



Hardware and Software
Engineered to Work Together

Oracle Application Express Workshop II

Student Guide
D79657GC20
Edition 2.0 | August 2015 | D92340

Learn more from Oracle University at oracle.com/education/

Authors

Salome Clement
Dimpi Sarmah

Technical Contributors and Reviewers

Bryan Roberts
Wayne Abbott
Iloon Ellen
Marc Swetz
Patrick Wolf
Sathish K Kumar
Shakeeb Rahman
Nancy Greenberg
Ashley.chen
Klaus.Husermann
Marcie Young
Andrew Rothstein
Anthony Rayner
Chaitanya Koratamaddi
David Peake
Denise Simpson
Krishnanjani Chitta
Yi Lu
Lakshmi Narapareddi
Fabrizio Siracusa
Hanson Hutabarat
Hilary Farrell
Ashwin Agarwal
Anjani Pothula

Editor

Nikita Abraham

Graphic Designers

Seema Bopaiah
Kavya Bellur
Maheshwari Krishnamurthy

Publishers

Syed Ali
Giri Venugopal
Jayanthy Keshavamurthy
Pavithran Adka

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

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

Contents

1 Course Overview

- Objectives 1-2
- Road Map 1-3
- Course Environment 1-6
- Workspace Details 1-7
- Accessing the labs Directory 1-8
- Introducing Course Persona: Jack 1-9
- Demo Application: Project Tracking System 1-10
- Practice Application: GlobalMart Management Tool 1-11

2 Introduction and Review

- Jack Reviews APEX Concepts and PTS Features 2-2
- Objectives 2-3
- What Is Oracle Application Express? 2-4
- Building Your Database Application 2-5
- Building and Accessing Database Objects 2-7
- Components of a Database Application 2-8
- Page Designer: The World's Most Advanced IDE 2-9
- Page Designer: Drag-and-drop from Gallery 2-10
- Securing Database Applications 2-11
- Components of Oracle Application Express: Team Development 2-12
- What Is a Packaged Application? 2-13
- Application Express Workspace Administration 2-15
- Oracle Application Express: Other Features 2-16
- Additional Resources: Application Express OTN Page 2-17
- Oracle Learning Library 2-18
- Additional Resources: Documentation and Tutorials 2-19
- Oracle Application Express Developer Certified Expert 2-20
- Summary 2-21
- Practice 2 Overview: Introduction and Review 2-22

Unit I: Enhancing Application Pages

- Project Tracking System: Scenario I-2
- Course Road Map I-3

- 3 Using Oracle APEX Collections**
 - Using Collections in PTS 3-2
 - Objectives 3-3
 - What Is a Collection? 3-4
 - Common Use Cases 3-6
 - APEX_COLLECTION API: Overview 3-8
 - Collection Structure and Data Types 3-9
 - Creating a Collection 3-10
 - Creating a Collection: Syntax and Example 3-12
 - Viewing the Collection in Session State 3-13
 - Adding Members to a Collection 3-14
 - Adding Members: Syntax and Example 3-15
 - Accessing a Collection 3-16
 - Updating Members of a Collection 3-17
 - Updating Members: Syntax and Collection 3-18
 - Updating Member Attributes of a Collection 3-19
 - Updating Member Attributes: Syntax and Collection 3-20
 - Other Useful Methods 3-21
 - Additional Resource: Collections Packaged Application 3-22
 - Summary 3-23
 - Practice 3 Overview: Using Oracle APEX Collection 3-24

- 4 Creating Dynamic Actions**
 - Creating Dynamic Actions in PTS 4-2
 - Objectives 4-3
 - Review: What is a Dynamic Action? 4-4
 - Review: Basic Dynamic Actions 4-5
 - Components of Dynamic Actions 4-6
 - Creating a Dynamic Action 4-7
 - Dynamic Actions: Events 4-8
 - Dynamic Actions: Actions 4-10
 - Deleting a Row in a Report 4-11
 - Processing a Modal Window 4-13
 - Creating Custom Events 4-14
 - Additional Resource: Sample Packaged Application 4-15
 - Summary 4-16
 - Practice 4 Overview: Using Dynamic Actions 4-17

5 Using Plug-ins in an Application

- Using Plug-ins in PTS 5-2
- Objectives 5-3
- What Is a Plug-In? 5-4
- Types of Plug-ins 5-5
- Accessing the Plug-In Repository 5-6
- How to Use a Plug-In in an Application 5-7
- Importing a Plug-In 5-8
- Using an Item Plug-in 5-9
- Using a Region Plug-In 5-10
- Using a Process Plug-in 5-11
- Using a Dynamic Action Plug-In 5-12
- Creating a Plug-In 5-13
- Optimizing Performance of a Plug-In 5-14
- Summary 5-15
- Practice 5 Overview: Using Plug-Ins in an Application 5-16

6 Incorporating Interactivity Using JavaScript and jQuery

- Using JavaScript, AJAX, and jQuery in PTS 6-2
- Objectives 6-3
- JavaScript, AJAX, jQuery: Overview 6-4
- Using JavaScript: A Simple Example 6-5
- Using JavaScript Functionalities in APEX 6-6
- JavaScript APIs 6-7
- Specifying Static ID for Regions 6-8
- Using jQuery 6-9
- Introducing qTip jQuery Plug-in 6-10
- Using jQuery: qTip2 plug-in Tooltip Example 6-11
- Using AJAX 6-12
- AJAX and APEX 6-13
- When to Use 6-14
- Summary 6-15
- Practice 6 Overview: Incorporating Interactivity Using JavaScript and jQuery 6-16

Unit II: Adding Advanced APEX Functionality to an Application

- Project Tracking System: Scenario II-2
- Course Road Map II-3

7 Generating and Using Table APIs

- Generating and Using Methods on Tables in PTS 7-2
- Objectives 7-3
- What Are Methods on Tables? 7-4
- Steps to Generate Methods on Tables 7-5
- Add Additional Business Rules to the API 7-6
- Creating a Form that Uses the API 7-7
- Raising Errors From the API 7-9
- Case Scenario 2 7-10
- Case Scenario 3 7-11
- Quiz 7-12
- Practice 7 Overview: Generating and Using Table APIs 7-13
- Summary 7-14

8 Creating and Using RESTful Web Services

- Using RESTful Services in PTS 8-2
- Objectives 8-3
- What Is a Web Service? 8-4
- What Are RESTful Web Services? 8-5
- Advantages of RESTful Web Services 8-6
- RESTful Web Service Components 8-7
- ORDS and RESTful Web Services 8-9
- Accessing RESTful Services 8-10
- Creating a RESTful Web Service 8-11
- Testing the RESTful Web Service 8-13
- Adding a Bind Variable to the RESTful Web Service 8-14
- RESTful Web Service: Examples 8-15
- RESTful Web Service: Example Simple Query 8-16
- RESTful Web Service: Example Simple Query JSON Results 8-17
- RESTful Web Service: Example Query with a Parameter 8-18
- RESTful Web Service: Example Query with Parameter Results 8-19
- RESTful Web Service: Example PL/SQL 8-20
- RESTful Web Service: Example Feed 8-21
- RESTful Web Service: Example Feed Results 8-22
- Accessing the Web Service Referencing Page 8-23
- Steps to Create and Consume a RESTful Web Service 8-24
- Step 1: Creating a Database Application 8-25
- Step 2: Creating a Web Service Reference 8-26
- Step 3: Creating a Form and Report on the Web Service 8-29
- Step 4: Testing Your Application 8-32
- Consuming Your APEX Web Service Using Java 8-33

Scenario 1: E-Business Suite	8-34
Scenario 2: Accessing the PTS Data	8-35
Practice 8 Overview: Creating and Using RESTful Web Services	8-36
Summary	8-37

9 Using Templates and Themes

Using Templates and Themes in PTS	9-2
Objectives	9-3
Types of Applications	9-4
User Interface	9-6
Desktop Versus Mobile Applications Home Page	9-7
Desktop Versus Mobile Applications Reports	9-8
Desktop Versus Mobile Applications Forms	9-9
Application Options	9-10
Types of Themes	9-11
APEX-supplied Themes and Templates	9-12
What Are Templates?	9-13
Types of Templates	9-14
Using Substitution Strings in Templates	9-15
Page Template: <head>	9-16
Page Source: <head>	9-17
Application Header	9-18
Page Template: <header>	9-19
Page Source: <header>	9-20
Page Template: Body	9-21
Page Template: Position	9-22
Page Template	9-23
Using Button Templates	9-24
Copying and Modifying a Template	9-25
Inspecting Browser Elements	9-26
Copying a Template	9-27
Modifying a Template	9-28
Associating Template with Button	9-29
Quiz	9-30
Creating a Custom Theme	9-31
Exporting Your Theme	9-32
Copying Your Theme	9-33
Editing Your Theme	9-34
Managing Workspace Themes	9-35
Creating a Master Application	9-37
Adding a Custom Theme to the Master Application	9-38

Switching the Current Theme	9-39
Deleting the Default Theme	9-40
Copying an Existing Application From the Master Application	9-41
Quiz	9-42
Practice 11 Overview: Using Templates and Themes	9-43
Summary	9-44

10 Developing a New Theme for Your Application Using Theme Roller

Theme Roller in PTS	10-2
Objectives	10-3
What is Theme Roller?	10-4
Components of Theme Roller	10-5
Using Theme Roller in PTS	10-7
Practice 10 Overview: Developing a New Theme for Your Application Using Theme Roller	10-10
Summary	10-11

Unit III: Making an Application Production-Ready

Jack Wants to Deploy the PTS Application	III-2
Course Road Map	III-3

11 Securing an Application

Securing PTS	11-2
Objectives	11-3
Common Hacking Mechanisms	11-4
SQL Injection	11-5
SQL Injection: Example	11-6
Correcting SQL Injection: Using Bind Variables	11-8
Assessing Vulnerability	11-10
Best Practices Against SQL Injection	11-11
Cross-Site Scripting	11-14
Cross-Site Scripting: Example	11-15
Specifying Browser Security	11-18
What Is a Directory?	11-19
Oracle Internet Directory	11-20
Authenticating Users by Using LDAP	11-23
Creating an LDAP Authentication Scheme in Oracle APEX	11-24
Running the Application and Logging In as a User in Oracle Internet Directory	11-26
Summary	11-27
Practice 11-1 Overview: Securing an Application	11-28

12 Deploying an Application

- PTS Scenario 12-2
- Objectives 12-3
- Steps to Deploy an Application 12-4
- What Is a Packaged Application? 12-5
- What Are Supporting Objects? 12-6
- Identifying the Supporting Objects for an Application 12-7
- Creating Installation Scripts 12-8
- Specifying Prerequisites and Other Options 12-9
- Specifying Build Options 12-10
- Creating an Installation Script 12-11
- Creating Upgrade Scripts 12-12
- Creating Deinstallation Scripts 12-13
- Accessing the Export Page 12-14
- Exporting an Application 12-15
- Importing an Application 12-17
- Installing the Application 12-18
- Publishing the Application URL 12-19
- Summary 12-21
- Practice 12 Overview: Deploying and Maintaining Your Application 12-22

13 Optimizing Your APEX Application

- Optimizing the PTS Application 13-2
- Objectives 13-3
- Application Performance Considerations 13-4
- Managing Services 13-5
- Monitoring Activity 13-6
- Monitoring Activity: Page Views Reports 13-8
- Monitoring Activity: Page View Analysis Reports 13-9
- Database Configuration Considerations 13-10
- Monitoring Application Performance 13-11
- Monitoring Application Performance: #TIMING# Substitution String 13-12
- Monitoring Application Performance: Object Reports 13-13
- Monitoring Application Performance: Caching 13-14
- Monitoring Application Performance: Tracing Your Session 13-16
- Monitoring Application Performance: Database Monitor 13-17

Monitoring Application Performance: Locks	13-19
Monitoring Application Performance: Tuning SQL	13-20
Monitoring Application Performance: Tuning Page Elements	13-21
Practice 13 Overview: Managing and Monitoring Your Application	13-22
Summary	13-23

14 Globalization And Translation

Translating the PTS Application	14-2
Objectives	14-3
Accessing the Globalization Attributes Page	14-4
Editing the Globalization Attributes Page	14-5
Translating an Application and Globalization Support	14-8
Step 1: Mapping the Target Language	14-9
Step 2: Seeding and Downloading to a Translation File	14-10
Step 3: Translating the XLIFF File	14-11
Step 4: Applying XLIFF Translation Files	14-12
Step 5: Publishing the Application	14-13
Specifying the Primary Language for an Application	14-14
Translating Messages Used in PL/SQL Procedures	14-15
Practice 14 Overview: Applying Globalization and Translation	14-16
Summary	14-17

Appendix A: Quick Reference: Additional How-To Guide

Objectives	A-2
Logging In to Oracle Application Express Administration	A-3
Creating a Workspace and Workspace Administrator	A-4
Logging In to a Workspace	A-5
Installing a Packaged Application	A-6
Creating Database Objects	A-7
Running SQL Commands	A-8
Importing and Running a SQL Script	A-9
Accessing the Create Application Wizard	A-10
Creating a Database Application	A-11
Creating a Desktop Database Application	A-12
Page Definition: Overview	A-13
Different Views of a Page	A-14
Switching Between Pages and View Types	A-15
Creating a Report	A-16
Editing Report Attributes	A-17
Types of Reports Supported for Mobile Interface	A-18
Types of Forms	A-19

Creating a Form on a Table with Report	A-21
Linking a Report to a Form	A-22
Creating a Tabular Form	A-23
Adding a Region from the Rendering Tree	A-24
Adding a Region from the Gallery	A-25
Page Items: Examples	A-26
Creating a Page Item	A-27
Types of Page Items	A-28
Editing an Item	A-30
Creating a Button	A-31
Editing Button Attributes	A-32
Creating a Branch	A-33
Creating a Branch: Example	A-34
Creating Navigation Menu Entries	A-35
Creating a Static List	A-36
Creating List Entries	A-37
Creating a Dynamic List	A-38
Creating a List Region	A-39
Summary	A-40

Appendix B: Oracle Application Express: Other Features

Objectives	B-2
Agenda	B-3
Remote PL/SQL Debugging with SQL Developer	B-4
Importing Your Packaged Application	B-5
Setting DEBUG DBA Privileges	B-6
Compiling the PL/SQL Package and Package Body for Debug	B-7
Setting the Remote Debug	B-8
Adding the DEBUG Code to Your Process	B-9
Agenda	B-11
Synchronizing Beta and Development Feedback	B-12
Reviewing the Feedback Synchronization Source Identifier	B-13
Synchronizing Beta and Development Feedback: Example	B-14
Exporting the Feedback from Beta System	B-15
Importing the Feedback into Development System	B-16
Responding to the Feedback	B-17
Exporting the Feedback Response from Development System	B-18
Importing the Feedback Response into Deployment System	B-19
Creating a Feedback Review Report	B-20
Summary	B-21

Appendix C: Extending Your Application with User-Defined Error Handling

Objectives	C-2
Error Handling Function	C-3
1. Creating Constraint Lookup Table	C-4
2. Creating Error Handling Function	C-5
2. Create Error Handling Function	C-6
Setting Up an Example to Test Error Handling	C-7
Associating the Error Handling Function with Your Page	C-8
Testing the Page for Error Handling	C-9
Summary	C-10

Appendix D: Migrating a Desktop Application to a Responsive Theme

Objectives	D-2
What Is Responsive Web Design?	D-3
Responsive Design Versus Mobile	D-7
Responsive Design in Oracle APEX	D-9
Theme 25: CSS3 Media Queries	D-10
Grid-Based Layout	D-11
Migrating an Application to Theme 25	D-13
Creating Theme 25	D-14
Switching to Theme 25: Template Conversion Compatibility	D-15
Page Template Changes	D-16
Revising Report Regions: Classic Report	D-17
Revising Report Regions: Interactive Report	D-18
Theme 25: Icons	D-19
Revising Buttons	D-20
Theme 25: Responsive Classes	D-21
Theme 25: High Resolution Display Support	D-22
Explanation of Grid Layout Definition	D-23
Revising Forms: Region Template	D-24
Revising Forms: Regions	D-25
Revising Forms: Page Items	D-27
Helpful Hints	D-28
Quiz	D-29
Summary	D-30

Appendix E: Making Your Application Accessible

Objectives	E-2
What Is Web Accessibility?	E-3
Why Is Web Accessibility Important?	E-4
Screen Readers	E-5

Turn Screen Reader Mode On	E-6
Themes and Templates: Page Template	E-7
Changing Text of Toggle Message	E-8
Themes and Templates: Region Template	E-9
Themes and Templates	E-10
Interactive Reports	E-11
Forms	E-13
Tabular Forms	E-15
Charts	E-16
High Contrast	E-17
Quiz	E-19
Summary	E-20

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

1

Course Overview

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this course, you should be able to:

- Enhance application pages using collections, dynamic actions, plug-ins, and JavaScript
- Extend application capabilities using RESTful web services, Table APIs, and custom themes and templates
- Secure, package, and deploy an application
- Optimize and maintain applications



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Application Express is a rapid application web development tool for the Oracle database.

This course is an advanced workshop designed to build on what you have learnt during the five-day *Oracle Application Express: Workshop I* course. This course is geared toward users who have some working knowledge of the tool and want to learn more advanced topics and techniques.

Road Map

Lesson 1: Course Introduction

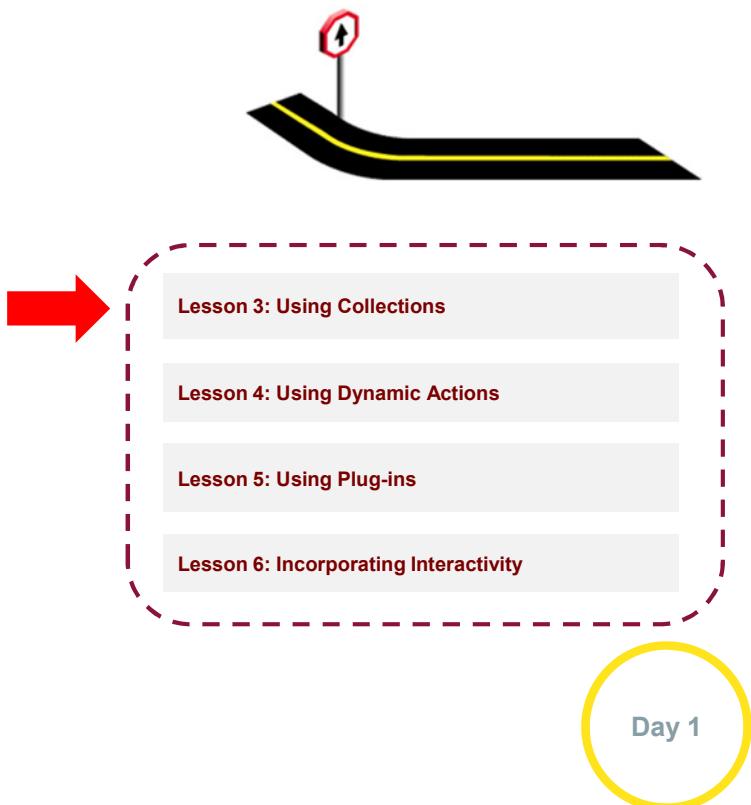
Lesson 2: Oracle APEX Overview

Unit 1: Enhancing Application Pages

Unit 2: Adding Advanced APEX Functionality to an Application

Unit 3: Making an Application Production-ready

ORACLE®



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In Unit I, you learn APEX features and functionalities that will help you enhance applications pages.

Road Map

Lesson 1: Course Introduction

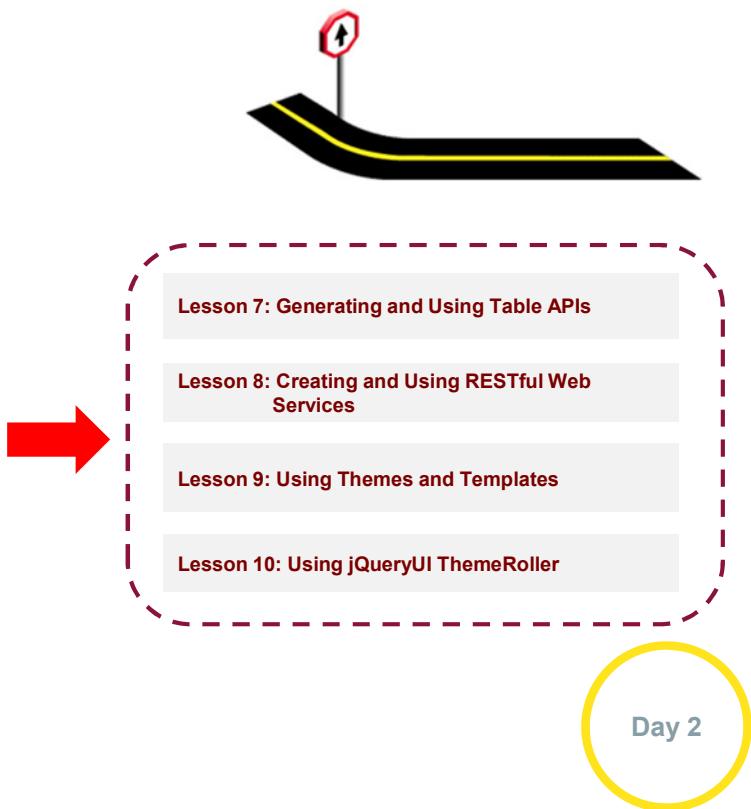
Lesson 2: APEX Overview

Unit 1: Enhancing Application Pages

Unit 2: Adding Advanced APEX
Functionality to an Application

Unit 3: Making an Application
Production-ready

ORACLE®



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In Unit II, you learn APEX features and functionalities that will help you extend an application's capabilities.

Road Map

Lesson 1: Course Introduction

Lesson 2: APEX Overview

Unit 1: Enhancing Application Pages

Unit 2: Adding Advanced APEX
Functionality to an Application

Unit 3: Making an Application
Production-ready



Lesson 11: Securing an Application

Lesson 12: Deploying and Maintaining an
Application

Lesson 13: Optimizing an Application

Lesson 14: Globalization and Translation



Day 3

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ORACLE®

In Unit III, you learn to secure, package, deploy, optimize, and maintain an application.

