`. You can also download the modified application from Oracle Technology Network (OTN) using these steps:

1. Download the Oracle APEX Sample DB Link Application script file from OTN <http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/2283639.sql>
2. Save the file as **apex_ebs_dblink_app_rev3.sql** on your local desktop
3. Log into Oracle APEX Application Builder
(**http://<Local APEX Home>:8080/apex**)
4. From the Application Builder – click *Import*
5. Select *Import File* **apex_ebs_dblink_app_rev3.sql**, click *Next >*, click *Next >*, click *Install*
6. Click Run Application

Defining Oracle E-Business Suite Profile and Form Functions

With Oracle E-Business Suite 11i you can redirect to Oracle APEX by creating a web-enabled `mod_plsql` procedure. However, with Oracle E-Business Suite Release 12 this integration must be implemented on the Oracle E-Business Suite Application Server as a Java Server Page (JSP). Oracle E-Business Suite provides a way of configuring URLs to Oracle APEX and will create the necessary JSP based on the defined parameters. This configuration requires setting an Oracle E-Business Suite profile option (FND: APEX URL) to contain the Oracle REST Data Services host name, and defining Oracle E-Business Suite functions that point to the actual APEX pages.

Update the FND: APEX URL profile option with the correct setting at the site level using the following steps:

1. Log in to Oracle E-Business Suite with the SYSADMIN user
(`http://<EBS_Hostname>:8000/OA_HTML/AppsLogin`)
2. Navigate to the System Administrator responsibility > Profile > System menu option
3. Search for *Profile %APEX%*, click *Find*
4. For profile option name FND: APEX URL enter *Site*
`http://<EBS_Hostname>:8080/apex`
5. Save the profile option value

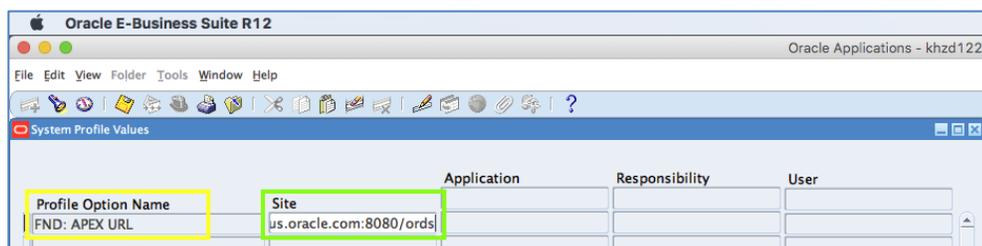


Figure 25. Updating Oracle E-Business Suite Profile option value

Note: All of the steps you need to perform within Oracle E-Business Suite can automatically be performed by running the `apex_ebs_env.sql` script outlined later in this paper.

Perform the following steps to define the Oracle E-Business Suite functions:

1. Navigate to the System Administrator responsibility > Application > Function menu option
2. For calls to the page without responsibility, create a function with the following details:

Description -

Function: **XX_APEX_DEMO_1**

User Function Name: **Update User Email (Without Responsibility)**

Description: **Update the current user email without checking the user responsibility.**