Course Environment



Operating system

- Linux x64

Installed products

- Oracle Database 12c R1
- Oracle Application Express 5.0
- Oracle REST Data Services
- Java Platform (JDK)
- Internet Browser (Firefox)



ORACLE®

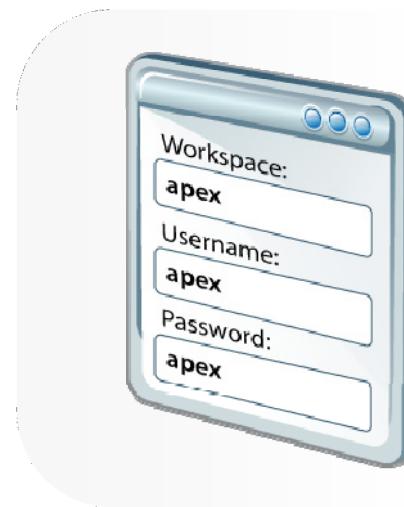
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The classroom setup provides each student with a separate instance of the Oracle Database. The other products installed are listed in the slide.

You use a web browser to access the Oracle APEX workspace which has already been created and installed for you.

Workspace Details

- An Oracle Application Express workspace is assigned to you.
 - Workspace name: apex
 - Username: apex
 - Password: apex
- Log in to your workspace to complete the practice tasks in the Activity Guide.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

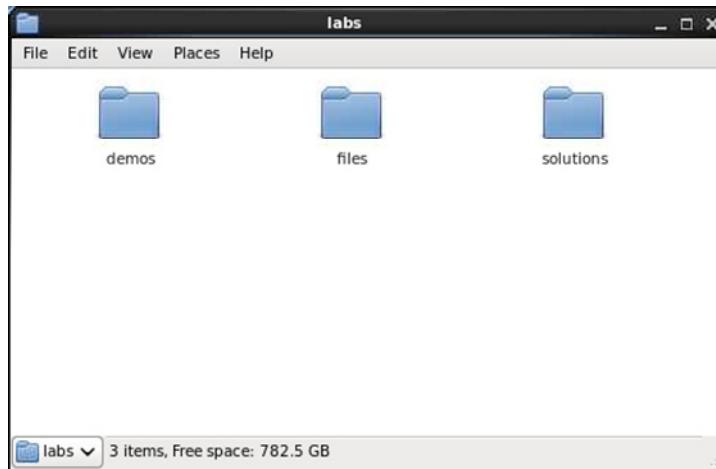
You can complete all the practices designed for this course using the workspace created in the APEX database on your machine. The workspace name, username, and password details are listed in the slide.

To access the Oracle Application Express development instance, open a web browser and enter the following URL in the address bar:

`http://<hostname>:8080/apex`

Accessing the `labs` Directory

/home/oracle/labs



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

All the files that are required to complete the practices are available in the `labs` directory. To access the `labs` directory, from the Applications menu, select System Tools > File Browser. From the `oracle` directory, open the `labs` directory. You will see three folders and their contents:

- **demos:** Files used by the instructor for demonstrations.
- **files:** Files you need to complete the practices. You can use this location to save files while performing the practices, if required.
- **solutions:** The solution scripts for the practices given in the Activity Guide. These files can also be used as catch-up applications that you can import in case you were not able to complete a practice.

Introducing Course Persona: Jack

Persona 1 Name **Jack** ● Designation **Application Developer**



Jack is an application developer in a start-up software company named 'XWHYZEE Technologies' and he reports to Jill.

Recently, Jack used Oracle's rapid application development tool 'Application Express 5.0' (APEX 5.0) to develop an application called 'Project Tracking System' (PTS). PTS enables Jill to track and manage multiple projects and team members seamlessly. Now that Jack has some experience creating a complete end-to-end solution using APEX, he decides to upgrade the application functionalities using some of the advanced APEX features.

Career Profile: Jack is an application developer and has good experience in using SQL and PL/SQL while working with database applications. Recently, he successfully created an application using Oracle Application Express.

Training Requirement: In order to upgrade PTS using Oracle APEX 5.0, Jack should be able to:

- ✓ Implement advanced APEX functionalities such as collections, dynamic actions, plug-ins, AJAX, jQuery, and JavaScript to application pages
- ✓ Integrate the web services functionality
- ✓ Create customized themes and templates
- ✓ Secure the application
- ✓ Package and deploy the application

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

All the topics in this course are based on scenarios related to this persona.

Jack represents the majority of the target audience for this course. He is an experienced Oracle APEX developer and is looking to sharpen and enhance his Oracle APEX skills.

Note: Throughout the course, references or examples explained using the PTS application are highlighted in a gray box as used on this page.

Demo Application: Project Tracking System

Project Tracking System (PTS) is a web application developed as part of APEX 5.0 Workshop I training.



New Functionality to be Built

- Team members should be able to create and send their weekly status reports to their reporting managers.
- Implement a rich user interface experience using advanced APEX features like AJAX, jQuery, JavaScript, dynamic actions, and plug-ins.
- Customize the look and feel using themes and templates.
- Secure the application against common threats.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The demonstrations for this course will build upon the PTS application, which was built during the demonstrations in the *Oracle Application Express: Workshop I* course.

Some of the new functionalities that will be built in this course are listed in the slide.

Practice Application: GlobalMart Management Tool

GlobalMart Management Tool (GMT) is a web application developed as part of APEX 5.0 Workshop I training.



New Functionality to be Built

- Add to cart functionality
- Implement a rich user interface experience using advanced APEX features like AJAX, jQuery, JavaScript, dynamic actions, and plug-ins.
- Customize the look and feel using themes and templates.
- Secure the application against common threats.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The practices for this course will build upon the GMT application, which was built during the practices in the *Oracle Application Express: Workshop I* course.

Some of the new functionalities that you will build during the practices sessions of this course are listed in the slide.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

2

Introduction and Review

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack Reviews APEX Concepts and PTS Features



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

It has been a couple of months since Jack released PTS as a production application to Jill and the other team members. Now that Jack is about to work on upgrading the application, he spends some time reviewing the features of APEX and the functionalities built into the PTS application.

Objectives

After completing this lesson, you should be able to:

- List the various Oracle Application Express (APEX) components and their features
- Examine additional APEX resources



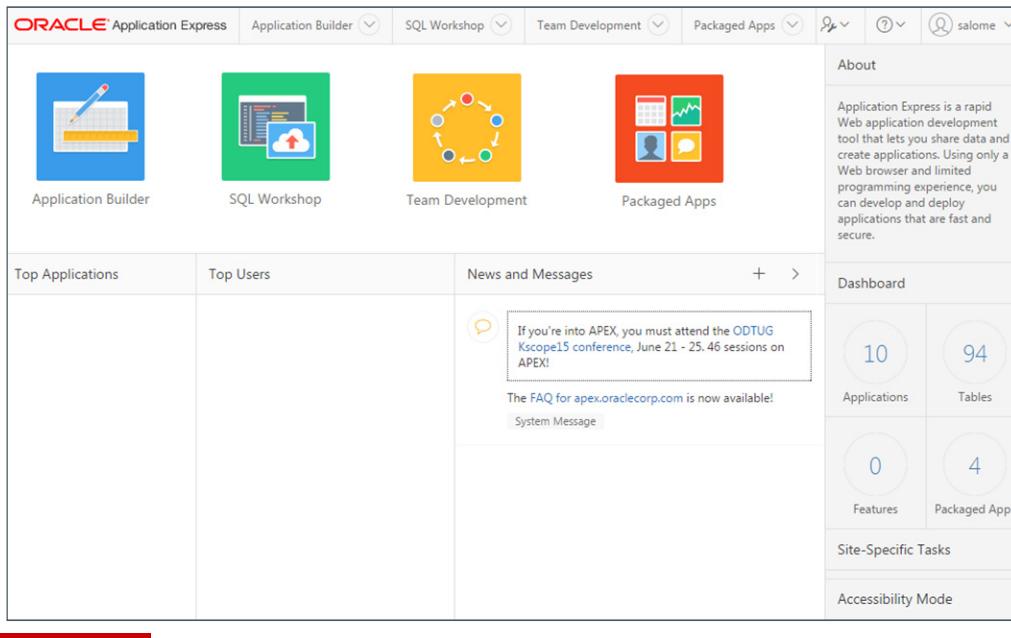
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you review the components of Oracle Application Express and what tasks are required while building, securing, and deploying an application. You also examine additional resources that are available to learn more about Application Express.

What Is Oracle Application Express?

Oracle Application Express is a rapid web application development tool for Oracle database.



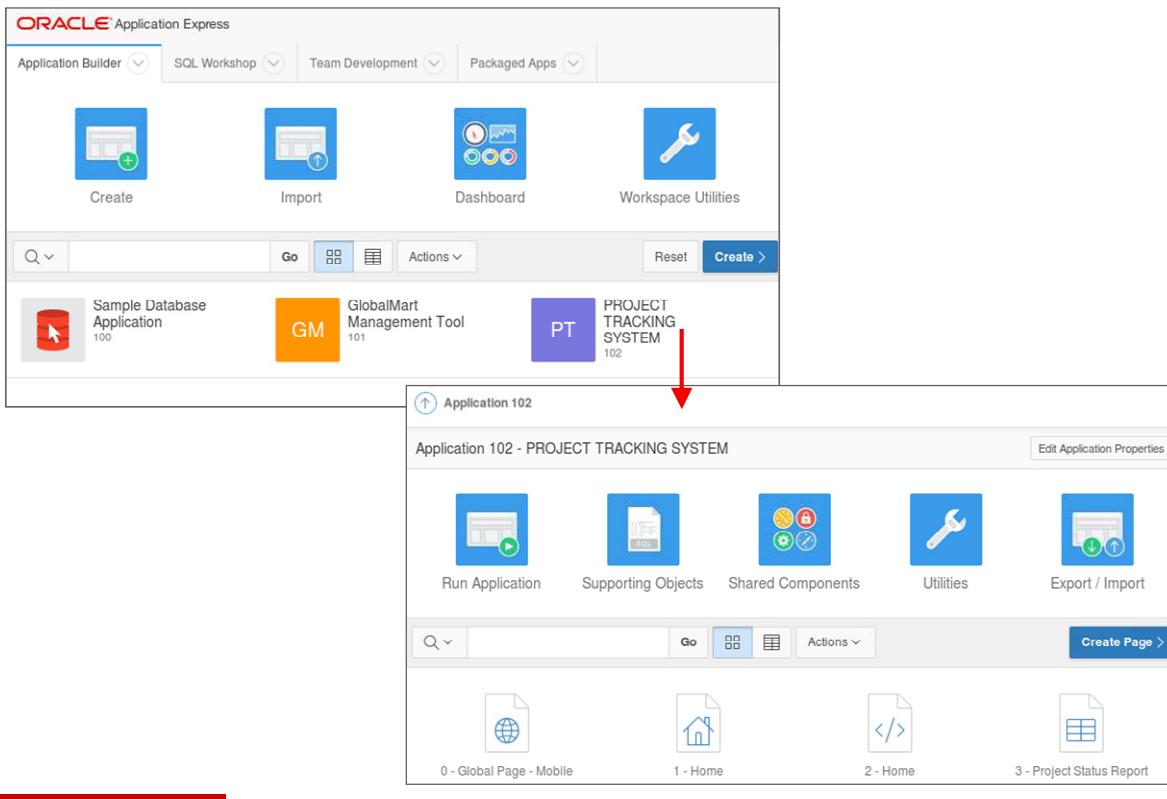
Oracle Application Express is a web-based development and deployment tool available with Oracle database. It helps you create database-centric web applications that are reliable, scalable, and secure. It has several built-in features and wizards that quicken your development process.

The tool has a graphical, user-friendly interface. To use Oracle Application Express, you require minimal programming knowledge.

The Oracle Application Express development environment consists of four components:

- **Application Builder:** Used to build dynamically rendered applications. You assemble an HTML interface (or an application) to interact with database objects, such as tables and procedures.
- **SQL Workshop:** Used to create, manage, and view database objects, and run SQL statements and scripts
- **Team Development:** Used to track new features, non-feature-related tasks (or To Do tasks), bugs, and milestones
- **Packaged Apps:** Used to install productivity applications supplied by the Oracle APEX development team

Building Your Database Application



When you click the application icon or name, the Application home page appears. You can see the application ID and name at the top of the page.

From the Application home page, you can:

- **Run the application:** Click the Run Application icon to submit the pages to the Oracle Application Express engine to render a viewable HTML page.
- **Define supporting objects:** Click Supporting Objects to access the utility to define the database object definitions, images, and seed data to be included in your application export for your packaged application.
- **Create shared components:** Click Shared Components to build shared application components and user interface controls.
- **Access utilities:** Click the Utilities icon to see summarized information across the application and access useful tools, such as Application Dashboard, Export Repository, Upgrade Application, Database Object Dependencies, Attribute Dictionary, Advisor, Application Express Views, Recently Updated Pages, and Debug Messages.

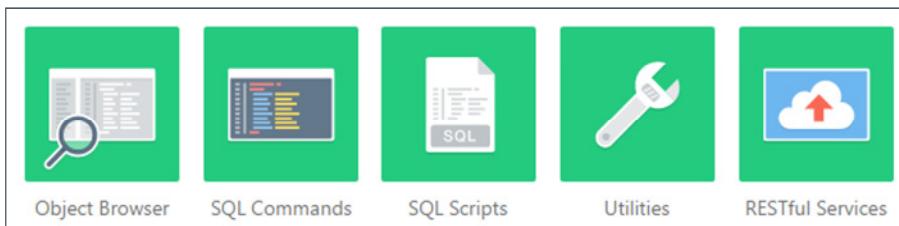
- **Export and import applications:** Click the Export/Import icon to export or import an entire application or components, such as cascading style sheets, images, static files, themes, and user interface defaults.

Click Edit Application Properties to make any changes to the definition, globalization, and security properties of the application. Click Create Page to add a page to your application.

Building and Accessing Database Objects

SQL Workshop enables you to view and manage database objects from a web browser. You can:

- Create and browse database objects
- Execute SQL commands
- Load and run scripts
- Specify RESTful services



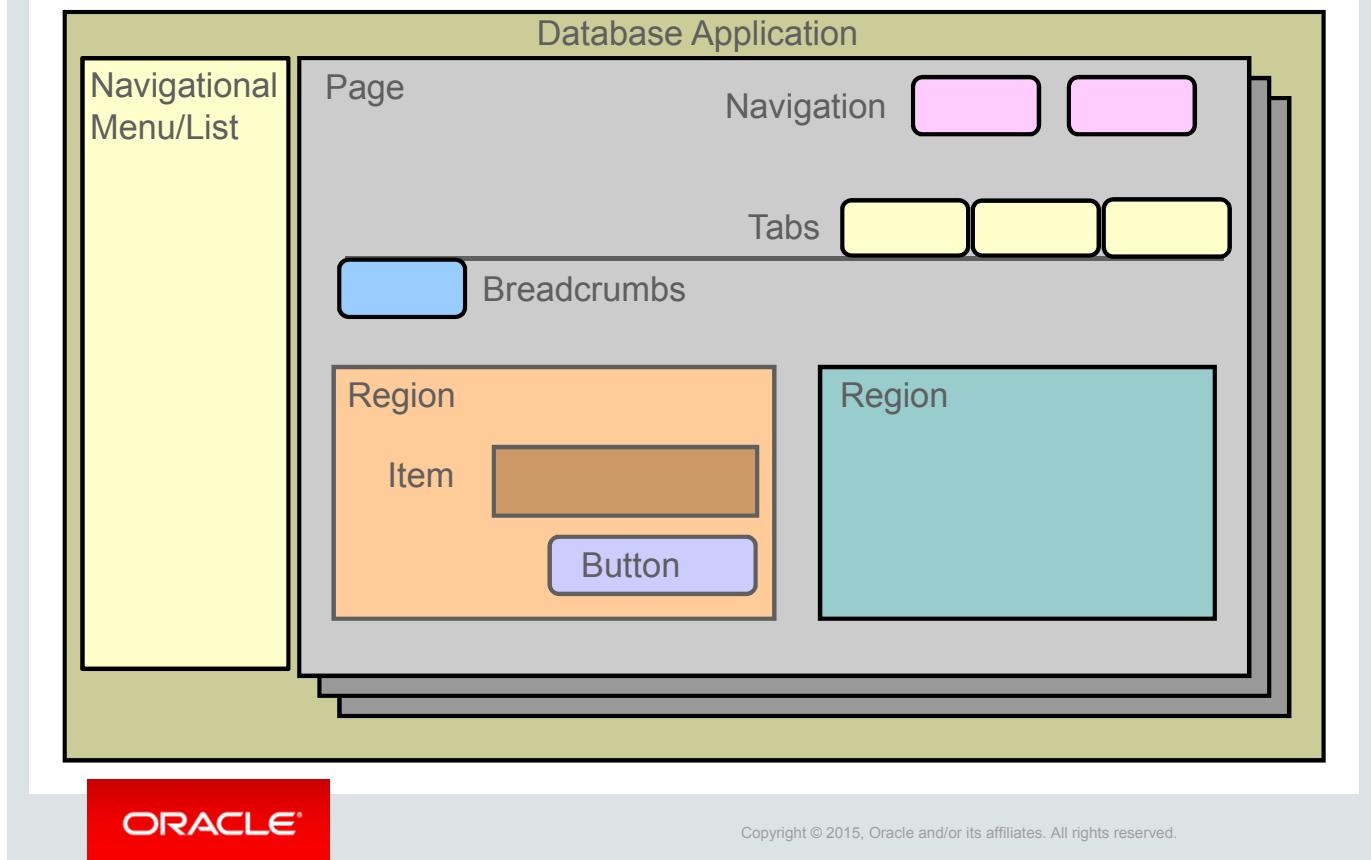
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

With SQL Workshop, you can view and manage database objects from a web browser. SQL Workshop is divided into five primary sections. They are:

- **Object Browser:** You use the Object Browser to view, insert, update, and delete data directly against tables.
- **SQL Commands:** You run SQL commands and anonymous PL/SQL, scripts, and saved queries by using SQL Command Processor.
- **SQL Scripts:** You use the SQL Script facility to create, edit, view, run, and delete script files. You can also upload and download scripts from your local file system.
- **Utilities:** You use Utilities to load and unload data from an Oracle database, generate DDL, view object reports, and restore dropped database objects.
- **RESTful Services:** You use RESTful services to declaratively specify RESTful services mapped to SQL and PL/SQL.

Components of a Database Application



A database application is a collection of database-driven web pages that are linked using navigational controls, such as navigational menus/lists, tabs, buttons, and hypertext links.

A page is the basic element of an application.

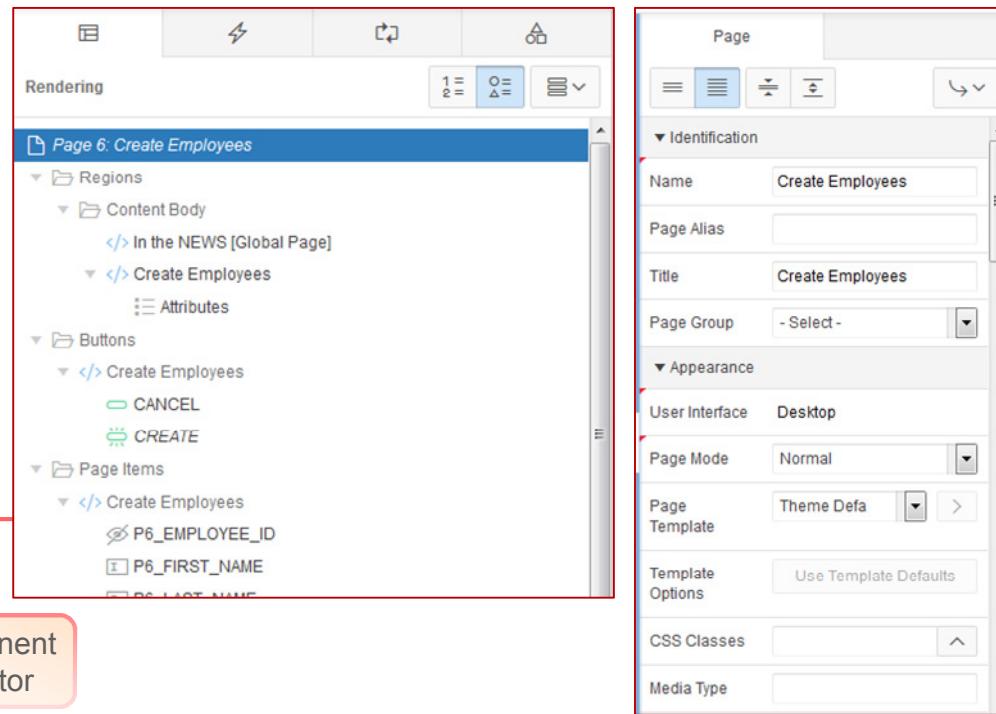
A page is divided into regions. A region is a section of a page that contains content. The content of a region is determined by the region source. For example, a region can contain a report based on a SQL query, or it can contain static HTML.

A region can also contain:

- Items such as a text field, text area, select list, and check box
- Buttons to direct users to a specific page or URL, and also to post and process information
- Breadcrumbs to provide hierarchical navigation

Navigation entries are placed outside regions to enable users to navigate between the pages of an application.

Page Designer: The World's Most Advanced IDE



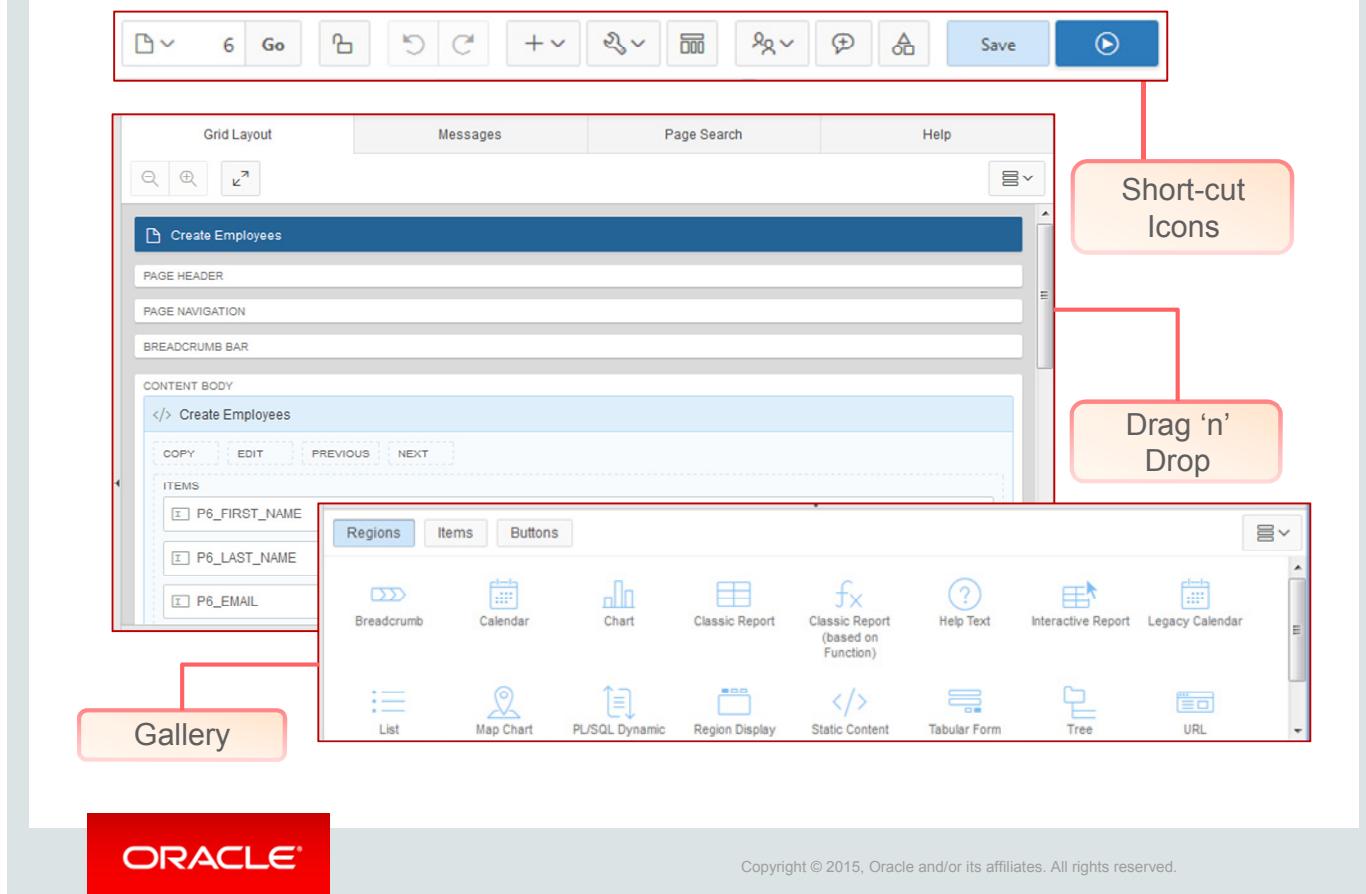
Next level of declarative programming → Design your page as you visualize.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Page items, regions, computations, processes, and validations are all organized under the 'Rendering' and 'Processing' tabs automatically. **Dynamic actions and shared components** are organized under separate tabs. Select a component in **Component Selector** and update its properties in **Property Editor**.

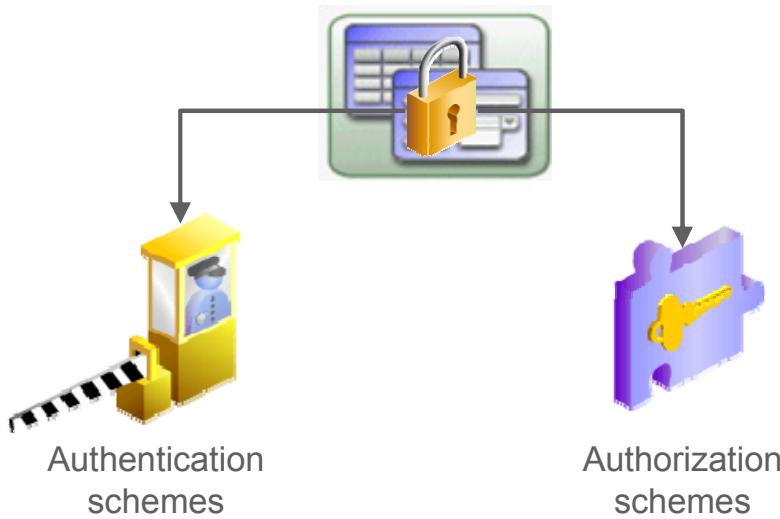
Page Designer: Drag-and-drop from Gallery



Page Designer provides many useful features for developers:

- You can **drag** regions, items, and buttons into **Grid Layout**. All the related **properties** will be updated automatically.
- Backward compatibility with **Component view**. You can easily view the page in component view just by clicking one icon.
- **Help tab** specifically for providing help on any attribute in the property editor of the selected component.
- **Shortcut icons** to create a new page, copy page, any type of region on the current page, shared component, bug, feedback, and so on.
- Shortcut icons to access debugging tools and other utilities.
- **Undo Redo** support
- **Errors and Warnings** are immediately visible.

Securing Database Applications



ORACLE®

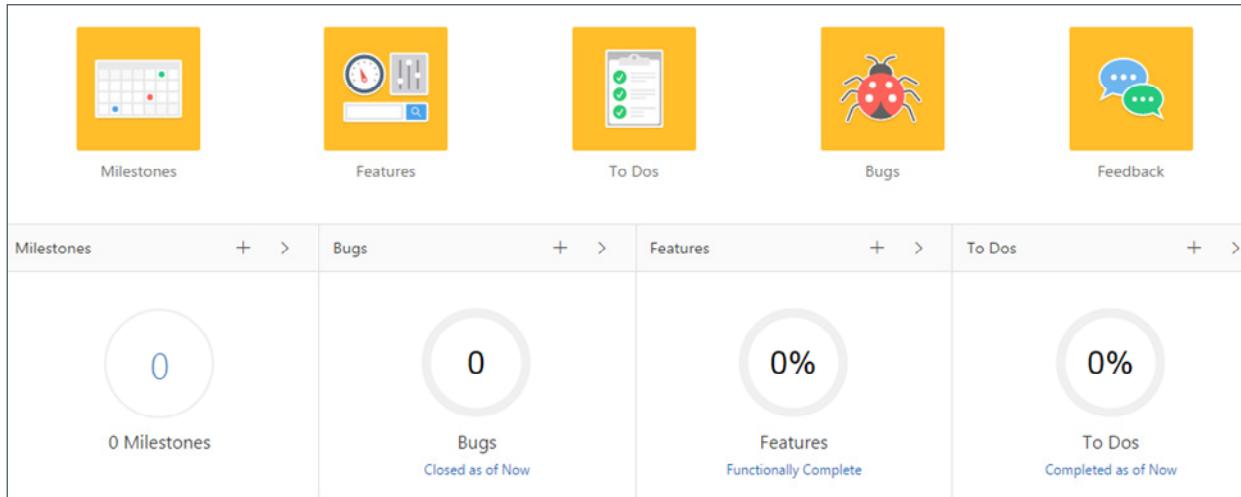
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In addition to developing applications, you secure your applications by using the following security mechanisms:

- **Authentication schemes:** An authentication scheme is a method that verifies a user's identity and informs the Oracle Application Express engine whether a login is successful or not. This login information is later used by the application. Your application gets the login information from a login page within an Oracle Application Express application, or from an external authentication page, such as Oracle Application Server Single Sign-On. You can change the authentication scheme at any given point in time. For example, you can use Lightweight Directory Access Protocol (LDAP) or Oracle Application Server Single Sign-On without changing the application logic.
- **Authorization schemes:** Authorization schemes (also known as reusable access control rules) can be centrally defined and reused by applying them to elements within an application. For example, using a single authorization scheme, you can control access to a single field, a button component, or the entire application.

Components of Oracle Application Express: Team Development

Team Development enables you to manage the application development process.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Team Development is a built-in project management tool that facilitates the management of the software development process. You can track features, to-dos, milestones, and bugs. Because Team Development is part of Application Express, you can add direct links between team development entries, and specific applications and pages. You can capture end-user feedback and convert it to features, to-dos, and bugs.

- **Milestones:** You use milestones to define the important dates within an application development release.
- **Features:** You use features to track functionality from initial conception through implementation. You can organize features by release, assigned to developers, tagged, and associated with milestones.
- **To-Dos:** To-do tasks are action items that can be assigned, prioritized, tagged, and tracked. To-do tasks can also be associated with features.
- **Bugs:** Bugs track software defects. Bugs can be assigned, associated with milestones, and tracked by due date, status, and other attributes.
- **Feedback:** Feedback provides the capability to solicit input from users of an application. Feedback streamlines the development process by structuring information from end users, thus making it useful to developers.

What Is a Packaged Application?

Packaged Applications are fully functional applications that you can view, use, and customize.

Single Export File

Application definitions

- Page rendering (regions, items, buttons, and so on)
- Page processing (computations, validations, and so on)

Supporting objects

- Database objects
- Seed data needed when application is installed
- Application image files
- CSS files
- Custom JavaScript files



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Packaged Applications are fully functional applications that you can view, use, and customize. Packaged Applications can include the following:

Application Definitions

An application definition includes any application or page component. This includes page-rendering components, such as regions, items, or button or page processing components, such as computations or validations.

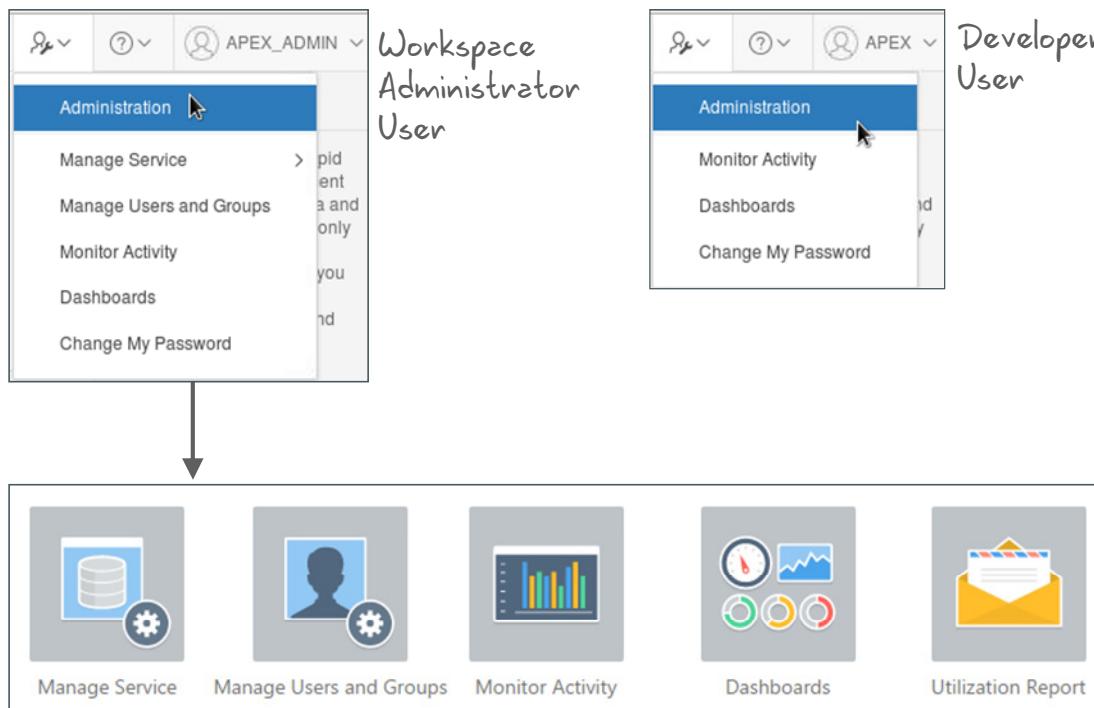
Supporting Objects

Supporting objects consist of the underlying objects and files necessary for the application. They typically include database objects, seed data needed when the application is installed, application image files, cascading style sheet (CSS) files, and custom JavaScript files.

Packaged Applications also include a collection of sample applications which demonstrate some of the major features of Oracle Application Express. You can find some packaged applications (both sample and productivity applications) by clicking Packaged Applications on the Application Builder home page. Both sample and productivity applications are fully functional applications that have been designed to address a specific business need. You can install, run, and use packaged applications as is, or analyze them to better understand how to use Application Builder to build specific types of functionality.

The main difference between a *sample* and *productivity* application is the level of support. By default, sample applications are fully editable. In contrast, you must unlock productivity applications before you can edit them. Unlocking a productivity application makes it ineligible for future upgrades or support by Oracle Support.

Application Express Workspace Administration



ORACLE®

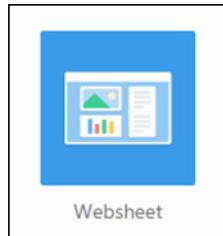
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Application Express users with workspace administration privileges can control their workspace. From the Administration home page, you can:

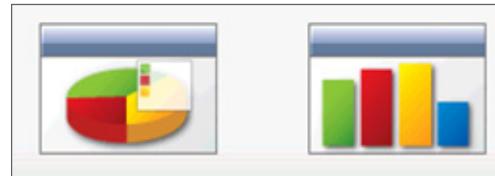
- **Manage Service:** Click Manage Service to manage session state, log files, service termination, schema requests, storage requests, cached content, and application models.
- **Manage Users and Groups:** Click Manage Users and Groups to control access for Application Express workspace administrators, application developers, and end users.
- **Monitor Activity:** Use the Monitor Activity page links to access reports and charts that allow you to monitor user and developer activity.
- **Dashboards:** Click Dashboards to view a summary of workspace attributes and statistics.
- **Utilization Report:** Click Utilization Report to summarize workspace activity and contents.

Oracle Application Express users with developer privileges have few options to monitor activity in the applications they are working on.

Oracle Application Express: Other Features



Websheets



Charts



Dynamic Actions



Plug-ins

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- **Websheet Applications:** Websheet applications enable users to build data-centric applications without any SQL programming knowledge. Websheet applications are simplified and support pages, data grids, and reports. When you create a Websheet application, Application Builder automatically handles the creation of tables, triggers, and sequences.
- **Charts:** Oracle Application Express provides built-in wizards for generating HTML and Flash charts. Oracle Application Express includes integration with AnyChart 5.1 charting engine and supports Gantt and Map chart types.
- **Dynamic Actions:** Dynamic Actions enable developers to define client-side behaviors declaratively, without the need for JavaScript or AJAX. Developers simply specify an element, when to execute, and what action to perform.
- **Plug-ins:** Plug-ins allow developers to declaratively extend the built-in types available with Application Express and enable developers to share and reuse them. Application Express supports a set of item, region, dynamic action, and process types. Plug-ins offer a means of augmenting these built-in types by declaratively creating and using new types in your application.

Additional Resources: Application Express OTN Page

The Oracle Application Express Product Page on OTN is a very useful place to gather information. It contains useful links such as:

- **Overview:** Get an overview of APEX and links to news, events, presentations, and books.
- **Downloads:** You can download the latest software.
- **Documentation:** Get access to the documentation and a whole host of how-to tutorials.
- **Community:** Get access to the APEX OTN forum, Community How-tos, and a list of blogs. Application Express OTN forum is one of the most popular forums on OTN. The forum has a knowledge base of hints, tips, and issues that users have encountered, and their resolutions. You can also access My Oracle Support to get additional support for Oracle products.
- **Learn More:** You can find a high-level overview of the major components within Application Express.

Oracle Learning Library

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Learning Library is an application built by using Oracle Application Express. Using the Learning Library, you can search for free online training content (OBEs, Demonstrations, and Tutorials) on OTN.

You can search various contents in the Learning Library. You can enter the content of your interest in the search text bar. You can also select the tab for the product area that you are interested in. You will see a default view of the content for that product area. Then you can further narrow your search by using the criteria. When you have found the content that you are looking for, select the title to view the content.

Additional Resources: Documentation and Tutorials

The screenshot shows a web page for Oracle Application Express Documentation. At the top, there is a navigation bar with tabs: Overview, Downloads, Documentation (which is selected and highlighted in blue), Community, and Learn More. Below the navigation bar is a logo for Oracle Application Express Documentation, featuring a stylized pencil icon. The main content area contains a paragraph of text about the history and features of Application Express. Below the text is a list of links to various documentation resources, each preceded by a small circular icon with a downward arrow:

- ❯ Full Library
- ❯ Release Notes
- ❯ Installation Guide
- ❯ Application Builder User's Guide
- ❯ Migration Guide
- ❯ SQL Workshop Guide
- ❯ API Reference
- ❯ Administration Guide
- ❯ End User's Guide
- ❯ Third Party Books
- ❯ Documentation Archives

At the bottom of the page, there is a red footer bar with the "ORACLE" logo on the left and a copyright notice on the right: "Copyright © 2015, Oracle and/or its affiliates. All rights reserved."

OTN has documentation to help answer your questions. You can access the “Documentation” page from OTN at the following URL:

<http://www.oracle.com/technetwork/developer-tools/apex/documentation/index.html>

Oracle Application Express Developer Certified Expert

The screenshot shows a web page from the Oracle Education website. At the top, there are navigation links: Home > Training > Database > Database Application Development > Oracle Application Express (Oracle APEX). On the right, there are contact details: Local: 1800 103 4775 Intl: +91 80 67863102 and a Live Chat link. Below the navigation, there are three tabs: Training Courses (selected), Certification (highlighted in red), and Learning Paths. The main content area is titled "Oracle Application Express (Oracle APEX) Certification" and contains the text: "Click on the boxes below to learn detailed requirements for achieving each certification." Below this, there is a diagram showing a flow from "Certified Expert" to "Oracle Application Express Developer Certified Expert". A callout box provides the definition: "The Oracle Certified Expert certification program grants credentials that recognize competency in specific technologies, architectures or domains not currently covered in the path-based Oracle Certifications." An arrow points down to a section titled "Step 1 - Pass Exam". This section includes a "Exam" box listing "Oracle Application Express (APEX) 4: Developing Web Applications 1Z0-450" and a "Register Now" button. To the right, there is an "Exam Preparation (optional)" box containing a single item: "Oracle Application Express: Developing Web Applications". The Oracle logo is at the bottom left, and copyright information is at the bottom right: "Copyright © 2015, Oracle and/or its affiliates. All rights reserved."

You can access information about *Oracle Application Express Developer Certified Expert* from the Certification Path > Database > Database Application Development section at the following location:

<http://education.oracle.com>

On the Oracle Application Express Developer Certified Expert page, click the links in the diagrams to view the *Oracle Application Express 4: Developing Web Applications* exam requirements in detail.

- **Exam number:** 1Z0-450

The recommended training and preparation for the Oracle Application Express Developer Certified Expert exam is the *Oracle Application Express: Developing Web Applications* course.

Summary

In this lesson, you should have learned how to:

- List the various Oracle APEX components and their particular functions
- Examine additional APEX resources



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you reviewed the basic concepts and components of Oracle Application Express. In addition, you should have examined the additional resources available to learn Oracle Application Express.

Jack is now all set to upgrade the PTS application and learn the new and advanced features of Oracle APEX.

Practice 2 Overview: Introduction and Review

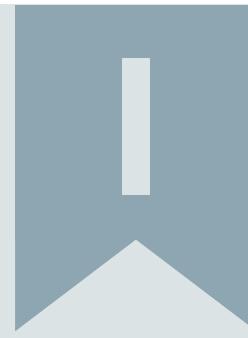
This workshop covers the following topics:

- Accessing the lab environment
- Reviewing the demonstration application: Project Tracking System
- Reviewing the practice application: GlobalMart Management Tool



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this practice, you access the lab environment and review the applications available in the APEX workspace.



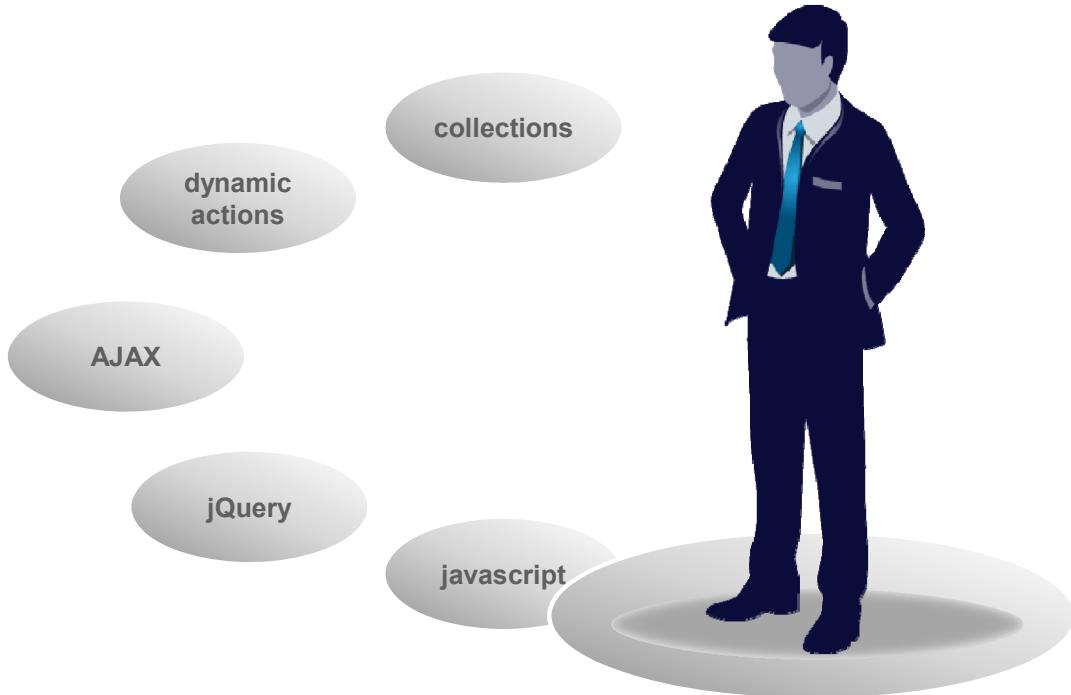
Enhancing Application Pages

Unit I

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Project Tracking System: Scenario



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack has gathered some feedback and new functionality requests from his team and Jill, all of whom are using PTS. He decides to implement the functionalities one by one. He starts to work on building the following requests. Team members should be able to submit a weekly status report to Jill using PTS itself. He wants to build this feature using advanced APEX functionalities so that the team members get the best user experience.