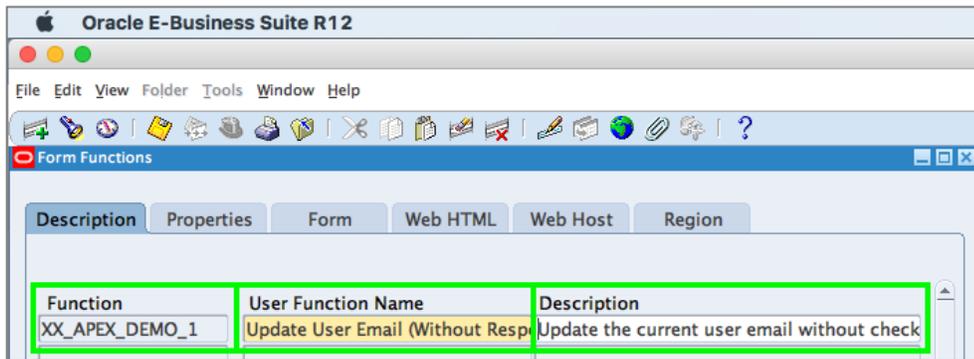


Figure 26. Creating Oracle E-Business Suite Function – Description Tab

Properties –

Type: **JSP** {Meaning: SSWA jsp function}

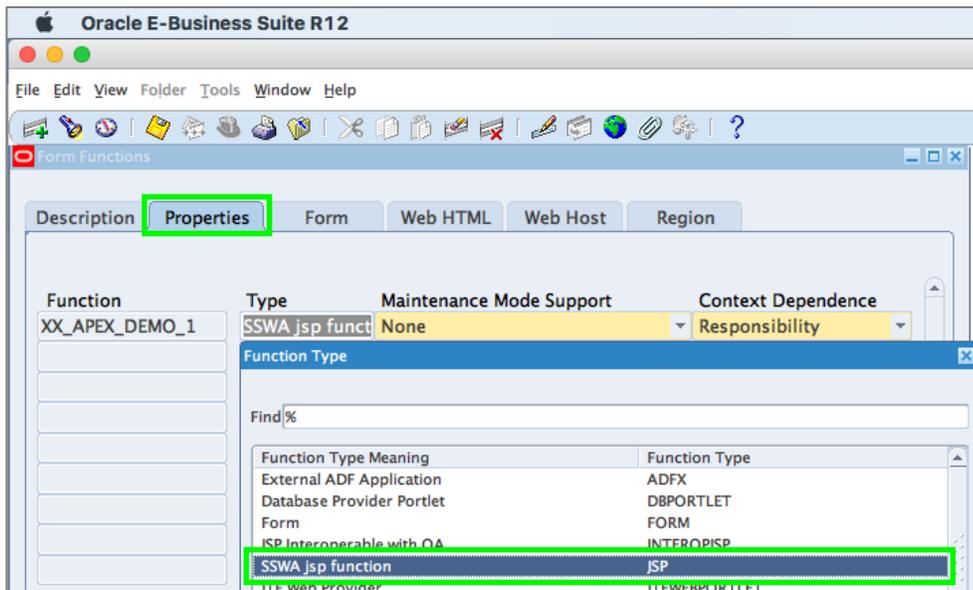


Figure 27. Creating Oracle E-Business Suite Function – Properties Tab

Web HTML –

HTML Call: `GWY.jsp?targetAppType=APEX& p=<APEX Application Id>:
<APEX Page>:<Session>:<Request>:<Debug>:<Clear Cache>:
<Parameter Pairs>
{For example, to call Oracle APEX application 109, Page 2
USE GWY.jsp?targetAppType=APEX&p=110:2, all other parameters are
optional}`

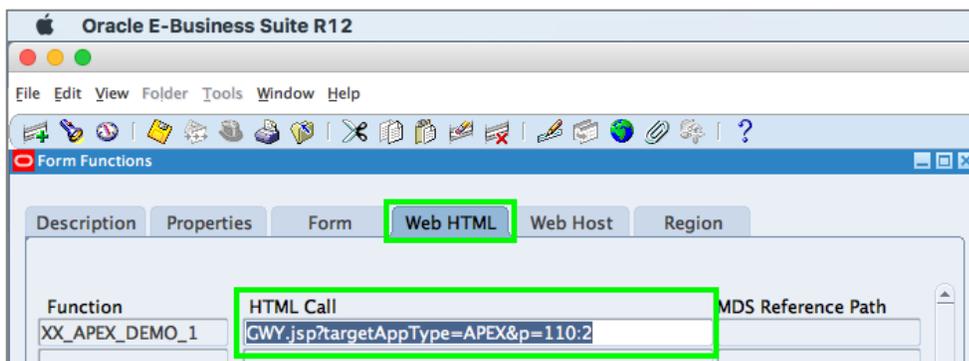


Figure 28. Creating Oracle E-Business Suite Function – Web HTML Tab

3. For calls to the page with responsibility create a function with the following details -
 - Function: **XX_APEX_DEMO_2**
 - User Function Name: **Update User Email (Using Responsibilities)**
 - Description: **Update the current user email after checking the user responsibility.
This option is more secure!**
 - Type: **JSP**
 - HTML Call: **GWY.jsp?targetAppType=APEX& p=<APEX Application Id>:
<APEX Page>:<Session>:<Request>:<Debug>:<Clear Cache>:
EBS_RESP_ID,EBS_APP_ID,EBS_SEC_GROUP:
[RESPONSIBILITY_ID],[RESP_APPL_ID],[SECURITY_GROUP_ID]
{For example, to call Oracle APEX application 109, Page 3 use
GWY.jsp?targetAppType=APEX&p=110:3::::EBS_RESP_ID, EBS_APP_ID,
EBS_SEC_GROUP:[RESPONSIBILITY_ID],[RESP_APPL_ID],
[SECURITY_GROUP_ID]}**
4. Save the functions

Defining Oracle E-Business Suite Menus and Responsibilities

The next step is to define menu options and responsibilities for your Oracle APEX applications within the Oracle E-Business Suite environment. Once the function is defined you may attach it to appropriate Menus and Responsibilities. Users having access to such responsibilities will be able to see the link in the Oracle E-Business Suite home page. Once they click the link the current browser will launch the targeted APEX page in Oracle APEX.