Course Road Map

Unit 1: Enhancing Application Pages



Lesson 3: Using Collections

Lesson 4: Using Dynamic Actions

Lesson 5: Using Plug-ins

Lesson 6: Incorporating Interactivity

Unit 2: Adding Advanced APEX
Functionality to an Application

Unit 3: Making an Application
Production-ready

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this unit, you learn APEX features and functionalities that will help you enhance applications pages.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

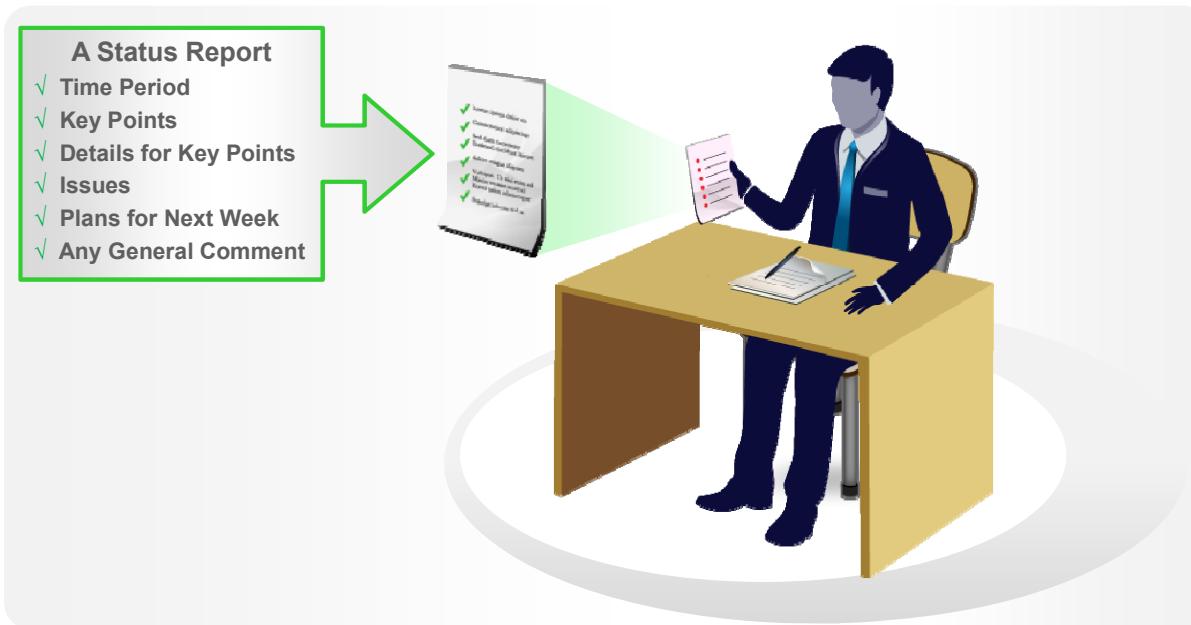
3

Using Oracle APEX Collections

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Using Collections in PTS



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To allow users to create status reports, Jack realizes that he needs to gather certain details from users such as the time period for which they want to submit the report, the main points they want to mention, any additional details they want to specify for the main points, any issues they encountered in the week, their action items or plans for the coming week and so on.

Instead of collecting all this information on one page, he decides to build a couple of pages for creating a status report. He also wants to allow users to preview the email before sending it to Jill. The challenge Jack faces now is how to store the session data that is collected in each page and refer it in other pages.

Objectives

After completing this lesson, you should be able to:

- Describe collections
- Identify APIs available for using collections
- Create, access, and manipulate collections

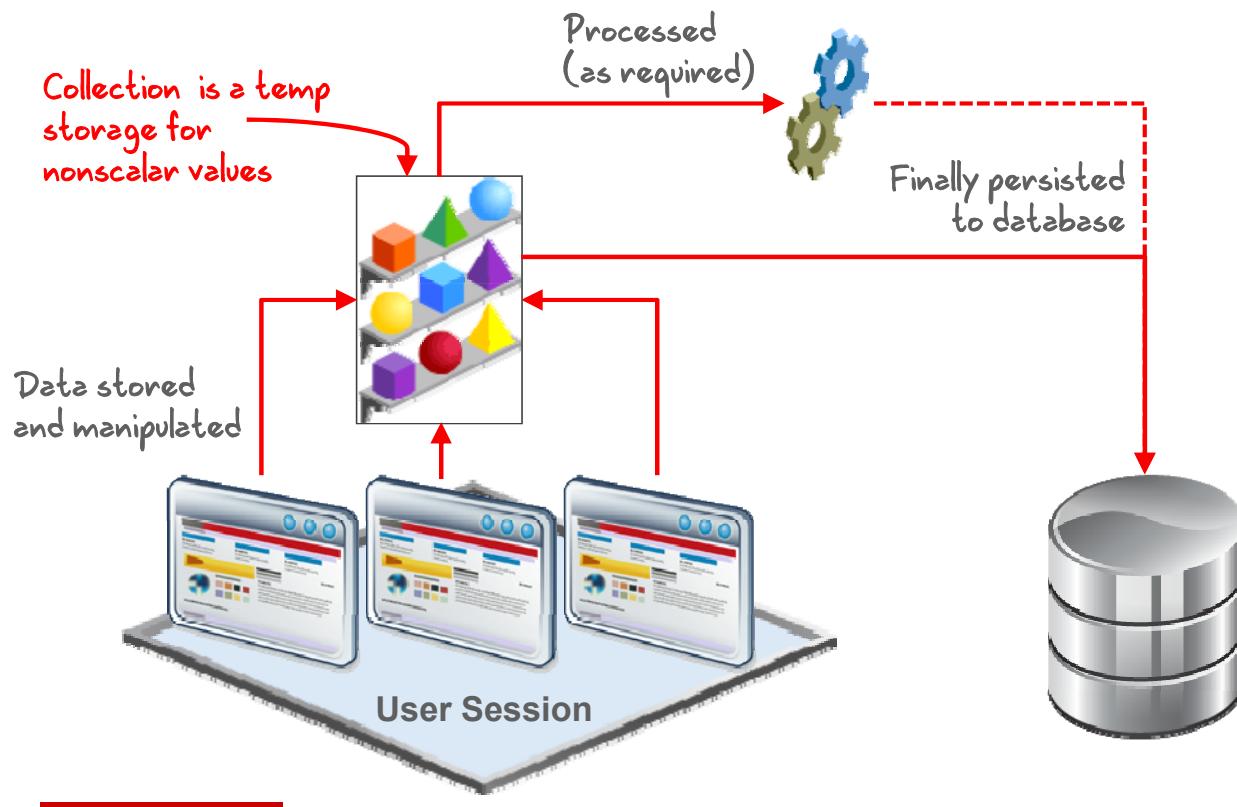


ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learn what are collections in Oracle APEX and how they can be used in an application. You look at some of the most frequently used collection methods and learn how to create and use collections.

What Is a Collection?



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Collection is a feature in Oracle APEX that allows you to temporarily store nonscalar values in session state. You use collections to store data, present in a user's session state, in row and column format. You can perform multiple validations and processes on this collection before finally persisting the required data in a database. A collection can be accessed and manipulated from multiple pages within the user's session. APEX automatically takes care of all the mechanics related to session state management.

Consider the requirement in the Project Tracking System that Jack is building. He needs to gather details like status period, key points and so on from the users to create a status report. If he were to use a database table to store all these values, he would have to create processes in each page that will write this data to the database. If he wanted to display the details collected from one page to another page, he will need to write processes to fetch the data from the database. Users might modify this data many times before finalizing their report. All the update operations will need to be written to handle any modification made by users. Other than the overhead of writing these processes, this scenario also involves multiple database sessions.

What is a Collection?

Let us consider an alternative scenario where Jack uses hidden items and/or application items to store the collected data to avoid the multiple database sessions as seen in the previous scenario.

In doing so, Jack would face certain challenges:

- His application might get too messy with the number of hidden items and application items.
- He will need to either restrict the data entered in the report to some number of key points, or create many hidden items to accommodate the unknown number of key points users might enter.

Jack can avoid these problems if he makes use of the collections feature in Oracle APEX.

Collections will allow Jack to temporarily store the collected data in session state itself, thus avoiding multiple database sessions. Jack need not worry about managing the data in the session state from page to page as the APEX engine will take care of all the session state management mechanics.

Common Use Cases



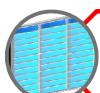
Data Entry Wizard



Multiple Updates to a Page



Collecting Unknown Number of Attributes



Build a Tabular Forms Manually



Create Report on Dynamic SQL



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You use collections in APEX applications in the five most common use cases shown in the slide:

- **Data entry wizard:** In certain situations, before you can complete a logical or physical transaction, you want users to enter data for a sequence of tasks or options. Consider a scenario in a student management system. For a given semester, students need to select various programs and activities they want to enroll into. You can build a wizard that allows the students to select these programs from various departments and divisions in the university. On a final page, you can show them all the choices they have made and if they want to finalize it. When they finalize the data, you want to save the data into the respective tables.

In such situations, you use collections to temporarily store the contents of the multiple rows of information entered through multiple pages. You then use the data in the collection to perform a final step in the wizard when both the logical and/or physical transactions are completed.

- **Multiple updates to a page:** Consider a situation when an application requires a page on which a user can update multiple detail rows.

You can use collections here so that users can make many updates, apply these updates to a collection, and then call a final process to apply the changes to the database.

- **Collecting unknown number of attributes:** You might want to build a wizard where you need to collect an arbitrary number of attributes. An ideal example is the shopping cart feature required in e-commerce sites. Here, you are not sure about the number of products a user might purchase.
You can use collections to store this data and at the end of the wizard, you take the temporarily stored data in the collection and apply it to the database.
- **Build Tabular Forms manually:** Oracle APEX provides a built-in functionality to create Tabular Forms. However, in few situations, these tabular forms might not meet your business requirements and you will need to create a tabular form manually.
You need to use the collections concept, along with manually creating items and referencing global variables, to create a manual tabular form.
- **Create reports on Dynamic SQL:** When you have to display a report based on Dynamic SQL, you can create a collection based on the Dynamic SQL and then create the report by querying the collection.



Point to Ponder

What category do you think Jack's requirement for collections falls under?

APEX_COLLECTION API: Overview

- APEX_COLLECTION is a PL/SQL Application Programming Interface (API).
- You use this API to:
 - Create a collection
 - View data in a collection
 - Insert rows to the collection
 - Update the collection
 - Delete the collection



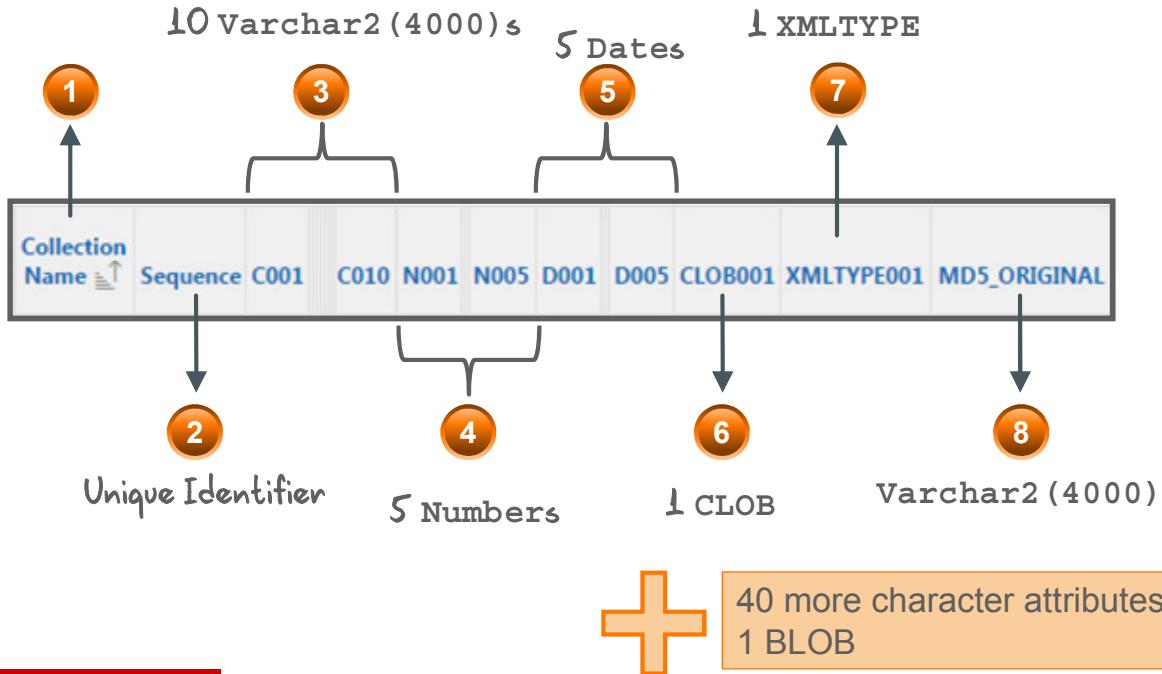
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle APEX provides a PL/SQL API for working with collections. You use this API to create a collection and perform many data manipulation operations on the collection. Few of these collections are listed in the slide. There are more operations that can be performed on a collection like merging collections, resequencing a collection, sorting collection data and so on. For a complete understanding of all these operations, refer to the Oracle Application Express API Reference guide.



Collection Structure and Data Types

A collection is created with the following default structure:



When you create a collection, it has a predefined data structure as shown in the slide.

1. A unique name for the collection.
2. An incremental sequence ID that uniquely identifies each row in a collection. Each row in a collection is called a member.
3. Columns to store character data. Columns in collections are called attributes. When you create a collection, you will be able to see only 10 character columns. However, you can use up to 50 such character columns. You refer to these columns by using the convention c001, c002, ..., c050. You can store up to 4000 characters in each of these columns.

Note: Collections are optimal for normal sized datasets and not very large data.

4. Columns to store numeric data.
5. Columns to store date values.
6. Column to store CLOB data. You use CLOB when the character data you want to store exceeds 4000 characters. There is also a column available to store binary or BLOB data.
7. Column to store well formed XML data.
8. Column to store message digest for the row.

Creating a Collection

Points to consider while creating a collection:

- Which of the create APIs to use?
- Where to place the code?

API	Usage (C=>Collection)
CREATE_COLLECTION	Creates an empty C with the given name
CREATE_OR_TRUNCATE_COLLECTION	If C with given name exists, it truncates (empties) the C. Else, creates empty C with given name
CREATE_COLLECTION_FROM_QUERY CREATE_COLLECTION_FROM_QUERY_B	Creates and populates a C using given SQL query (up to 50 character attributes in C)
CREATE_COLLECTION_FROM_QUERY2 CREATE_COLLECTION_FROM_QUERYB2	Creates and populates a C using given SQL query (first 5 number attributes, then up to 50 character attributes in C)



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are two points to consider when you create a collection:

- Which of the APIs available to create a collection do you need?
 - Do you want to create an empty collection? There are two APIs available to create an empty collection, which are the first two APIs shown in the slide table.
 - Do you want to create a collection based on the results of a SQL query? There are four APIs for this.
 - The two APIs, listed in the third row of the slide table, allow you to create collections based on SQL queries. The API ending with B offers significantly faster performance. However, it limits the size of all column values to 2000 bytes and does not compute MD5 for any of the rows.
 - The last two APIs also create collections on SQL queries, but require that the first five columns in the select list be number values and the next five be data columns. After that you can select up to 50 character columns. The API ending with B2 gives better performance.

When you create a collection, you need to give it a name. This name cannot exceed 255 characters. Also, the collection names are not case-sensitive and are always converted and stored in uppercase.

- When do you want the create collection code to be executed.?
 - You might choose to place this code at pre-rendering or post-rendering points,. Alternatively, execute this code when the page is submitted.



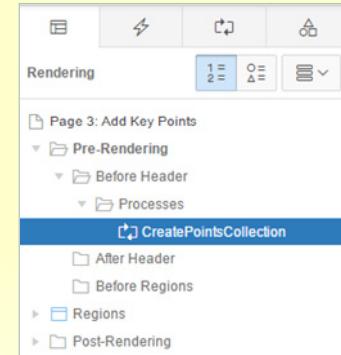
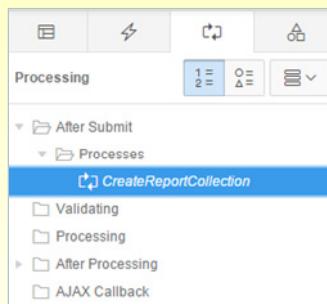
Point to Ponder

Which of the Create APIs should Jack use to create a collection in the PTS application?
Where should he create the process, which will create a collection?

Creating a Collection: Syntax and Example

```
APEX_COLLECTION.CREATE_COLLECTION  
( p_collection_name IN VARCHAR2 );
```

```
APEX_COLLECTION.CREATE_COLLECTION(  
    p_collection_name => 'REPORT');
```



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The syntax for the `CREATE_COLLECTION` API is given in the first box in the slide. The `CREATE_COLLECTION` API takes the collection name as a parameter.

In the second box, an example of using the API is given.

Jack chooses the `CREATE_COLLECTION` API to create a collection to store status report details. He decides to use two collections: one called REPORT to store the report period, issues, plans for next week, and comment for the week; another called POINTS to store the key points and additional details for the key points. He decides to create the REPORT collection when the first page of the wizard, which allows users to select a start and end date for the report, is submitted. That is, when the Next button is clicked in the Enter Report Period page, the REPORT collection is created and the entered dates are stored in the collection. On the next page of the wizard, users can enter their key status points for the report. In this page, Jack creates a collection called POINTS in the Before Header section under Pre-Rendering.

Viewing the Collection in Session State

The screenshot shows the Oracle Application Express session state viewer. The toolbar at the top includes items like Items, Pages, Queries, Tables, PL/SQL, Images, Debug, Session (which is selected), and Errors. Below the toolbar, there are fields for Page (set to 3), Rows (set to 50), Find, View (set to Collections), and a Set button. Below these, session information is displayed: Application: 23419 Salome PTS, Session: 1939939626641, User: ORA01, Workspace: 222551113530400223, and Browser Language: en. The main area is a grid table titled 'Collections' with columns for Collection Name, Sequence, C001 through D002. Two rows are present: 'POINT' and 'REPORT'. The 'POINT' row has a sequence of 1 and values for C001 through C009 all set to '-' (dash). The 'REPORT' row also has a sequence of 1 and similar dash values. The last two columns show dates: 04/06/2015 and 04/10/2015.

Collection Name	Sequence	C001	C002	C003	C004	C005	C006	C007	C008	C009	C010	N001	N002	N003	N004	N005	D001	D002
POINT	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
REPORT	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	04/06/2015	04/10/2015

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When a page that creates or updates a collection renders, you can view what the values are in the collection in session state.

To view the collection details in session state:

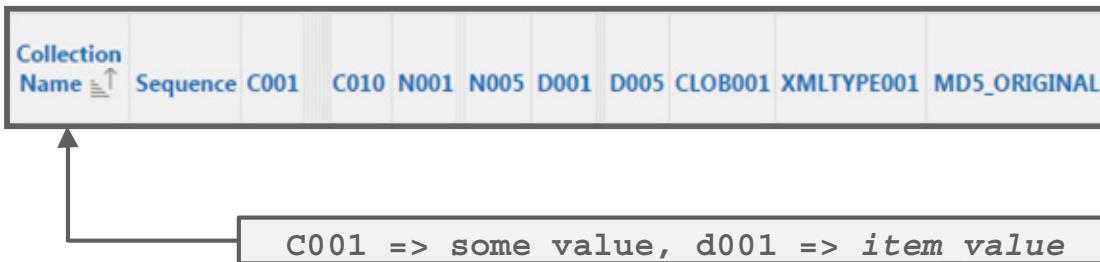
1. Run the page containing the code to create or access a collection.
2. Select Session in the Developer toolbar.
3. Select Collections from the View drop-down menu and click Set.

The collection details will be displayed as existing in the session state.

Adding Members to a Collection

Points to Remember:

- The collection must be created.
- Seq_id is generated automatically.
- Map the values to the respective member attribute.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

After you have created a collection, you want to add one or more members or rows of information to the collection.

You must ensure that the collection to which you need to add members is already created. Otherwise an exception will be thrown.

Each member in a collection is uniquely identified by a seq_id. This ID is a number value generated internally in incremental order starting at one. As you add members to a collection, the seq IDs change in increments of one, with the newest member having the largest ID.

As mentioned previously, a member can have up to 50 character attributes, five number attributes, five date attributes, one CLOB, one BLOB, one XMLType, and one MD5. You must map the values you want to store in a member to its respective collection attribute.

Adding Members: Syntax and Example

```
APEX_COLLECTION.ADD_MEMBER ( p_collection_name IN VARCHAR2,  
    p_c001 IN VARCHAR2 default null, ...  
    p_c050 IN VARCHAR2 default null,  
    p_n001 IN NUMBER default null, ...  
    p_n005 IN NUMBER default null,  
    p_d001 IN DATE default null, ...  
    p_d005 IN DATE default null,  
    p_clob001 IN CLOB default empty_clob(),  
    p_blob001 IN BLOB default empty_blob(),  
    p_xmltype001 IN XMLTYPE default null,  
    p_generate_md5 IN VARCHAR2 default 'NO');
```

```
declare  
    seq_id          varchar2(100);  
begin  
    seq_id := apex_collection.add_member(p_collection_name => 'REPORT',  
                                         p_d001 => :P5_START_DATE,  
                                         p_d002 => :P5_END_DATE);  
  
    :reportid := seq_id;  
end;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are three APIs available to add members to a collection:

1. ADD_MEMBER procedure
2. ADD_MEMBER function
3. ADD_MEMBERS procedure

The ADD_MEMBER procedure adds the given member to the named collection.

The ADD_MEMBER function adds the given member to the named collection and returns the seq_id of the newly added member. You can use the function API if you need to save the seq_id of a member and refer it elsewhere. If you need not use the seq_id, you can use the procedure itself.

The syntax for the procedure and function are the same. This is shown in the first box in the slide. The second box gives an example of using the ADD_MEMBER function.

You can also add new members (or an array of members) to a collection using a ADD_MEMBERS procedure. The number of members added is based on the number of elements in the first array (c001).

Accessing a Collection

```
SELECT c001, c002, c003, n001, d001  
FROM APEX_collections  
WHERE collection_name = '<collection name in CAPS>'
```

```
select  
    APEX_ITEM.DISPLAY_AND_SAVE(1,seq_id) Sno,  
    APEX_ITEM.TEXT(2,c001) KeyPoint  
from apex_collections  
where collection_name = 'POINT'
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

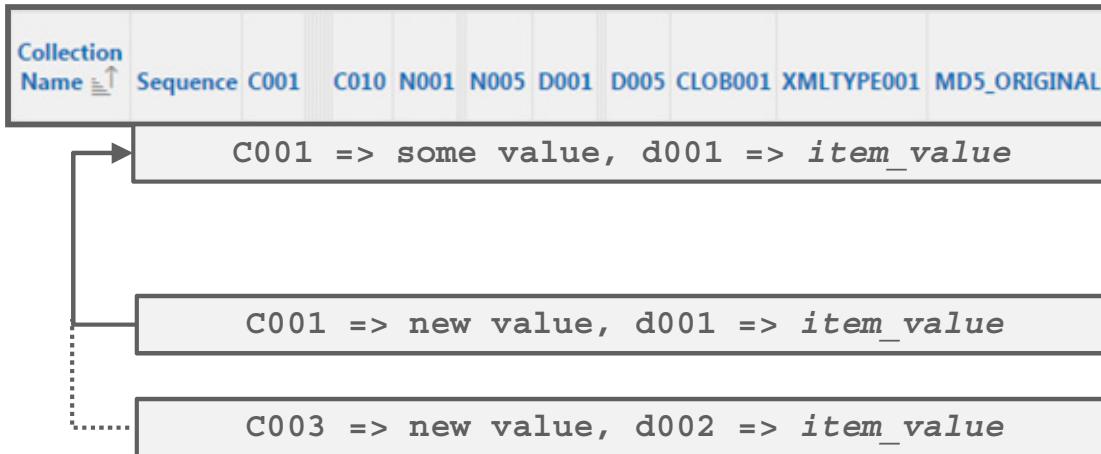
You can access the collection in the session state by creating a report on the collection. You can select the required attributes or columns by specifying the attribute name. You must ensure that the collection name is given in uppercase. The slide shows the syntax and example of using a select query on a collection.



Updating Members of a Collection

Point to Remember:

- Update method will replace an entire collection member.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The update member APIs allow you to replace an entire collection member, not individual member attributes.

Note: There are APIs available to update individual member attributes, which are discussed next in this lesson.

You use the update member API when you want to completely replace a member in a collection. You need to mention the seq_id for the member you want to update.

Assume you have a collection member that stores data for attributes c001 and d001. If you use the UPDATE_MEMBER API to store values for attributes c003 and d002, the data already stored in c001 and d001 will be lost and these fields will be set to null.

Updating Members: Syntax and Collection

```
APEX_COLLECTION.UPDATE_MEMBER ( p_collection_name IN VARCHAR2,  
                                p_seq IN VARCHAR2 DEFAULT NULL,  
                                p_c001 IN VARCHAR2 default null, ...  
                                p_c050 IN VARCHAR2 default null,  
                                p_n001 IN NUMBER default null, ...  
                                p_n005 IN NUMBER default null,  
                                p_d001 IN DATE default null, ...  
                                p_d005 IN DATE default null,  
                                p_clob001 IN CLOB default empty_clob(),  
                                p_blob001 IN BLOB default empty_blob(),  
                                p_xmltype001 IN XMLTYPE default null,  
                                p_generate_md5 IN VARCHAR2 default 'NO');
```

```
APEX_COLLECTION.UPDATE_MEMBER (  
    p_collection_name => 'POINT',  
    p_seq => APEX_APPLICATION.G_F01(i),  
    p_c001 => APEX_APPLICATION.G_F02(i));
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

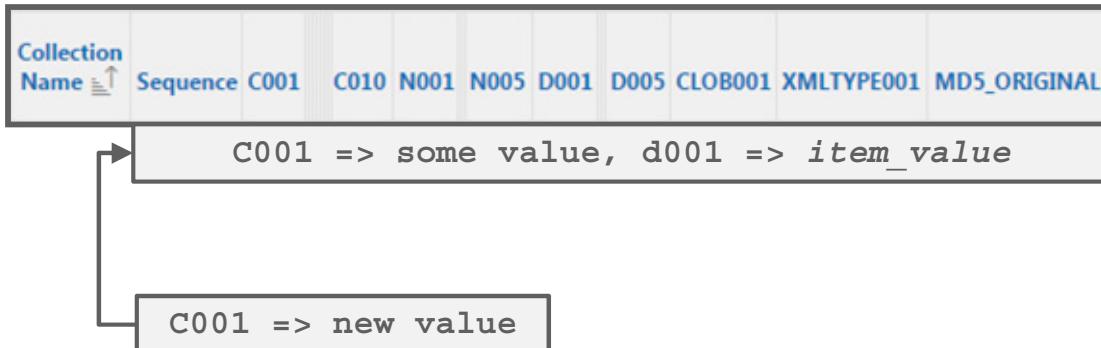
You can update collection members by using two APIs:

- UPDATE_MEMBER procedure: To update a single member.
- UPDATE_MEMBERS procedure: To update multiple members.

The first box in the slide shows the syntax for the UPDATE_MEMBER procedure. It is very similar to the ADD_MEMBER procedure. The only difference is that the UPDATE_MEMBER procedure accepts a seq_id parameter in addition to the rest of the parameters.

The second box in the slide shows an example of using the UPDATE_MEMBER procedure.

Updating Member Attributes of a Collection



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can update a specific member attribute in a collection member. For example, you can just change the value for the c001 attribute and retain the original values for the rest of the collection attributes.

Updating Member Attributes: Syntax and Collection

```
APEX_COLLECTION.UPDATE_MEMBER_ATTRIBUTE (
    p_collection_name IN VARCHAR2,
    p_seq IN VARCHAR2,
    p_attr_number IN VARCHAR2,
    p_attr_value IN VARCHAR2);
```

```
begin
    apex_collection.update_member_attribute(
        p_collection_name => 'REPORT',
        p_seq => :reportid,
        p_attr_number => '6',
        p_attr_value => :P7_ISSUES);
end;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The slide shows one of the definitions of the UPDATE_MEMBER_ATTRIBUTE and an example of using the procedure. You should specify a valid collection name and seq_id.

The third parameter depends on the attribute you want to change.

- For character attribute, use p_attr_number and specify the attribute number to modify. For example, if you want to update c006, specify 'p_attr_number => 6'. Use p_attr_value to specify the value for the attribute.
- To update the CLOB attribute, use p_clob_number and p_clob_value as the third and forth parameters.
- To update the BLOB attribute, use p_blob_number and p_blob_value .
- To update the XMLTYPE attribute, use p_xmltype_number and p_xmltype_value.
- To update a number attribute, use p_attr_number and p_number_value.
- To update a date attribute, use p_attr_number and p_date_value.

Other Useful Methods

API	Usage (C=>Collection)
COLLECTION_EXISTS	Returns True if C with given name already exists. Else, returns False.
COLLECTION_MEMBER_COUNT	Returns total number of members in the C
RESEQUENCE_COLLECTION	Removes gaps in seq IDs, but maintains member order
DELETE_MEMBER	Deletes member with given seq_id
TRUNCATE_COLLECTION	Removes all members from the C with the given name
DELETE_COLLECTION	Deletes all members of C and the C itself



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The slide lists some of the other useful APIs available while working with collections. For a complete list of all the collection API methods, refer to the Oracle Application Express API Reference guide.

Additional Resource: Collections Packaged Application

The screenshot shows the Oracle APEX Sample Collections application. The left sidebar has a dark blue background with white text and icons. It includes links for Home, Basic Collections (which is selected and highlighted in orange), Data Synchronization, API Examples, and Administration. The main content area has a light gray header with the title "Sample Collections" and a subtitle "Use collections to store and manipulate large amounts of data". Below this is a detailed description of Oracle APEX collections. The main content is divided into three cards:

- Basic Collections**: Shows a red icon of a folder with a document. Description: "Creating a Basic Collection. Create a user defined collection name on a simple Employee table structure."
- Data Synchronization**: Shows a blue icon of a circular arrow. Description: "Populate a collection from a sample Employees table, update the collection, and then synchronize the data in the collection with the base table."
- API Examples**: Shows an orange icon of a clipboard. Description: "View Oracle APEX API examples which you can use in your own application."

At the bottom left is the Oracle logo, and at the bottom right is the copyright notice "Copyright © 2015, Oracle and/or its affiliates. All rights reserved."

A Sample Collections packaged application is provided in Oracle APEX, which you can refer to for additional information on how collections can be created and used.

Summary

In this lesson, you should have learnt how to:

- Describe collections
- Identify APIs available for using collections
- Create, access, and manipulate collections



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You should now be able to describe collections and how they can be used in an application. You are familiar with the common collections APIs and can use them successfully in an application.

Jack has used Oracle APEX Collections in the wizard pages for creating a status report. He uses two collections to store data entered by the users. He uses REPORT collection to store the start and end date for the status report along with the issues, plans for next week, and general comment for the week. He uses another collection called POINTS to store the key points. He creates a manual tabular form on the POINTS collection, so that users can easily add and delete key points and details. He then displays the data entered by the users in a Preview Email page.

Practice 3 Overview: Using Oracle APEX Collection

This practice covers the following topics:

- Creating and Updating a Collection
- Accessing a Collection



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In these practices, you will use collections feature of Oracle APEX to add functionality that will allow users to place an order for multiple products in the GlobalMart Management Tool.

4

Creating Dynamic Actions

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Creating Dynamic Actions in PTS



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the PTS application that Jack is upgrading, he has a page that allows users to enter the Key Points for their status report. On this page, users should be able to add n number of Key Points. They should also be able to delete a Key Point if required.

Jack realizes that if he creates On Submit processes on this page, it will hinder the user experience as users will have to wait for the page to be submitted and refreshed each time. He wonders if he can perform these actions without submitting the page by using dynamic actions.

Jack has previously used dynamic actions in the PTS application for some basic interactivity. He now decides to take a closer look at the dynamic action capabilities of Oracle APEX.

Objectives

After completing this lesson, you should be able to:

- Describe how dynamic actions work in Oracle APEX
- Use custom code and advanced properties to create dynamic actions
- Delete a Row in a Report using dynamic actions
- Process a Modal Window



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This lesson discusses how to extend your application to process a modal window and delete a row in a report using dynamic actions.

Review: What is a Dynamic Action?

Dynamic actions are behaviors that are executed as users interact with the application.

Dynamic Action Example

Employee Details	
First Name	Diana
Last Name	Lorentz
Email	DLORENTZ
Phone Number	590.423.5567
Hire Date	07-FEB-99
Job Id	IT_PROG
Salary	4200
Commission Pct	(disabled)
Manager Id	103
Department Id	60

Employee Details	
First Name	Diana
Last Name	Lorentz
Email	DLORENTZ
Phone Number	590.423.5567
Hire Date	07-FEB-99
Job Id	SA_REP
Salary	4200
Commission Pct	(enabled)
Manager Id	103
Department Id	60

Commission Pct is disabled when Job is not Sales Representative.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Dynamic actions are specific behaviors that are executed as and when users interact with the application page. Usually, such interactions are built into web applications using scripting tools like AJAX and JavaScript. Oracle APEX allows you to easily define dynamic actions by selecting and specifying some values. You can create dynamic actions without writing the entire JavaScript code on your own.

Consider the example in the slide. In the screenshot on the left, when the value for Job is IT_PROG, the Commission Pct item is disabled. In the screenshot on the right, when the value for Job is SA_REP, the Commission Pct item is enabled.

Note: Dynamic actions increase the size of a page. In most cases, this overhead is not significant. But in cases where you are primarily focused on performance or require more control over the interactivity, then you might need to manually code your dynamic actions.

Review: Basic Dynamic Actions

Some of the common dynamic actions are:

- Hide/Show items and buttons
- Enable/Disable items and buttons
- Highlighting an Item
- Creating a Cascading LOV
- Setting the Value of an Item



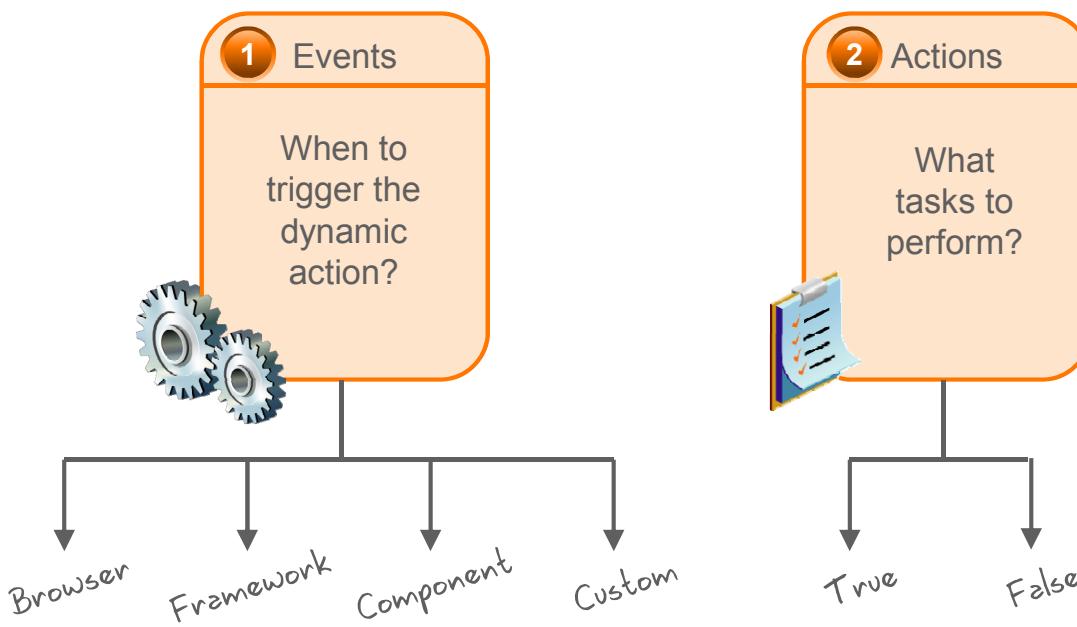
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Some of the common dynamic actions that are required in most applications are listed in the slide. How to create these dynamic actions is covered in detail in the *Oracle Application Express Workshop I* course.

In this course, you review the various options available to define dynamic actions and closely look at some of the more advanced and complex dynamic actions you can create in an application.



Components of Dynamic Actions



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are two main components of a dynamic action:

- **Events**: The event determines what will cause the dynamic action to run, and when that dynamic action is triggered.
- **Actions**: The actions contain the code that should be run when a dynamic action is triggered.

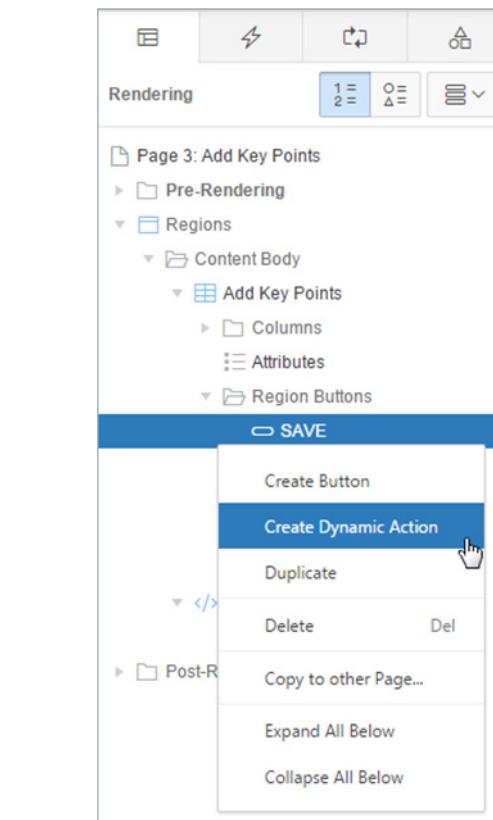
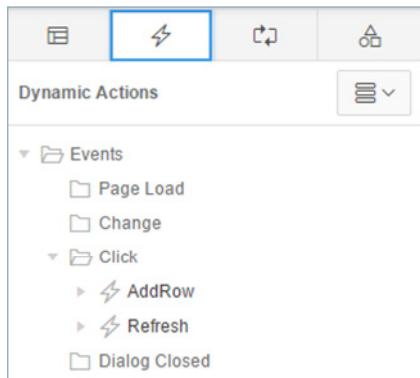
In Oracle APEX, you can define dynamic actions for three major types of events: Browser, Framework, and Components. If there are any other events that you want to trigger your dynamic action, you can create a Custom event. You learn how to create a custom event later in this lesson.

The actions you define are of two types: True and False. The actions defined as True are run when the event specified occurs. The actions defined as False are run when the event specified does not occur.

Creating a Dynamic Action

Create a dynamic action from:

- A specific item or button
- The Dynamic Actions tab



ORACLE®

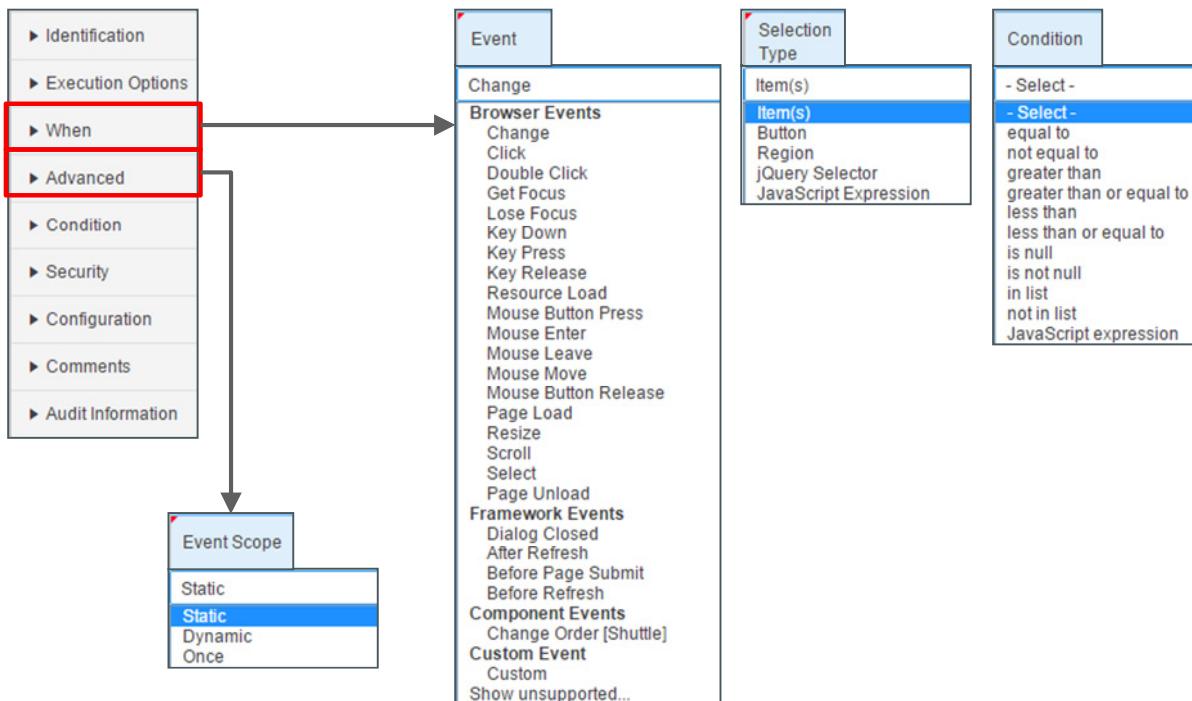
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In Oracle APEX, you can create a dynamic action in two ways:

- From a specific item
- From the Dynamic Actions tab.

Whichever method you choose, once the dynamic action is created, the details are listed in the Dynamic Actions tab as well as the specific item it is related too.

Dynamic Actions: Events



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

After selecting Create Dynamic Action, you are shown a page where you can define the dynamic action. The When and Advanced sections are the main sections to fill out. Here, you define the event that will trigger the dynamic action.

The When section has the following sections:

- **Event (mandatory):** The Event field has a list of values from which you can select the event that should trigger the dynamic action. If none of these events meet your requirements, you can define your own Custom event. You learn how to create a Custom event later in this lesson.
- **Selection (mandatory):** Here, you select the page component that relates to this event. It could be an item, a button, or a region. You can either directly specify the item, button, or region, or you can use jQuery Selector or a JavaScript Expression to identify the triggering element on the page.
- **Condition (optional):** You use this Condition field to link the event to the True or False action based on the condition. If this condition is met, the True actions are performed. If the condition is false, the False actions are performed. If you do not specify a condition here, the True actions are run when the event is triggered.

In the Advanced section, you can specify the scope of the event. That is, you can define how the events are bound to the triggering elements. You can select one of the following three options:

- **Static:** Select this option to bind the event to the triggering elements till the current page is refreshed. If the page is refreshed, then the event is not longer bound to the triggering elements.
- **Dynamic:** Select this option to bind the event to the triggering element for the lifetime of the page, including any page refreshes.
- **Once:** Select this option to bind the event to the triggering element for only one time.

Other than these two sections, there are other sections that are similar to the usual APEX features or objects.

In the Identification section, you enter a unique name for the dynamic action.

In Execution Options, you specify a sequence number that determines the order in which the dynamic action is executed.

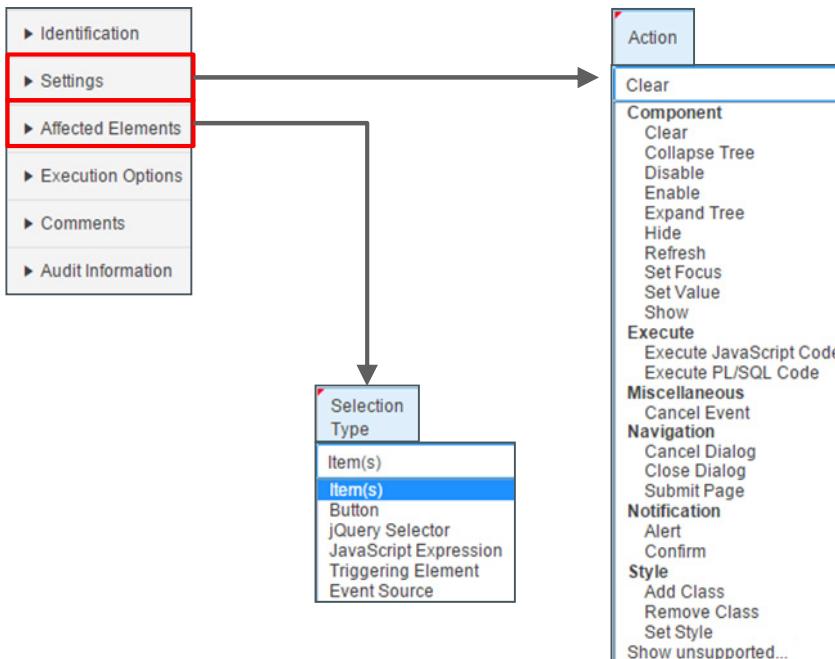
The Condition section is used to specify whether the dynamic action should be run or not.



Point to Ponder

What Event should Jack use in the PTS application to add a new row to the Key Points tabular form?

Dynamic Actions: Actions



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the previous slide, you saw how to define the event that will trigger the dynamic action. Next, you define the actions that should be performed as True or False actions. Right-click the True or False node and select the Create True Action or Create False Action, respectively. A page displays on the right side of the Page Designer.