Perform the following steps to add an Oracle E-Business Suite menu:

1. Navigate to the System Administrator responsibility > Application > Menu option
2. Create a new menu -
 - Name: **XX_APEX_MENU**
 - User Menu Name: **XX_APEX Demo**
 - Menu Type: **Home Page**
 - Description: **Menu for Oracle APEX Demonstration app.**
3. Create menu component –
 - Sequence: **1**
 - Prompt: **Update Email**
 - Function: **Update User Email (Without Responsibility)**
{Form function XX_APEX_DEMO_1}
 - Description: **Update your email without using responsibility.**
4. Create menu component –
 - Sequence: **2**
 - Prompt: **Update Email (Using Responsibility)**
 - Function: **Update User Email (Using Responsibility)**
{Form function XX_APEX_DEMO_2}
 - Description: **Update your email using EBS Responsibility.**
5. Save the menu

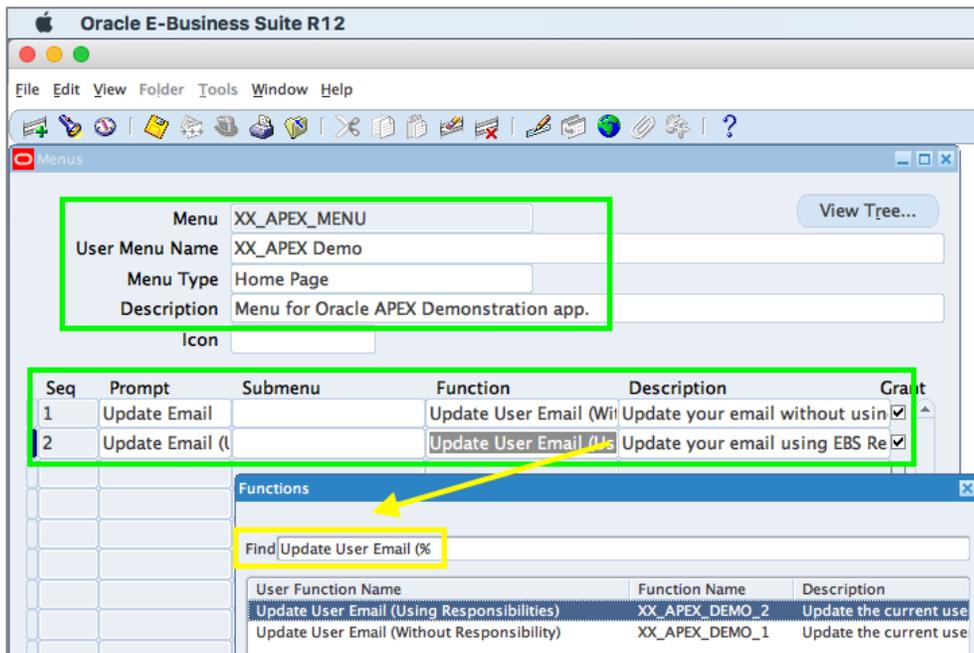


Figure 29. Creating Oracle E-Business Suite Menu

Perform the following steps to add an Oracle E-Business Suite responsibility:

1. Navigate to the System Administrator responsibility > Security > Responsibilities > Define option
2. Create a new responsibility -
 - Responsibility Name: **XX_APEX_RESP**
 - Application: **Application Object Library**
 - Responsibility Key: **XX_APEX_RESP**
 - Description: **Responsibility for APEX Demonstration**
 - Available From: **Oracle Self Service Web Applications**
 - Data Group: **Standard**
 - Application: **Application Object Library**
 - Menu: **XX_APEX Demo**
3. Save the responsibility