The two main sections you need to fill out to define the True or False actions are Settings and Affected Elements. In Settings, you choose the required action to be performed. Under Affected Elements, you choose the page elements on which the selected action should be performed.

Note: Not all the Actions will allow you to select an affected element. For example, notifications actions do not require an affected element to be defined.



Point to Ponder

What Action should Jack define when the Add Row button is clicked (Event).

Deleting a Row in a Report

To create a dynamic action to delete a row in a report:

1. Create the column with a link to click to delete the row.
2. Create a dynamic action event that is triggered when the delete link is clicked.
3. Create a set of actions that should be performed when the delete link is clicked:
 1. Ask for confirmation to delete
 2. Get the ID of the row to delete and set it to a hidden item
 3. Execute the code to delete the row
 4. Refresh the report
 5. Display a notification that the row was deleted



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You may have a requirement in an application to delete rows from a report dynamically. This can be achieved by creating a Event that will trigger a series of actions.

There are three main steps to achieve this interactivity, which are listed in the slide.

First, you need to modify the report and create a new column that will contain the link to initiate the delete Event. To do this, perform the following steps:

1. Right-click the report region and select Create Virtual Column.
2. Enter an appropriate column heading.
3. Select URL for Target and enter # in the URL field.
4. Specify the link text. You can also choose to display an image as the link. To display a red cross icon as the link, enter the following code in the Link Text field:

```

```
5. You need to specify the following attributes for the Link. The class will be used in the jQuery to select the row to delete. The ID value will be used to fetch the ID of the row selected for delete.

```
id='#ID#' class="delete"
```

The link column to initiate the delete Event is created.

The second step is to create the Dynamic Action Event component. To do this, perform the following steps:

1. On the Dynamic Actions tab, right-click the Click node and select Create Dynamic Action.
2. Enter an appropriate name to uniquely identify the dynamic action.
3. In the When section, you should have the following values:
 - Event: click
 - Selection Type: jQuery Selector
 - jQuery Selector: a.delete
4. In the Advanced section, select Dynamic for event scope.

The Event to trigger the dynamic action is created.

The final step is to create the actions that should be executed when the Event is triggered.

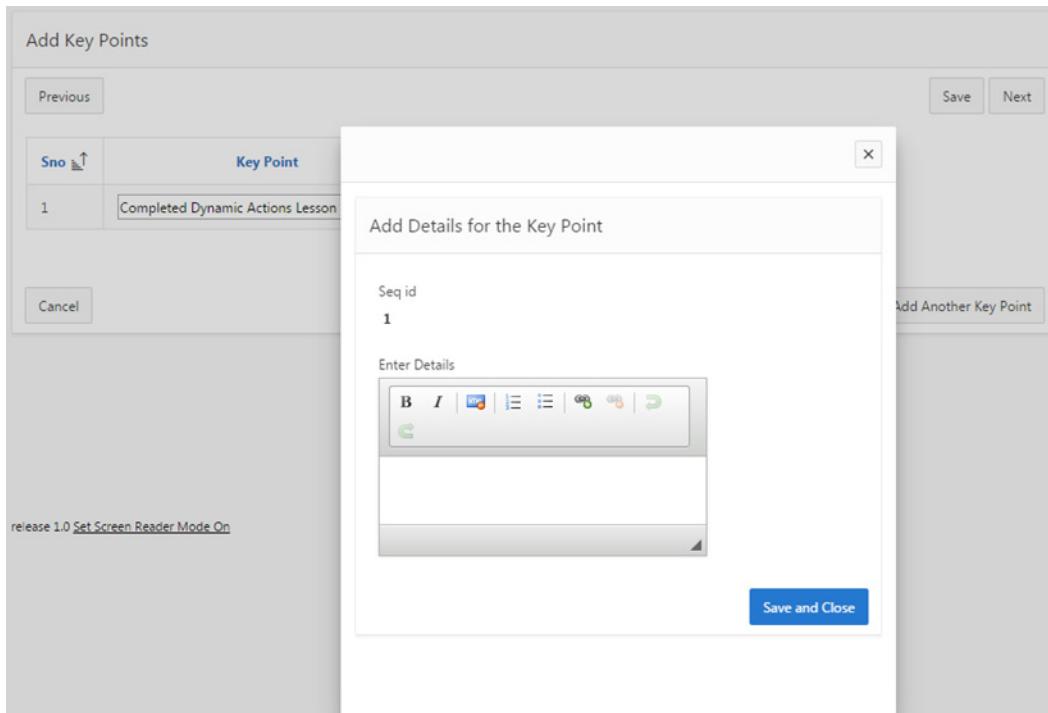
To create a True action, right-click the True node and select Create True Action. By default, a Show True action is created when an Event is created.

You need to define the following True actions that should be run when the delete icon is created:

- **Confirm:** Enter confirmation message in the Text field under Settings.
- **Set Value:** Create a hidden item and set its value to the ID of the selected row.
 - Under Settings, select JavaScript Expression for Set Type and enter this.triggeringElement.id for JavaScript Expression.
 - Under Affected Items, set Selection Type to Item(s) and select the hidden item for Item(s)
 - Select No for Fire on Page Load
- **Execute PL/SQL Code:** Enter the code to delete the row from the table or collection and select the hidden item for Page Items to Submit
- **Refresh:** Select Region for Selection Type and select the region containing the report
- **Alert:** Enter a message notifying that the row is deleted.

In the PTS application, Jack uses these steps to create dynamic actions that will allow users to delete Key Points from the Add Key Points page.

Processing a Modal Window



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A modal window or dialog is a child page that you invoke from a parent page. When the modal page is displayed, the parent page is made inactive in the background. After you have completed and closed the modal page, the parent page becomes active again.

To create a modal window, you create a page and specify the page type as Modal. You can call this page as required, and atomically this page is displayed as a modal window.

In the PTS Application, Jack invokes a modal window from the Add Key Points report. He has added a link to the report that allows a user to click the link to enter additional details for that Key Point. In the modal window, the user can enter details and save the details to the collection. When the modal window is closed, the Key Points report is refreshed and the added details are displayed.

Creating Custom Events

1. Create a custom dynamic action
2. Define the action
3. Invoke the event

The screenshot illustrates the process of creating a custom dynamic action and invoking it. It consists of three panels:

- Panel 1 (Top Left): Identification** shows a dynamic action named "Custom Event DA" with a sequence of 10 and a custom event type "customEvent".
- Panel 2 (Bottom Left): Action** shows the action set to "Execute JavaScript Code" with the following code:

```
alert( 'customEvent triggered, ID:' + this.data.id );
```
- Panel 3 (Right): Execute when Page Loads** shows the JavaScript code:

```
apex.event.trigger(document,'customEvent',{ id: 101 });
```

A red box at the bottom left contains the **ORACLE** logo.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can define custom events and invoke a custom event to execute the dynamic action. The slide shows a simple example to explain how custom events can be created and used.

Note: In most cases, custom events are not required. You will be able to create a dynamic action using the many events that are listed in Oracle APEX. In the practice for Lesson 6, you will create custom dynamic actions.

1. Click the Dynamic Actions tab in Page Designer, right-click the Events node, and select Create Dynamic Action.
2. Under When, select Custom for Event and enter a name for the Custom Event. In the example, customEvent is used.
3. Select JavaScript Expression for Selection Type and enter document in the JavaScript Expression field.
4. Click the default Show action under the True node and change the action to Execute JavaScript Code.
5. Enter the code you want to run. In the example, a simple alert box is displayed.
6. You have so far created the dynamic action. Now you need to invoke the event so that the dynamic action can be fired.
7. Click the page name node and under JavaScript, specify the code to invoke the event in the execute when page loads field.
8. Save and run the page. An alert dialog box should be displayed.

Additional Resource: Sample Packaged Application

The screenshot shows a web-based application interface for 'Sample Dynamic Actions'. On the left is a dark blue sidebar menu with options like Home, Simple, Style, Server Side, Complex, and Administration. The main content area has a title 'Sample Dynamic Actions' with a gear icon and a subtitle 'Add interactivity to your applications with dynamic actions'. Below this is a message 'Please click on any of the examples to get started.' followed by a grid of 12 cards arranged in three rows of four. Each card contains a title, a brief description, and a color-coded border.

Disable/Enable	Hide/Show	Add/Remove Class (Error)	Add/Remove Class (Focus)
Disable and enable items automatically	Declaratively hide and show items based on user input	Using the Add and Remove Class actions to highlight errors	Using the Add and Remove Class actions to show focus
Stripe Report	Execute PL/SQL Code	Set Values (SQL)	Set Values (PL/SQL)
Using a dynamic action plug-in to enhance a report.	Executing PL/SQL as part of a dynamic action	Using a dynamic action to execute SQL	Automatically calculate values via PL/SQL based on user input
Timer	Refresh	Filter and Refresh	Shuttle Refresh
Utilize a timer for repeating dynamic actions	Refresh a report based on user interactions with an interactive report	Use a dynamic action to filter a report	Use a dynamic action with a shuttle page item to control a report
Delete and Refresh	Slider Plug-In		
Use a multi-part dynamic action to confirm deletions and refresh a report	Use a dynamic action with a custom plug-in page item		

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A Sample Dynamic Actions packaged application is provided in Oracle APEX, which you can refer to for additional information on how dynamic actions can be created and used.

Summary

In this lesson, you should have learned how to:

- Describe how dynamic actions work in Oracle APEX
- Use custom code and advanced properties to create dynamic actions
- Delete a Row in a Report using dynamic actions
- Process a Modal Window



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You should now be able to create simple, complex, and advanced dynamic actions. You should be able to create a dynamic action to delete a row in a report. You should also be able to use and process modal pages in Oracle APEX.

Jack has used dynamic actions in this PTS application to achieve the following functionalities:

In the Add Key Point page, he has included a button called Add Row. When this button is clicked, a dynamic action is fired, a member is added to the POINTS collection, and the manual tabular form region in the page is refreshed.

He has also included a new column in the tabular form that displays a delete icon link. When this link is clicked, a dynamic action is fired and a series of actions are run to delete the row from the report and display a refreshed report.

Jack has also made use of modal pages to allow users to enter additional details for any entered key point.

Practice 4 Overview: Using Dynamic Actions

This practice covers creating advanced dynamic actions.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this practice, you modify the Add to Cart and Place Order functionality created in the GlobalMart Management Tool. You use dynamic actions to deliver a better user experience.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

5

Using Plug-ins in an Application

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Using Plug-ins in PTS



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Previously, Jack used a couple of item plug-ins in the Project Management System. The plug-ins got him many compliments from Jill and other team members.

He now discovers that there are many more plug-ins made by the active APEX community. He decides to look up the various plug-ins available and incorporate some of them in the Project Tracking System.

Objectives

After completing this lesson, you should be able to:

- Identify different plug-ins available for Oracle APEX
- Import and use plug-ins
- Optimize the performance of plug-ins

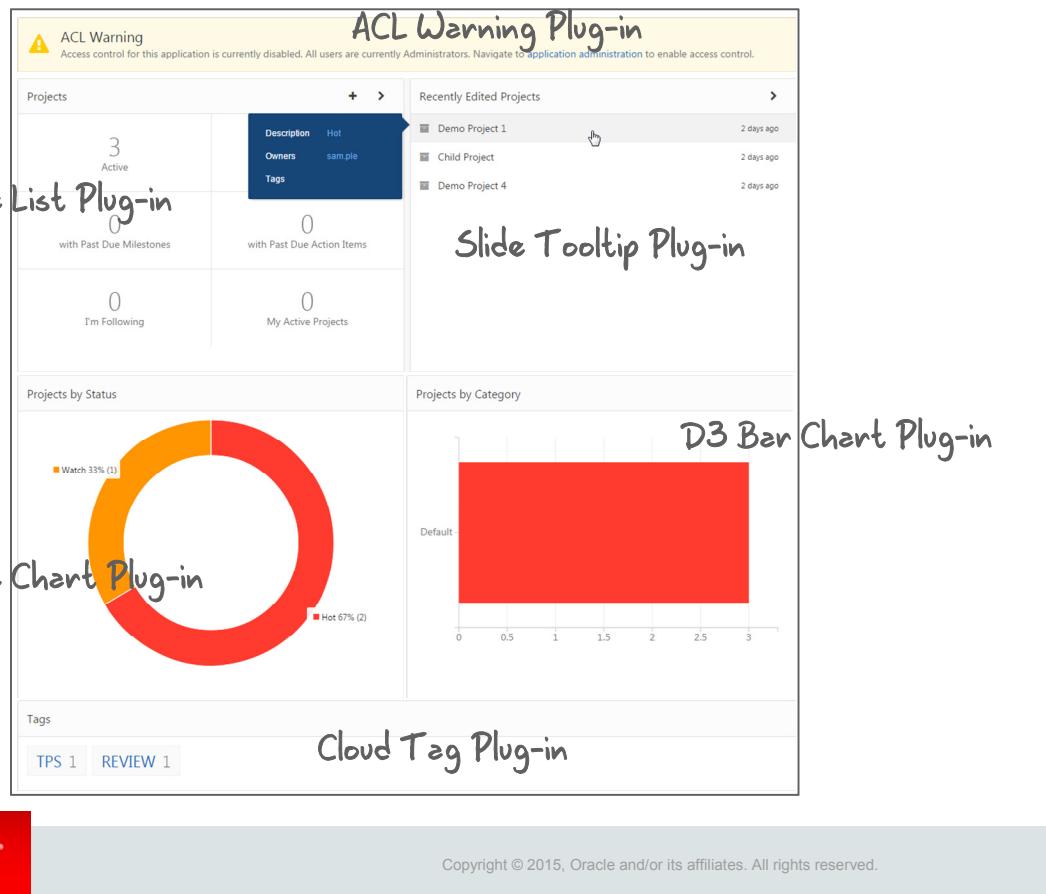


ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This lesson discusses how to extend your application by importing and using different types of plug-ins.

What Is a Plug-In?



ORACLE®

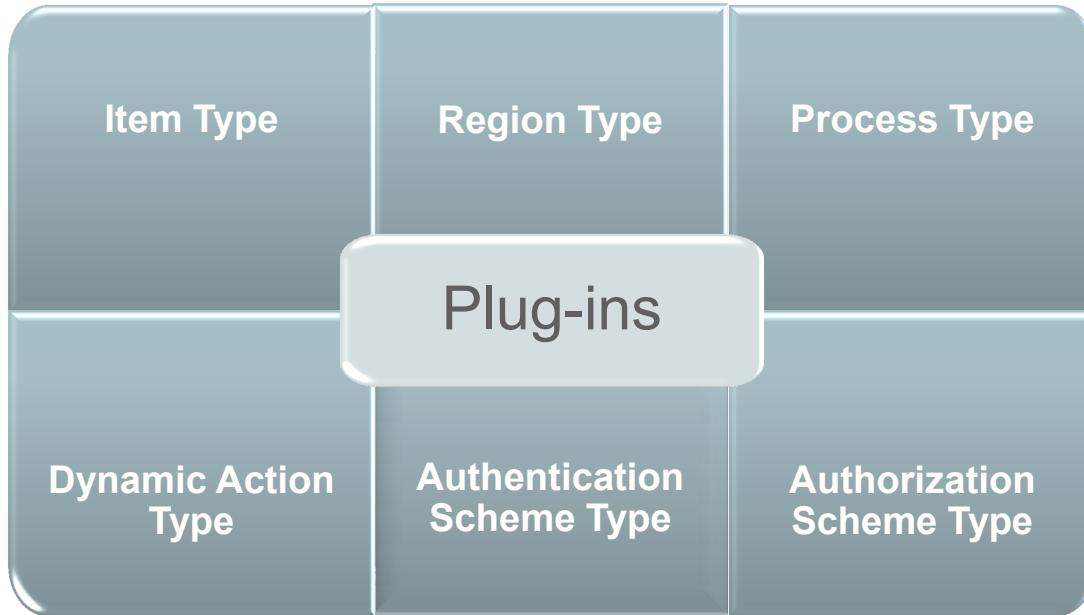
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Plug-ins are code snippets created and shared by Oracle APEX developers and APEX community developers that provide rich user functionalities. You can declaratively use the capabilities of a plug-in within an application. Plug-ins promote sharing and reusability within the Oracle APEX community.

Because plug-ins are designed for reuse, developers can export and import them to other applications in the same or another workspace, and also share them with the Application Express Plug-in community by using the Plug-in Repository.

The example in the slide is Page 10 of the P-Track packages application. It shows different types of region plug-ins such as Badge List Plug-in, Slide Tooltip Plug-in, D3 Bar Chart Plug-in, ACL Warning Plug-in, Cloud Tag Plug-in, and Flot Pie Chart Plug-in.

Types of Plug-ins



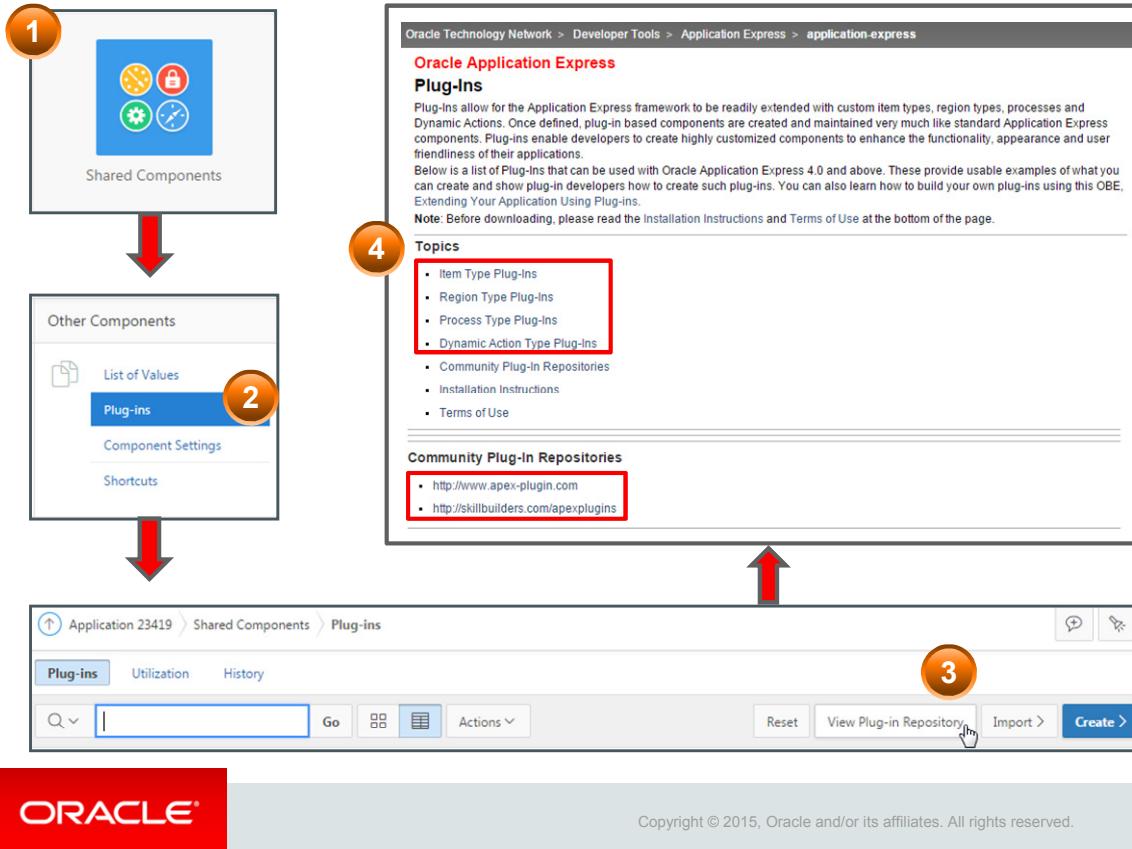
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In Oracle APEX, you can create or use different types of plug-ins. These are listed in the slide.

In this lesson, you learn to use the Item, Region, Process, and Dynamic Actions type plug-ins.

Accessing the Plug-In Repository



The plug-in repository provides a series of plug-ins developed by Oracle that can be used by customers to perform various tasks. This repository continues to be updated with additional plug-ins for use by the Oracle APEX user community.

To access the plug-in repository:

1. Access the application development home page and click Shared Components.
2. On the Shared Components page, under Other Components, click Plug-ins.
3. From the Plug-ins page, click the View Plug-ins Repository button.
4. The repository is located on the Oracle Technology Network (OTN). An OTN page is displayed in a new web browser.

Click the different types of plug-ins to view examples of each type. Scroll down to view links to community plug-in sites.

How to Use a Plug-In in an Application

To use a plug-in in an application:

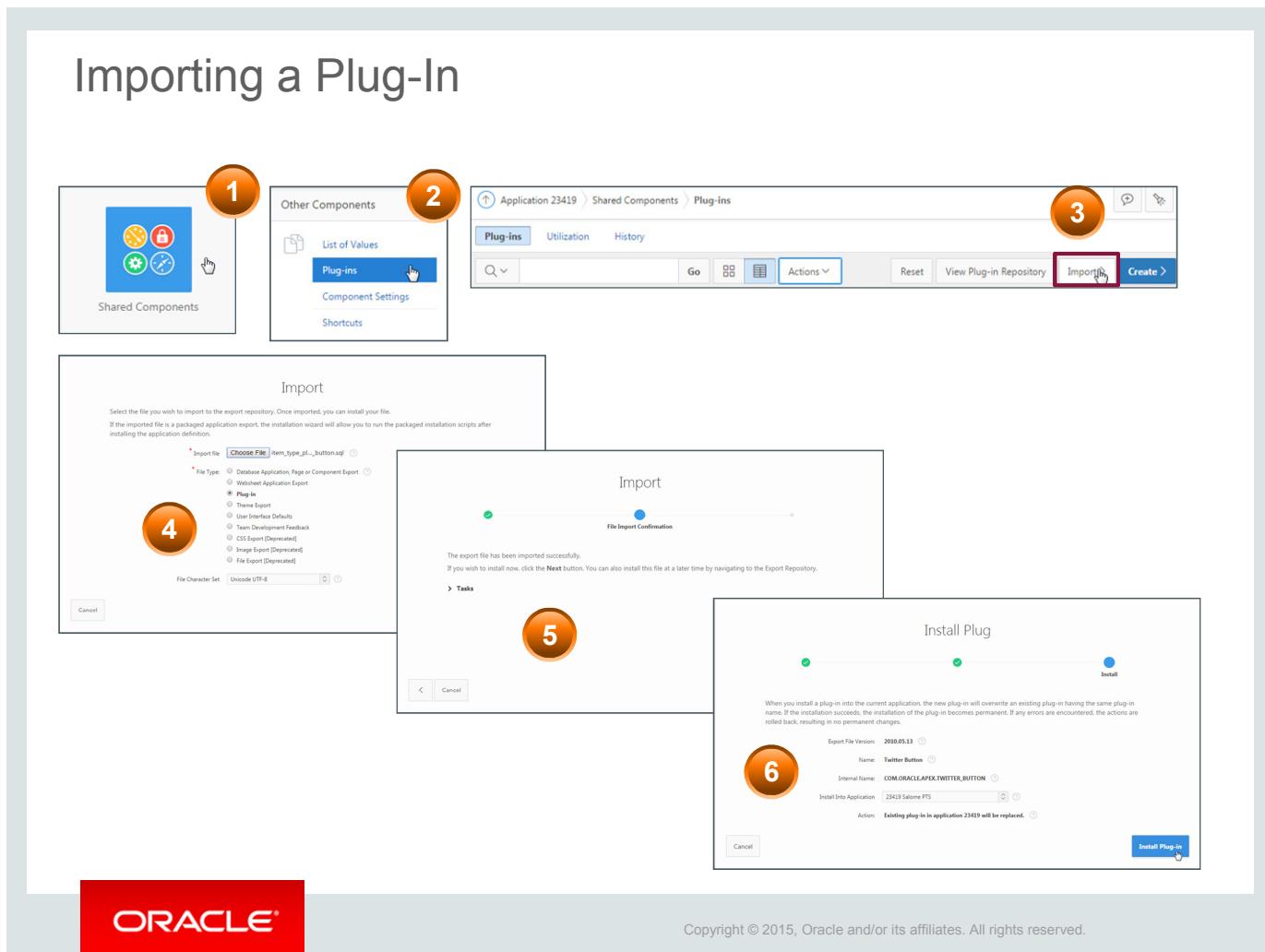
1. Import a plug-in.
2. Edit or create an item, region, process, or dynamic action type to use the plug-in.
3. Run the application to test the plug-in functionality.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The slide shows the steps involved in using a plug-in within an application.