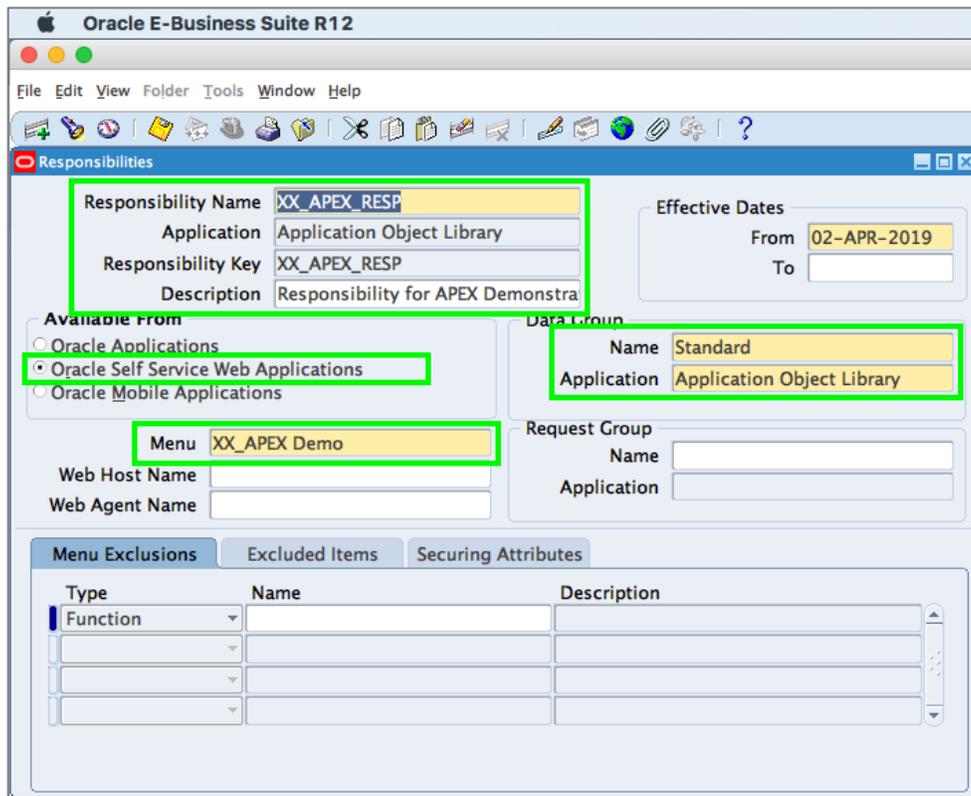


Figure 30. Creating Oracle E-Business Suite Responsibility

Perform the following steps to add a user:

1. Navigate to the System Administrator responsibility> Security > User > Define option
2. Create a new user -
 - User Name: **APEX**
 - Password: **Welcome0**
 - Direct Responsibilities:
 - Responsibility: **XX_APEX_RESP**
 - Description: **Add access to XX_APEX Demo menu options**
3. Save the user

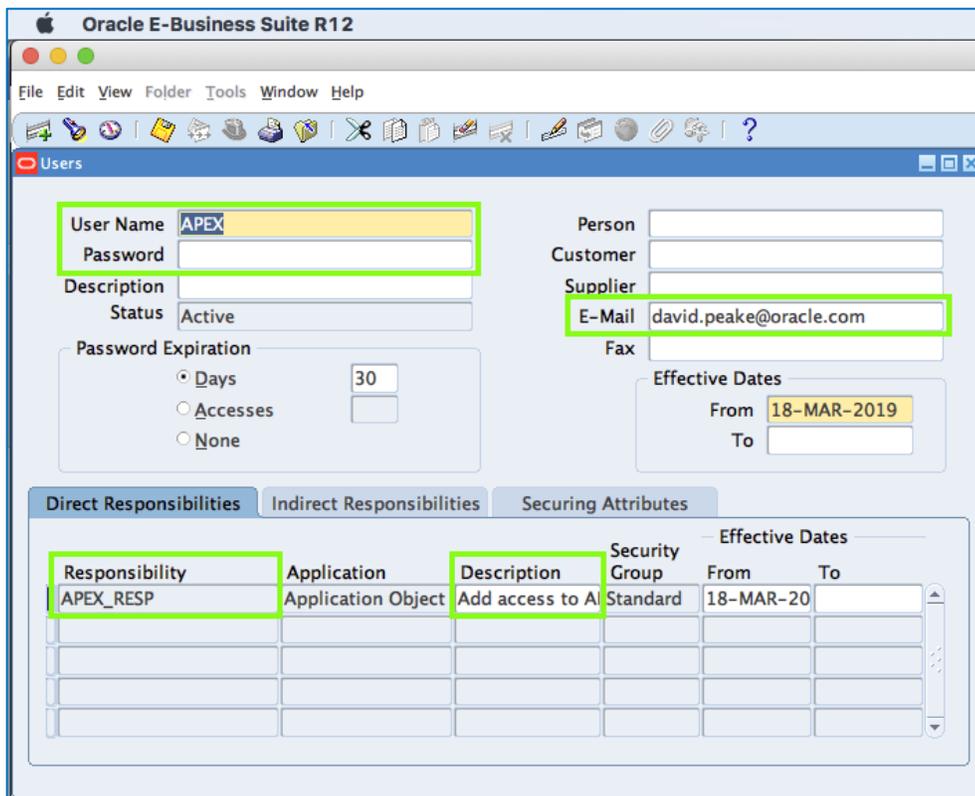


Figure 31. Creating Oracle E-Business Suite User

Perform the following steps to complete the configuration:

1. Navigate to *System Administrator responsibility > Concurrent > Requests*, click *Submit a New Request*, for Name select **Compile Security Menu**. Click *Submit*.
2. Click *Find* and review the request to ensure it has completed successfully.
3. Navigate to Functional Administrator responsibility > Core Services > Caching Framework
4. Click Global Configuration, click Clear All Cache, click Yes

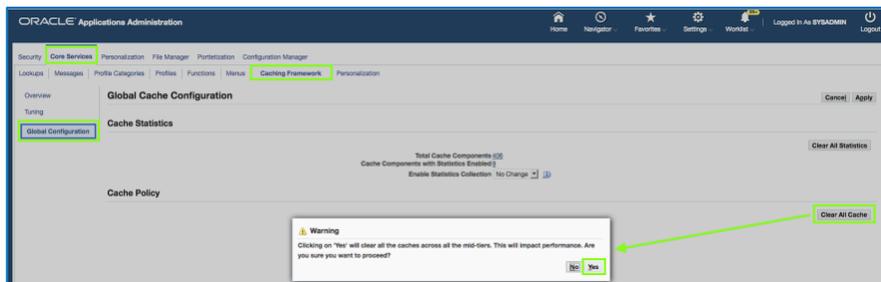


Figure 32. Clearing the Cache

5. Log out of Oracle E-Business Suite
6. Log in to Oracle E-Business Suite with the APEX user
[\(\[http://<EBS_Hostname>:8000/OA_HTML/AppsLogin\]\(http://<EBS_Hostname>:8000/OA_HTML/AppsLogin\)\)](http://<EBS_Hostname>:8000/OA_HTML/AppsLogin)

Downloading and Installing the Environment Script

The code samples given above for modifying Oracle E-Business Suite are available for download from the Oracle Technology Network (OTN). To run this script, perform the following steps:

1. Download the environment script file from OTN –
<http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/332106.sql>
2. Save the file as **apex_ebs_env.sql** and upload to your Oracle E-Business Suite Database server
3. Connect to your Oracle E-Business Suite Database server
4. Start SQL*Plus and connect as APPS
5. Run the script, providing the appropriate criteria:
@apex_ebs_env

Running the Application Express Extension

Now that you have successfully built the Oracle APEX extension you can run Oracle E-Business Suite and select your new menu links to access your Oracle APEX application. Follow these steps to test your application:

1. Log in to Oracle E-Business Suite using APEX / Welcome0
(http://<EBS_Hostname>:8000/OA_HTML/AppsLogin)
2. Navigate to the APEX Menu

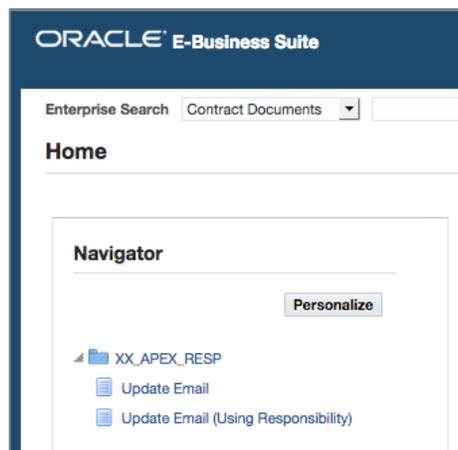


Figure 33. Oracle E-Business Suite runtime menu

3. Select *Update Email* – This will redirect you to the Oracle APEX login page or to the application depending on if you have configured OSSO above
4. If you need to log into Oracle APEX - enter *Username APEX*, *Password <<Welcome0>>*, click *Login*.
5. From the APEX application home page, click *Update Email – Without Responsibility*
6. Update the email address, click *Apply Changes* – This will return you to the application home page

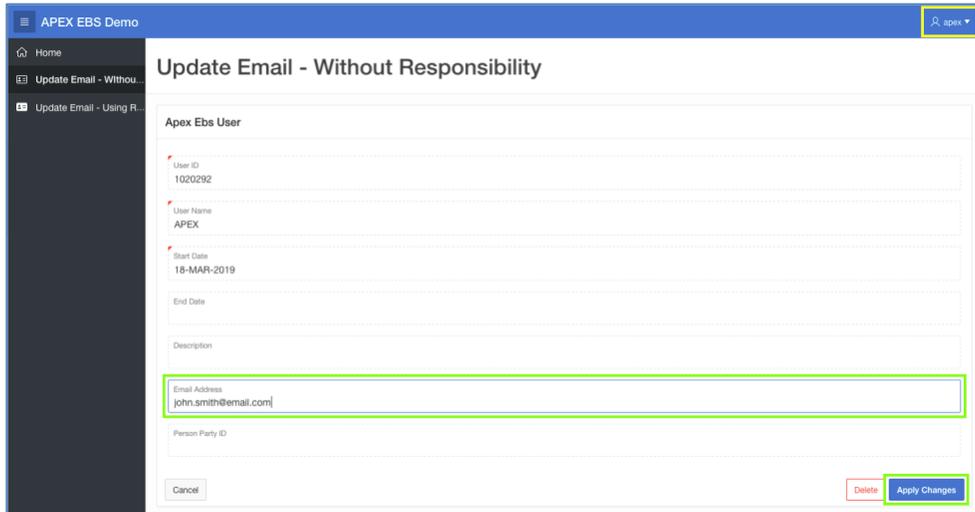


Figure 34. Oracle APEX application page for updating user email

7. Click *Update Email - Using Responsibility* – This will produce an authentication error as you navigated to the application without responsibilities
8. Navigate back to Oracle E-Business Suite
9. Select *Update Email (Using Responsibility)*. Update the email address, click *Apply Changes* – This will redirect you to the Oracle APEX application and set your Oracle E-Business Suite responsibility. The responsibility region will display your current responsibility

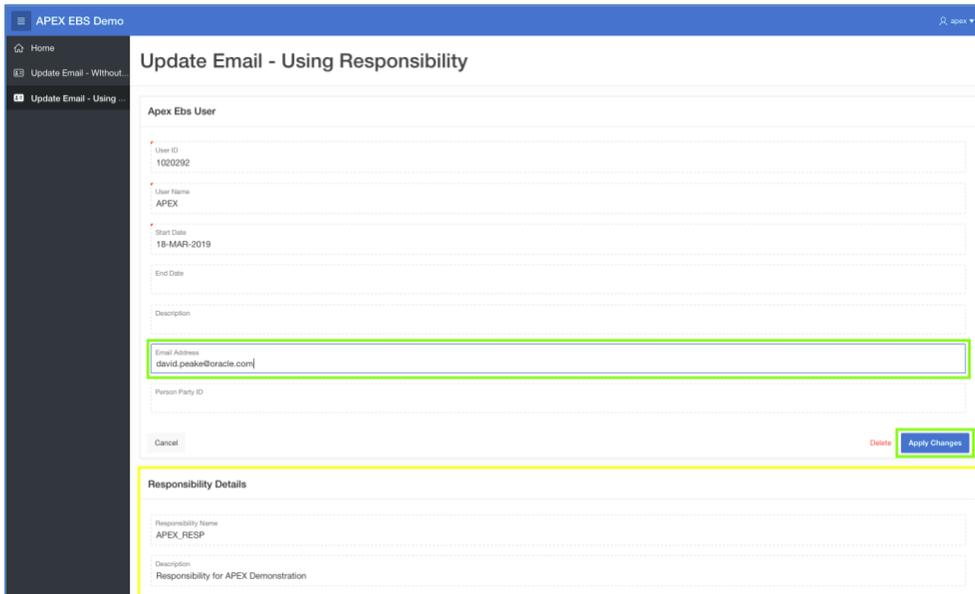


Figure 35. Oracle APEX application page showing responsibility

Custom Tables on Oracle E-Business Suite R12.2

Since your Oracle E-Business Suite custom schema (XX_APEX in this document), which you used to create the Oracle APEX application, is Edition-Based to comply with Oracle E-Business Suite Online Patching, there are a few differences related to your custom tables and how Oracle APEX handles DML operations.

Once your custom table is created, you need to generate the editioned view which will also create the grant and the synonym for APPS. For example, if you create a table in XX_APEX called XX_CUST_TABLE then execute the following EBS API, as the Apps user:

4. Connect to your Oracle E-Business Suite Database server
5. Start SQL*Plus and connect as APPS
6. Execute the following command:

```
exec AD_ZD_TABLE.UPGRADE('XX_APEX','XX_CUST_TABLE');
```

The custom editioned view created will have the same name as the custom table with the hast (#) sign at the end of the name. In order for the API to execute correctly it is essential that the custom schema name and the table name are in uppercase.