1. You need to have access to the plug-in installation file. You can then import the plug-in to the application where you want to use it. Note that a plug-in is imported into an application and not to an entire workspace. That means, if you want to use a plug-in in different applications within the same workspace, you need to import that plug-in into all those applications.
2. Depending on the type of plug-in, you need to create or edit an item, region, process, or dynamic action to use the plug-in.
3. After the plug-in is configured, you can run the page or application to test the plug-in functionality.



To use a plug-in in your application, you need to import or create it under Shared Components.

To import a plug-in:

1. Navigate to your application's Shared Components page.
2. Under Other Components, select Plug-ins.
3. You can use the Import option to import an exported plug-in to an application. Click Import.
4. Select your plug-in import file and click Next.
5. After the file is imported, click Next to install it.
6. Select the application you want to install the plug-in into and click Install Plug-in.

The plug-in is imported into an application. If the plug-in already exists in the target application, it will be replaced.

To review the contents of any plug-in, navigate to Shared Components > Plug-ins for the application, and select the plug-in you want to review. Details of the plug-in are displayed.

Using an Item Plug-in

1

2

3

4

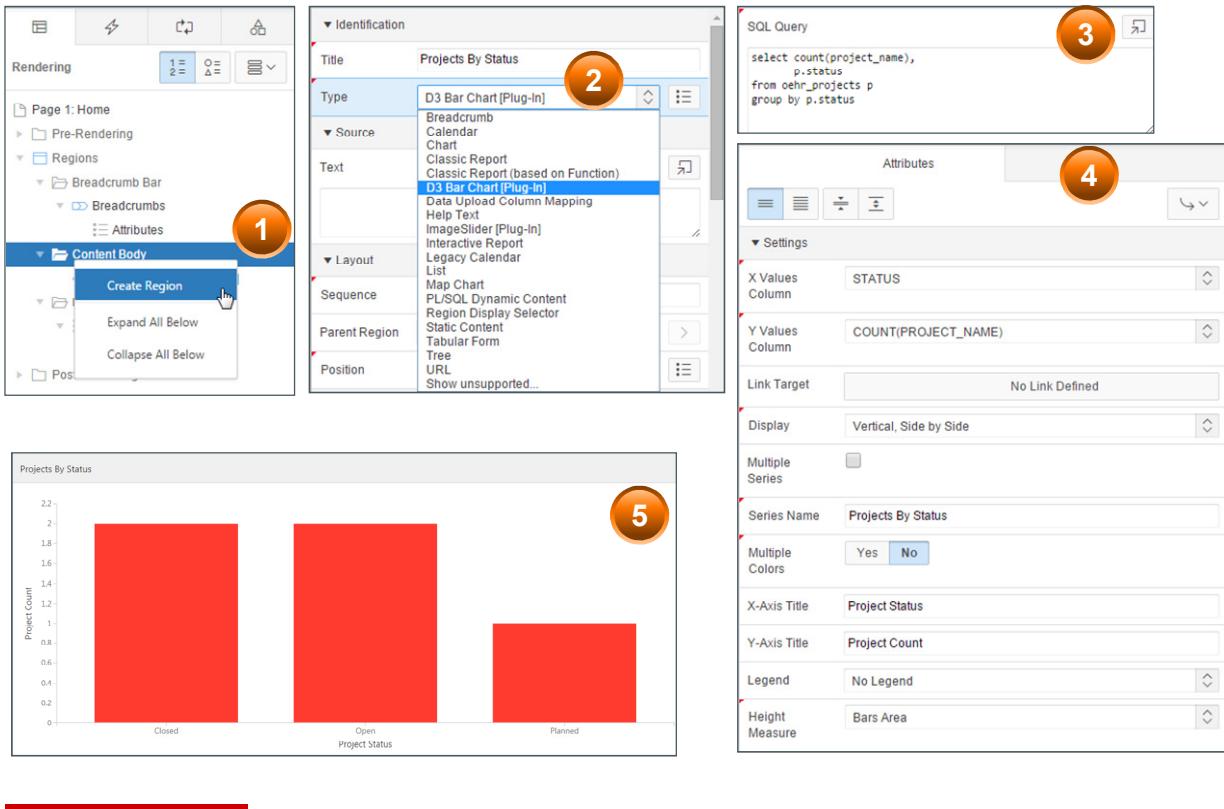
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The slide shows the steps to use an item plug-in. As discussed earlier in this lesson, you first need to import the plug-in into the application where you want to use it. Then, perform the following steps:

1. Create or select the item that should use the plug-in.
2. On the item description page that displays at the right-side in the Page Designer, select the plug-in from the drop-down list for the Type field.
3. Under Settings, select the field values depending on the plug-in being used.
4. After the plug-in is configured, run the page to view the plug-in functionality.

In the PTS application, Jack imports a PicInsideEditBox item plug-in. This plug-in, when applied to a text box field, displays a small image icon at the extreme left inside the text box field. The plug-in provides a common set of images that can be used with text boxes. Jack uses the plug-in in the Create Employee form page to enhance the appearance of the user entry form page.

Using a Region Plug-In



ORACLE®

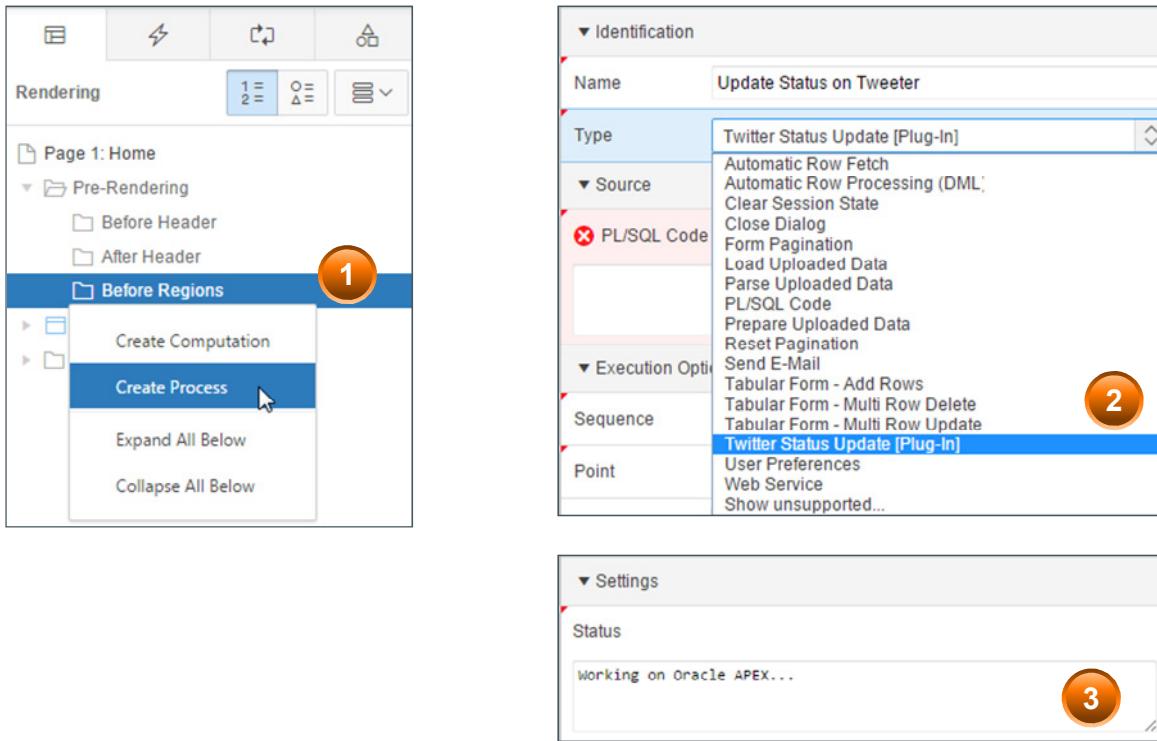
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The slide shows the steps to use a region plug-in. After importing the plug-in, perform the following steps to create a plug-in region:

1. Navigate to the page where you want to create the plug-in region and right-click the Content Body node. Select Create Region.
2. Name the region and for Type, select the region plug-in.
3. Almost all the region plug-ins will require a Source SQL Query that will be used to populate the region. Enter the SQL Query.
4. Fill out the region attributes. This will differ for each region plug-in.
5. Click Save and run the page. The region using the plug-in will be loaded.

In the PTS application, Jack imports a D3 Bar Chart region plug-in. He uses this plug-in to display a chart visual in the application home page to show the total number of projects under each of the project status categories.

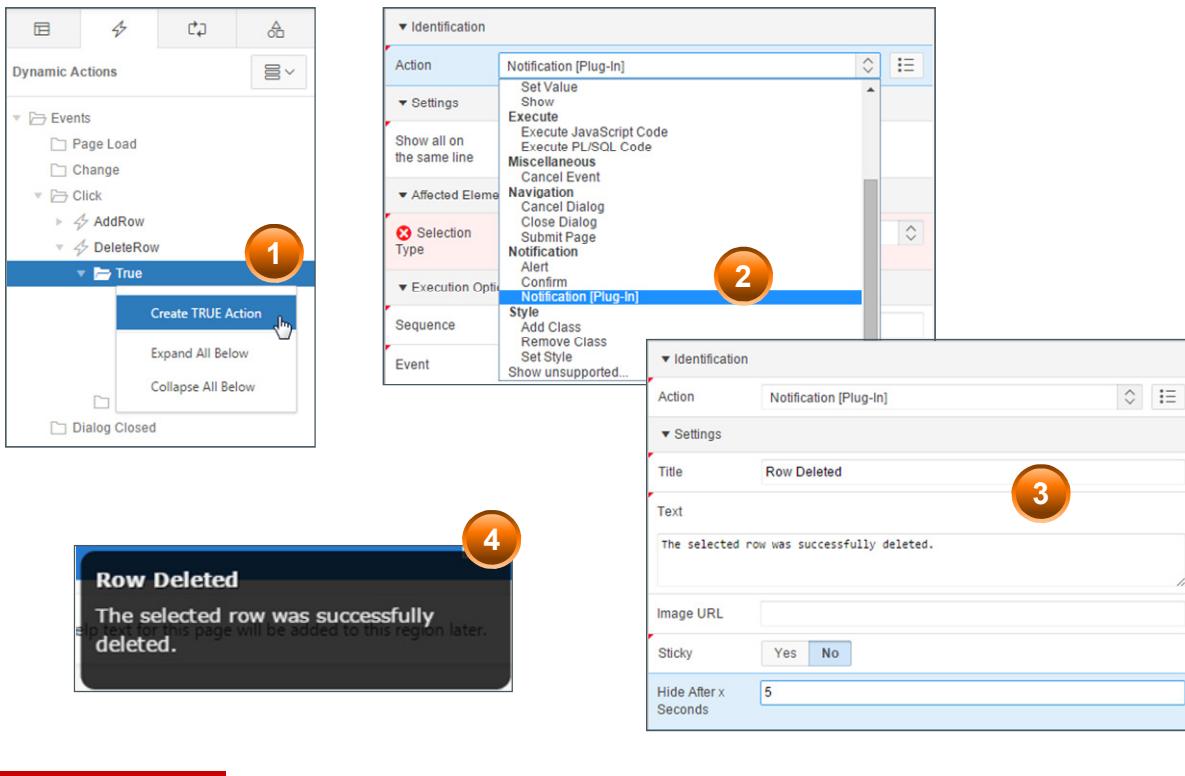
Using a Process Plug-in



The slide shows the steps to use a process plug-in. After importing the plug-in, perform the following steps to create a plug-in process:

1. Navigate to the page and position where you want to create the process, right-click and select Create Process.
2. Select the plug-in from the Type drop-down list.
3. Fill the settings specific to the plug-in.
4. Save and run the page. The process will be executed.

Using a Dynamic Action Plug-In



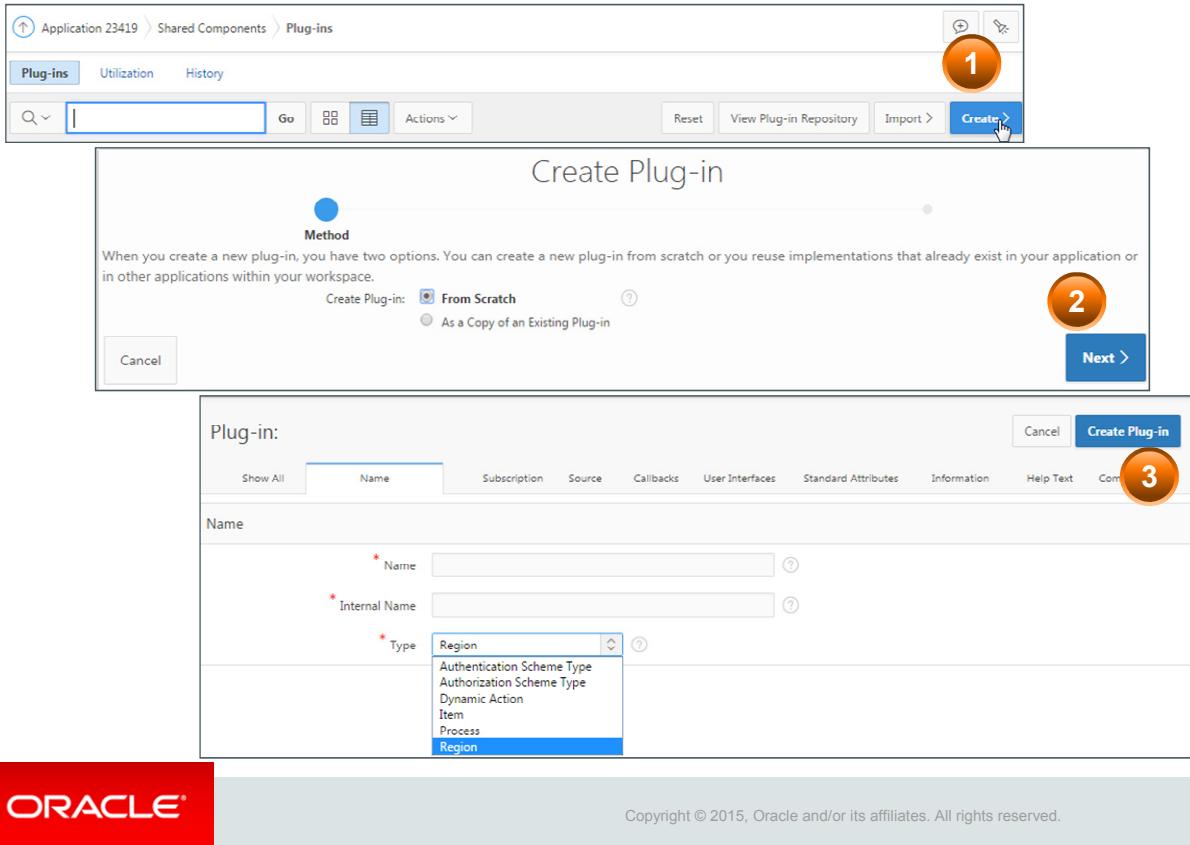
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The slide shows the steps to use a dynamic action plug-in. After importing the plug-in, perform the following steps to create a plug-in dynamic action:

1. Navigate to the page where you want to create the plug-in dynamic action. Click the Dynamic Actions tab, right-click the True node under a Dynamic Action event, and select Create True Action.
2. Select the plug-in from the Action drop-down list.
3. Fill out the Dynamic Action settings. This will differ for different plug-ins.
4. Click Save and run the page. The plug-in dynamic action will be displayed when the event is triggered.

In the PTS application, Jack imports a Notification Dynamic Action plug-in. He uses this plug-in in the Add Key Points page. When the delete icon for a key point is clicked, a series of dynamic actions were fired to ask the user for confirmation, to delete the row, and refresh the region. Jack adds a notification dynamic action plug-in at the end of this series of dynamic actions to notify that the row was deleted successfully.

Creating a Plug-In



If you want to create plug-ins for reusability or sharing purposes, perform the following steps:

1. Access the plug-in repository for an application and click the Create button.
2. Select From Scratch and click Next.
3. Fill out the plug-in definition. You should specify a unique name and select a type for the plug-in. Then, enter the code that will define how the plug-in will work. After filling all the details, click Create Plug-in.

Optimizing Performance of a Plug-In

- Invoke the callback from a PL/SQL package rather than from the plug-in.
- Store the files that are attached to the plug-in definition in a directory on the web server.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To optimize the performance and efficiency of your plug-in:

- Invoke the callback from a PL/SQL package rather than from the plug-in. This is because if the PL/SQL code of a plug-in is stored within the plug-in definition, Oracle APEX has to use DBMS_SQL to dynamically execute that code. This means that the PL/SQL engine has to first validate if the code is syntactically correct, parse and compile it, and finally execute it. If the code is already stored in a database PL/SQL package, the code is already valid and pre-compiled, thereby allowing the PL/SQL engine to execute it. To perform the task of invoking a callback from a PL/SQL package:
 1. Create a new PL/SQL package in the database.
 2. Copy the PL/SQL code from the Source area of the plug-in definition into the PL/SQL Package Body.
 3. Create a PL/SQL specification based on the functions in the PL/SQL package.
 4. Change the Callback name to use the package function.
- Store the files that are attached to the plug-in definition in a directory on the web server. A web server is optimized to serve files and can cache it on the middle tier, thereby avoiding unnecessary load on the database.

Summary

In this lesson, you should have learned how to:

- Identify different plug-ins available for Oracle APEX
- Import and use plug-ins
- Optimize the performance of plug-ins



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you should have learned how to extend your application by importing and using different types of plug-ins.

Jack has used different types of plug-ins to improve the user experience and enrich the application functionalities.

Practice 5 Overview: Using Plug-Ins in an Application

This practice covers importing and using the following types of plug-ins:

- Region
- Process
- Dynamic Action



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this practice, you extend the GlobalMart Management Tool by using different types of plug-ins in the application.



Incorporating Interactivity Using JavaScript and jQuery

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Using JavaScript, AJAX, and jQuery in PTS



Hey Jack, hearing lots of good news about the Project Tracking System. You seem to have put a lot of effort into it.



Thanks! Yes. However, the tool I used for development was really user-friendly and I was able to build the application quite effortlessly.

So, are you all done with this application or do you plan to include more features?

I am currently upgrading the application. I have added some new features to the application and I am now thinking about using some of my Java web scripting skills too.

Sounds really interesting. Looking forward to what you have put together.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack is gradually becoming very experienced in using APEX. He has used almost all the functionalities that APEX has to offer to make application pages user-friendly and attractive. A team member appreciates his APEX skills and enquires about what he is planning to do next in PTS.

In his earlier application development efforts, Jack had learnt some Java web scripting concepts. Because APEX has provisions to enter customized and specific JavaScripts and other Java web programming concepts like AJAX and jQuery, Jack decides to enrich the Project Tracking System by using these concepts.

Objectives

After completing this lesson, you should be able to:

- Use JavaScript, AJAX, and jQuery functionalities in an application



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you will learn how an application can be made interactive by using JavaScript, AJAX, and jQuery.

JavaScript, AJAX, jQuery: Overview

JavaScript

- A programming language
- Incorporates interactivity in web pages
- Supported and interpreted by all browsers
- Best used to incorporate client-side processing and validations

AJAX

- Expansion: Asynchronous JavaScript and XML
- Enables submission of page components to server for validation and processing

jQuery

- A framework
- Provides a JavaScript library
- Many free jQuery plug-ins available
- Oracle APEX 5.0 includes jQuery version 2.1.3



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

JavaScript is a scripting language created by Netscape. It is embedded into the HTML code of a web page to add functionality. JavaScript implementations are usually used by web developers to provide enhanced user interfaces and dynamic websites. JavaScript code handles client side interactions.

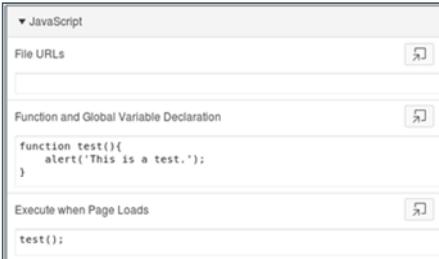
Asynchronous JavaScript and XML or AJAX enables a web page to interact with the server without submitting the entire page. You can use AJAX to fetch or submit values from the server and update a region on a page.

jQuery provides a library of JavaScript functions that can be readily used by developers. Oracle Application Express release 5.0 includes jQuery version 2.1.3.