If creating triggers then you must create them on the editioned view and not the custom base table.

In the Oracle APEX applications, any access to custom tables should instead be referencing the corresponding editioned view. When utilizing the Create App Wizard or Create Page Wizard within Oracle APEX select one of the editioned views and complete the wizard as usual.

For existing applications, review any DML process having a custom table name and replace it with the editioned view of that table:

For example XX_CUST_TABLE becomes XX_CUST_TABLE#

On the Oracle E-Business Suite side it is business as usual! The AD_ZD_TABLE.UPGRADE API created a grant for APPS and a synonym on the name of the custom base table pointing to the editioned view. So using this object in views, PL/SQL packages, procedures or functions is totally transparent to the developer.

Upgrading your Application Express Extensions

Now that you have your initial Application Express extension to E-Business Suite deployed, it is important to understand how to enhance your Application Express application(s), within this integrated environment.

To fix bugs, modify existing functionality, enhance an application, or implement a new application, you will want to import an Application Express application into your QA / Test and then Production environments. This application import may also require database object upgrades.

When there is no modification or new database objects required, for the enhanced application(s), then the Application Express application(s) can simply be imported into the relevant environments. This can be performed without impacting the Oracle E-Business Suite environment and while users are utilizing the extensions. The user will pick up the latest application definition when they next access the imported application. If they were already in that application, then they will pick up the new application definition when they next request or submit a page.

If an updated or new Application Express application requires only new database objects, such as a new table, view, or database package, then these new database objects should be implemented into the environment before the Application Express application is imported. For Oracle E-Business Suite Release 12.2 environments, Oracle recommends creating these new database objects outside of the E-Business Suite Online Patching Cycle. Once the new database objects are available then the Application Express application can be imported. Again, this can be performed while users are utilizing the extensions.

However, if an updated or new Application Express application requires modification or updates to existing database objects or existing data then this should be performed while users are prevented from accessing the extensions. For Oracle E-Business Suite Release 12.1 and prior, this can be achieved by stopping Oracle WebLogic Server, modifying the database objects and data as required, importing the Application Express application, and then restarting Oracle WebLogic Server. For Oracle E-Business Suite Release 12.2 environments, Oracle recommends implementing these changes within an E-Business Suite Online Patching Cycle.

The following steps outline how to deploy such applications:

1. Initiate the Oracle E-Business Suite Online Patching Cycle, by running the Online Patching Prepare phase. Review the [Oracle E-Business Suite Maintenance Guide R12.2](#), Chapter 3 “Patching Procedures”, section “The Online Patching Cycle” for further details.
2. Run scripts to modify existing database objects and data within the “patch” edition
3. Run scripts to create new database objects within the “patch” edition (if required)

4. Shutdown the Oracle WebLogic Server used by Oracle APEX
5. Import updated and new Application Express applications using SQL*Plus
6. Complete the Oracle E-Business Suite Online Patching Cycle
7. Restart the Oracle WebLogic Server used by Oracle APEX

For example, if you wanted to enhance the application developed earlier in this paper to allow users to update “Known As” as well as their email address then you would need to modify existing database objects before importing the enhanced Application Express application. You would need to update the `XX_APEX_EBS_USER` view to include the `known_as` column from the `FND_USERS` table, and also update the `apex_update_email` function within the `xx_apex_sample_apis` package to accept an extra parameter and update the API call.

In an Oracle E-Business Suite Release 12.2 environment, the Oracle Database feature Edition-Based Redefinition (EBR) is employed within your database for Oracle E-Business Suite Release 12.2 Online Patching. You should utilize this capability to deploy changes to database objects within the schema(s) associated with Oracle APEX workspaces, such as `XX_APEX`. The Oracle recommendation above allows the Database Administrators to implement all modifications to database objects within the patch edition, so that users are not impacted, except for minimal downtime when the Oracle WebLogic Server is shutdown while the Application Express application is imported. In fact, it is critical to ensure that you do not implement such database object changes in the run edition, as any changes will be overwritten once the E-Business Suite Online Patch Cycle is completed.

Given that the Oracle APEX Engine (For Application Express Release 18.1 that schema is `APEX_180100`) is not editioned, when you import a new Application Express application definition, it will be “live” immediately to all users. This is irrespective of if Oracle E-Business Suite is currently in Online Patching Cycle or not. If the new application relies on updated database objects and those updates have been performed in the patch edition, then your applications will not see those updated database objects as they will be referencing the run edition. Therefore, it is very important to coordinate when required updates to underlying database objects and when Application Express applications are performed, to minimize disruption to users.

Upgrading your APEX Extensions in a Different Database

When you have installed Oracle APEX into a different Oracle Database to the database where E-Business Suite is installed, then existing database objects may need to be modified in the local database, where Oracle APEX is installed, or in the Oracle E-Business Suite database, or in both databases.

If an updated or new Application Express application requires database objects within the local database (within the XX_APEX_LOCAL schema) to be modified, then you should prevent user access while the database objects are modified and the application(s) are being imported. This can be achieved by stopping Oracle WebLogic Server, modifying the database objects and data as required, importing the Application Express application, and then restarting Oracle WebLogic Server.

If an updated or new Application Express application requires only new database objects within the Oracle E-Business Suite database (within either the APPS or XX_APEX schemas), such as a new table, view, or database package, then these new database objects should be implemented into the environment before the Application Express application is imported. For Oracle E-Business Suite Release 12.2 environments, Oracle recommends creating these new database objects outside of the E-Business Suite Online Patching Cycle. Once the new database objects are available then the Application Express application can be imported.

However, if an updated or new Application Express application requires modification or updates to existing database objects or existing data, within the Oracle E-Business Suite database, for Oracle E-Business Suite Release 12.2 environments, Oracle recommends implementing these changes within an E-Business Suite Online Patching Cycle.

The following steps outline how to deploy such applications:

1. Initiate the Oracle E-Business Suite Online Patching Cycle, by running the Online Patching Prepare phase. Review the [Oracle E-Business Suite Maintenance Guide R12.2](#), Chapter 3 “Patching Procedures”, section “The Online Patching Cycle” for further details.
2. Run scripts to modify existing database objects and data within the “patch” edition
3. Run scripts to create new database objects within the “patch” edition (if required)
4. Shutdown the Oracle WebLogic Server used by Oracle APEX
5. Run scripts to modify existing database objects and data on the local database
6. Import updated and new Application Express applications using SQL*Plus
7. Complete the Oracle E-Business Suite Online Patching Cycle
8. Restart the Oracle WebLogic Server used by Oracle APEX

Conclusion

Oracle APEX is an excellent complement to Oracle E-Business Suite. As a development framework, Oracle APEX allows cross-functional teams to easily collaborate and quickly deliver solid custom extensions to Oracle E-Business Suite. The Oracle APEX footprint on an Oracle E-Business Suite environment is negligible but allows deeply-integrated extensions using only standard customizations within the Oracle E-Business Suite implementation.

Use of a separate Oracle E-Business Suite custom schema for Oracle APEX with only the minimum required privileges for Oracle E-Business Suite objects ensures that Oracle APEX applications only have access to the interfaces and data required for their operation. This minimizes the security risk to the Oracle E-Business Suite inherent in making any extensions. By integrating with Oracle E-Business Suite Function Security (Authorization), Oracle APEX extensions utilize a single point of security administration. When a responsibility is added or removed from a user within Oracle E-Business Suite, the user's access to the associated functionality in Oracle APEX applications is affected likewise.

Further, the advantage of utilizing Oracle APEX over building extensions from scratch is that Oracle APEX provides a hardened declarative framework to minimize security vulnerabilities. Oracle APEX release procedures include extensive security testing to minimize the threat of security breaches from cross site scripting and SQL injection.

Oracle APEX gives you the power to quickly create forms, reports and complete applications rapidly, with little or no programming. With the proper deployment architecture, as outlined in this paper, you can use Oracle APEX to extend the capabilities of your Oracle E-Business Suite, using the same data for both sets of applications, sharing user authentication, and calling applications and components seamlessly.

Acknowledgements

This paper is a collaboration between the Oracle APEX and Oracle E-Business Suite teams. George Buzsaki, Oracle E-Business Suite Architecture, has provided significant input to ensure the solutions provided meets the Oracle E-Business Suite best practices, and presents fully supported configurations. Kevin Hudson, Oracle E-Business Suite Senior Director, responsible for architecting the E-Business Suite Online Patching Cycle, has provided significant input to ensure the Oracle APEX extensions work correctly with Oracle E-Business Suite 12.2.

A special thanks to Sylvain Martel from Insum Solutions for his reviews and invaluable input. Content assistance was provided by Rod West from Cabot Consulting and Wolfgang Moritz from Farwest Steel.



Extending Oracle E-Business Suite
Release 12.1 and above using
Oracle APEX
Revision 3.02

April, 2019
Author: David Peake
Contributing Authors: Kevin Hudson,
Joel Kallman, Santiago Bastidas

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



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

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 1010

Hardware and Software, Engineered to Work Together