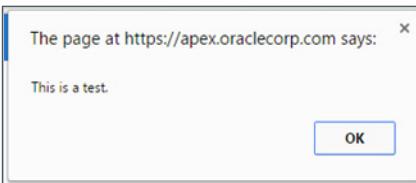
Oracle APEX uses JavaScript and AJAX in many of its features. With the dynamic actions capability in Oracle APEX, you can declaratively create many of the interactions you need easily. However, you might have requirements in an application where you need to use JavaScript, AJAX, or jQuery. The goal of this lesson is not to teach you these technologies. This lesson describes how you can use these technologies within Oracle APEX.

Using JavaScript: A Simple Example

1. Add JavaScript to Page Definition



2. Save and run the page



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The slide shows a simple example that displays an alert box. The JavaScript code function is written in a page's JavaScript section.

Adding JavaScript to a web application is a great way to add features that mimic those found in client/server applications. However, JavaScript is not appropriate for data intensive validations. For example, to verify that a name is contained within a large database table, you must download every record, creating a huge HTML document. In general, complex operations are much better suited for server-side APEX validations instead of JavaScript.

Using JavaScript Functionalities in APEX

There are two main considerations while using JavaScript in APEX:

- Where to write the script?
 - Page HTML header
 - Page JavaScript
 - Separate JS file
- Where to call the script?
 - From links (button, page branch, reports)
 - Target -> URL => # (this will call the JavaScript)
 - From Dynamic Actions
 - From Processes/Computations/Validations



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To use JavaScript with Oracle APEX, you need to know where to write the JavaScript code and how to invoke the code. The slide lists the places where JavaScript can be written and from which components you can call the scripts.

JavaScript APIs

Namespaces	Variable	Functions
apex	apex.gPageContext\$	apex.confirm apex.submit
apex.da	-	apex.da.resume
apex.event	-	apex.event.trigger
apex.item	-	apex.item.hide apex.item.getValue
apex.debug	-	apex.debug.log apex.debug.info
apex.navigation	-	apex.navigation.dialog apex.navigation.redirect



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle APEX provides JavaScript APIs that can be used in JavaScript code to perform various actions. Some of the most common ones are listed in the slide. You can call these functions from within other JavaScript functions. You can also use them with the Execute JavaScript code dynamic action.

An example of using one of these functions is:

Even though the dynamic action framework provides refresh capability for Oracle APEX components, you might require to manually invoke a refresh from JavaScript code. You can do this by triggering the APEX refresh event for the relevant component as follows:

```
apex.event.trigger( "#myRegionStaticID", "apexrefresh" );
```

Specifying Static ID for Regions

Advanced

Static ID	emplist
Custom Attributes	
Region Image	
Image Tag Attributes	
Region Display Selector	<input checked="" type="radio"/> Yes <input type="radio"/> No
Exclude Title from Translation	<input checked="" type="radio"/> Yes <input type="radio"/> No

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In most cases, to refer to a region in a JavaScript or jQuery code, you need to specify a static ID for the region. You can specify a static ID under the Advanced attributes for a region.

Using jQuery

Oracle APEX includes the following libraries:

- jQuery 1.7.1,
- jQuery UI 1.8.22,
- jQuery Mobile - 1.1.1

Reference the jQuery library in JavaScript code using:

- \$
- jQuery
- apex.jquery



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle APEX installs the listed jQuery libraries by default and you can use them without explicitly importing or installing these libraries. Oracle APEX only loads the components of jQuery UI that are required for base Oracle Application Express functionality. Oracle Application Express does not include the entire jQuery UI library because doing so would significantly add to download and processing time for each page load.

You can reference jQuery from within a JavaScript code using the syntaxes given in the slide. Determining when to use the \$, jQuery, and apex.jquery references to the jQuery library in your own JavaScript code depends on where you use it.

Introducing qTip jQuery Plug-in

The screenshot shows the homepage of the qTip2 jQuery plugin website. At the top, there's a navigation bar with links for Home, Demos, Download, Guides, Options, Plugins, API, FAQ, Donate, and Forum. The 'Home' link is highlighted with a yellow square. Below the navigation, a large header features the text "Introducing... qTip²" next to a small owl icon. On the left, there's a sidebar with sections for "About", "Features", and "Compatibility". The "About" section describes qTip² as the second generation of the advanced qTip plugin for the ever popular jQuery framework. It highlights its user friendly nature, feature rich base, and various licenses like MIT/GPLv2. The "Features" section lists several key features: Speech bubble tips, Integrated AJAX, Viewport repositioning, z-index stacking, Modal tooltips, Imagemap & SVG support, and IE6 support (BGIframe). Below these, it says "...and lots more!". The "Compatibility" section shows logos for supported browsers: Google Chrome 8+, Mozilla Firefox 3+, Microsoft Internet Explorer 6+, Opera 9+, and mobile devices iOS 2+ and iPad 2+. The footer contains the Oracle logo and a copyright notice: "Copyright © 2015, Oracle and/or its affiliates. All rights reserved."

Oracle APEX installs the qTip jQuery plug-in by default. The slide shows the official website of qTip². This plug-in consists of various functionalities that you can incorporate in an application.

Using jQuery: qTip² plug-in Tooltip Example

1. Link the plug-in details to the page
2. Customize a label text
3. Run the page

Screenshot 1: A screenshot of a web developer tool's JavaScript panel. It shows the following code:

```

    ▼ JavaScript
    File URLs
    /i/libraries/jquery-qtip2/2.0-6.26.2011/jquery.qtip.min.js

    Function and Global Variable Declaration
    jQuery.browser = {};
(function () {
    jQuery.browser.msie = false;
    jQuery.browser.version = 0;
    if (navigator.userAgent.match(/MSIE ([0-9]+)\./)) {
        jQuery.browser.msie = true;
        jQuery.browser.version = RegExp.$1;
    }
});

    Execute when Page Loads
    $(document).ready(function(){
    $('a[title]').qtip();
});

    ▼ CSS
    File URLs
    /i/libraries/jquery-qtip2/2.0-6.26.2011/jquery.qtip.min.css
  
```

Screenshot 2: A screenshot of a configuration interface for a Textarea item. The 'Label' field contains the following HTML code:

```
<a href="#" title="Add any additional information you want to share">Enter comment for the week </a>
```

Screenshot 3: A screenshot of a modal dialog titled 'Enter General Comments'. It contains a 'Previous' button, a 'Next' button, and a 'Cancel' button. In the center, there is a 'Textarea' input field with the placeholder 'Enter comment for the week'. A yellow tooltip box is displayed above the input field, containing the text 'Add any additional information you want to share'.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The steps to use the qTip² jQuery plug-in in an application are given in the slide. First, you navigate to the page where you want to add the qTip functionality. In the JavaScript section for the page, specify the URL to the plug-in. Mention the required functionality code in the Execute when Page Loads section. In some cases, you might have to include a code to fix the broken links issue in the Function and Global Variable Declaration section, as used in the slide example. Then, in the CSS region, specify the URL to the qTip CSS file.

In the example in the slide, you want to specify a custom tooltip for a Textarea item. For this, the required code is written in the Label field.

Save and run the page. Roll the mouse pointer over the item label. A tooltip defined by the plug-in is displayed.

Using AJAX

Use AJAX to interact with the server/database without submitting an entire page.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Using AJAX, you can communicate with the database without submitting the entire page. This is extremely useful when your APEX application is being run on a very slow network. In its most simple usage, using AJAX, you can update a portion of a page without reloading the whole page. Though that seems rather minor, it can provide huge improvements in page loading time and user experience.

AJAX and APEX

Built-in AJAX in Oracle APEX

- Pagination in an Interactive Report
- Cascading LOVs
- Autocomplete items
- Refreshing a region



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You may already have encountered AJAX functionality without really being aware. Plenty of functionality in Oracle APEX uses AJAX. Some of them are listed in the slide.

- Paginating an Interactive Report will fetch the next or previous page from the server and replace the old source.
- Cascading LOVs: When you change the parent select list of another select list, the values have to be refreshed in the child select list.
- Autocomplete items: When you enable lazy loading in an autocomplete item, the values will be refreshed as you enter characters.
- Refreshing a region will replace the current HTML with an updated version fetched from the server

When to Use

Use AJAX when you want to:

- Set session state
- Execute a block of PLSQL
- Update a field of a record when a button is clicked, or a check box is changed.
- Fetch values from the database



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The most common uses cases are listed in the slide.

Some other uses cases are:

- Update a form item based on changes to another form item.
- Load extended information on a report row but only if requested.
- Create a Tree Item where only branches that are requested are loaded.
- Pull content from another page.
- Pull content from an application process.

Summary

In this lesson, you should have learned how to:

- Use jQuery, JavaScript, and AJAX functionalities in an application



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you should have learned how JavaScript, jQuery, and AJAX functionalities can be used with Oracle APEX.

Jack uses the jQuery qTip2 plug-in to customize the labels in the Create Status Report wizard. He also uses JavaScript and AJAX technologies to enrich the PTS application.

Practice 6 Overview: Incorporating Interactivity Using JavaScript and jQuery

This is a guided practice that covers adding:

- Animation
- Customization
- Auto-Scrolling Region



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Perform this guided practice to create rich functionalities in the GMT Application.



Adding Advanced APEX Functionality to an Application

Unit II

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Project Tracking System: Scenario



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack has enhanced the PTS application pages as required. Now, Jack decides to enhance the entire application using advanced APEX features.

Course Road Map

Unit 1: Enhancing Application Pages

**Unit 2: Adding Advanced APEX
Functionality to an Application**

**Unit 3: Making an Application
Production-ready**



Lesson 7: Generating and Using Table APIs

**Lesson 8: Creating and Using RESTful Web
Services**

Lesson 9: Using Templates and Themes

**Lesson 10: Developing a New Theme for Your
Application Using Theme Roller**

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this unit, you learn APEX features and functionalities that will help you add advanced APEX functionalities to the application pages.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

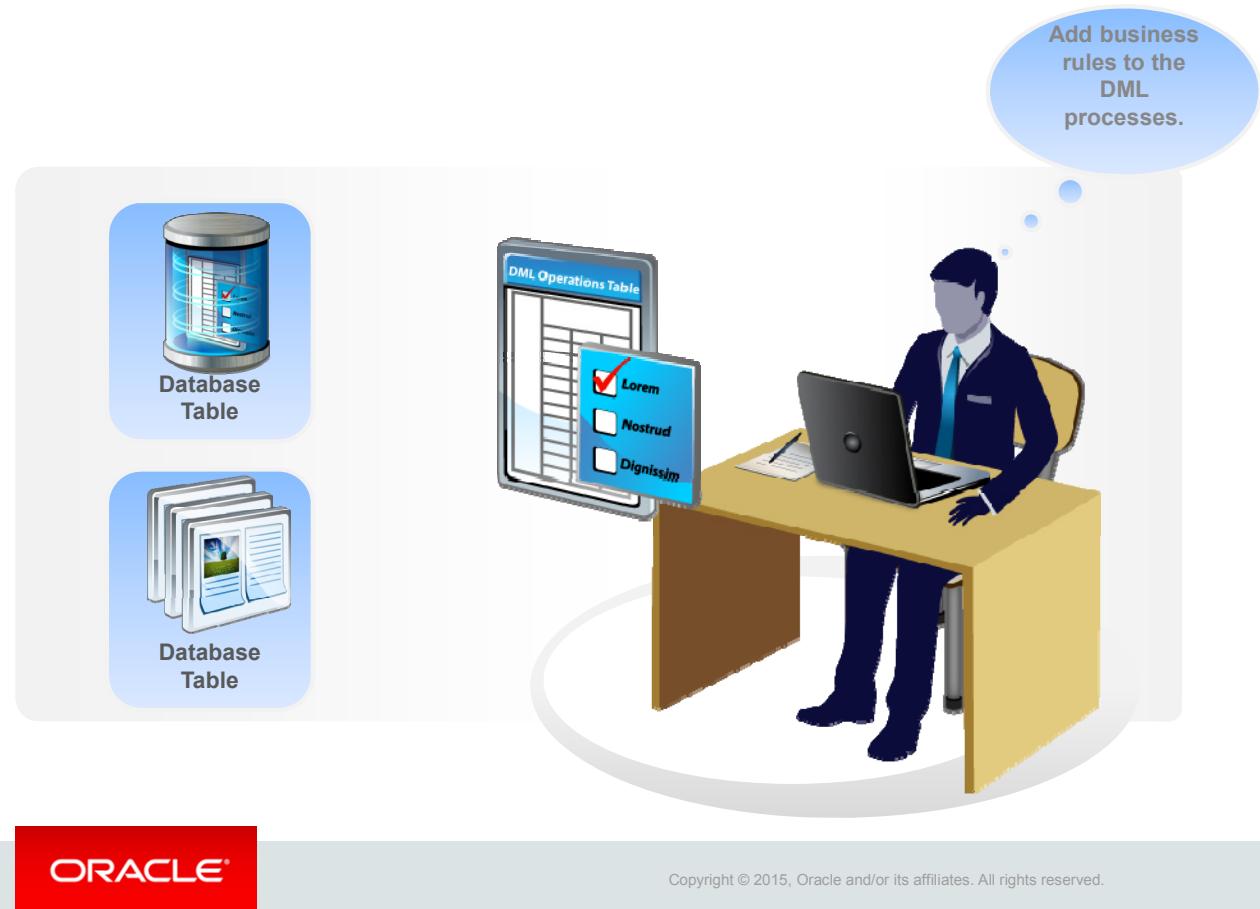
7

Generating and Using Table APIs

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Generating and Using Methods on Tables in PTS



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack realizes that at times, he is creating different pages using the same table. He is now looking for options in APEX that can help him add business rules in all the pages without manually adding validations or computations.

Objectives

After completing this lesson, you should be able to:

- Generate methods on tables
- Modify the API with custom rules
- Create a form that uses the API



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learn how to generate methods on tables, modify the application program interface (API) using custom rules, and then create a form that uses this API.

What Are Methods on Tables?

- A utility that allows you to create a PL/SQL package-based API on a specified table.
- Package on a table includes:
 - INS procedure
 - UPD procedure
 - DEL procedure
 - GET procedure
 - GET procedure—MD5
 - Build MD5 procedure



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The Methods on Tables utility is part of SQL Workshop. Using the Managing Methods on Tables utility, you can generate a PL/SQL package-based API based on a specified table. This utility will create a package that contains:

- **INS Procedure:** To insert records into the table
- **UPD Procedure:** To update records in the table
- **DEL Procedure:** To delete records from the table
- **GET Procedure:** To select records from the table
- **GET Procedure—MD5:** To select a row for update
- **Build MD5 Procedure:** To check whether the MD5 row has been violated before an update occurs

The advantage of using this API rather than using the DML processes provided by APEX is that you can add business rules in the package which will be executed when the API is invoked.

Steps to Generate Methods on Tables

1. Navigate to **SQL Workshop > Utilities**.
2. Click **Methods on Tables**. The Create Table API page appears.
3. Enter a Package Name.
4. Select the Tables for which you want to generate a PL/SQL package.
5. Create the Package.
6. Review the results.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The steps to generate methods on tables are displayed in the slide.

Add Additional Business Rules to the API

- Computations
- Validations

```
insert into "PROJECTS" (
    "PROJECT_ID", "PROJECT_NAME", "PROJECT_TYPE", "PROJECT_DESCRIPTION" ,
    "PROJECT_STATUS", "PROJECT_PLANNED_START_DATE", "PROJECT_START_DATE", "
    PROJECT_PLANNED_END_DATE", "PROJECT_END_DATE", "PROJECT_UPGRADE_YN",
    "PROJECT_CREATED_BY"
) values (
    "PROJECT_ID", initcap("PROJECT_NAME"), "PROJECT_TYPE", "PROJECT_DESCRIPTION" ,
    "PROJECT_STATUS", "PROJECT_PLANNED_START_DATE", "
    PROJECT_START_DATE", "PROJECT_PLANNED_END_DATE", "PROJECT_END_DATE
    ", "PROJECT_UPGRADE_YN", "PROJECT_CREATED_BY"
);
```

```
if P_PROJECT_END_DATE <= P_PROJECT_START_DATE then
    raise_application_error(-20002, 'End date should be greater than start
date');
end if;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Table APIs allow you to add business rules to the package so that they can be executed on multiple pages in your application. You can add any computations, validations, or logic that is necessary before manipulating data in the database.

Jack wants to add computation and validation in the PTS application. He plans to add a validation in the PROJECT_API package to check if the project end date is greater than the project start date. He also adds an initcap computation to the project name.

Creating a Form that Uses the API

1. Create a Report and Form in a Table.
2. Change the conditions for the DML processes to Never.
3. Change the page items Source to:
 - Source Type: Null
 - Source Used: Only when current value in session state is null
4. Add an After Header process to run the GET procedure.
5. Add three On submit–After Computations and Validations processes:
 - UPD procedure: Based on Save button
 - DEL procedure: Based on Delete button
 - INS procedure: Based on Create button



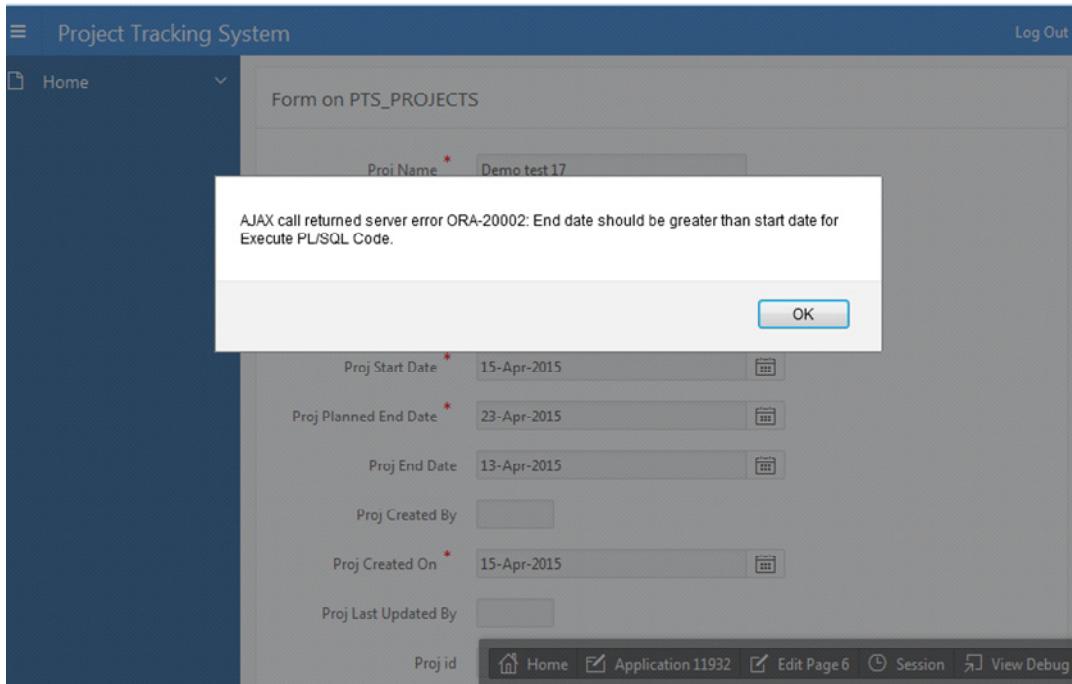
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When you create a form using the APEX wizards, DML processes to select the data and manipulate it (insert, update, delete) are created automatically. If you want to use an API on the table (you generated previously), you need to make the changes outlined in the slide. You will first create a form and change the DML processes to Never as you do not want the automatic DML fetch row and process row processes to be used. Next, you have to change the Source attributes so that values are not based on the database columns. You will then add an After Header process which uses the GET procedure from the API to load the values into the form. You will also add three On submit–After Computations and Validations processes by using the:

- UPD procedure from the table API to update the data in the database table when the Save button is clicked.
- DEL procedure from the table API to delete data in the database table when the Delete button is clicked.
- INS procedure to insert data into the database table when the Create button is clicked.

Jack creates a Report and Form based on the PTS_PROJECTS table. In the form, he sets the DML processes to Never as he does not want the automatic DML fetch row and process row processes to be used. He changes the Source attributes so that values are not based on the database columns. Then, he adds an After Header process which uses the GET procedure from the API to load the values into the form. He also adds three On submit–After Computations and Validations processes by using the INS, UPD, and DEL procedures.

Raising Errors From the API



ORACLE®

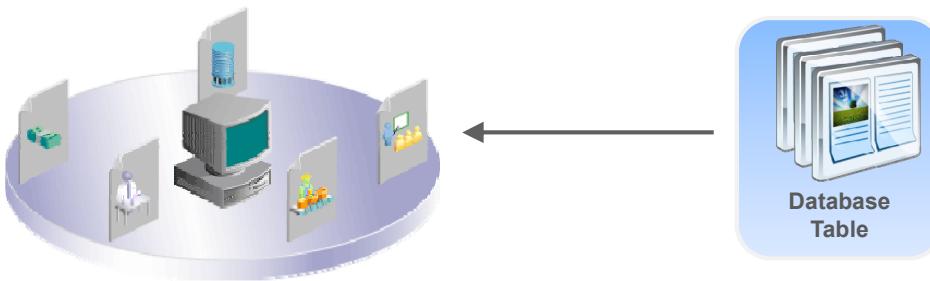
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can view how errors are raised by the API by violating one of the rules that you included in the Insert and Update procedure.

Case Scenario 2

In Jack's organization, there is a table that many applications are using currently. There are different business rules applied and handled by each application separately.

Using table APIs, Jack suggests building the packages so that business rules are consistent and manual workload is also reduced.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the scenario described in the slide, Jack's organization has a table that is used by many applications. Each application has different business rules. There is an extra overload of applying business rules each time the application accesses the same table.

Hence, Jack suggests using the package created using Table APIs, and adding common business rules to the package.

Case Scenario 3

XYZ is an e-commerce application developed using APEX. It needs to add the following business rules:

- Whenever the stock is below the reorder point, only good customers will have their order immediately processed.
- Good customers of a product are defined as those who have bought at least twice the average sales per customer over the last 12 months.
- Put the orders of bad payers on a waiting list, until they pay the amount due.

It needs to add the following computation:

- Increase the credit limit of the customers' credit card by 25% if the card usage is greater 5,000 per month.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the scenario described in the slide, XYZ is an APEX-based application. It wants to add the business rules and computation to the different pages which make use of the same tables. The company want these business rules to be added in a way that makes it consistent throughout.

To achieve this, XYZ application can use the Table APIs, and add the business rules and computation to the package which can be used throughout the application.

Quiz



Generating Methods on a Table provides for easy standardization across applications.

- a.** True
- b.** False

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Practice 7 Overview: Generating and Using Table APIs

This practice covers the following topics:

- Generating Methods on Tables
- Modifying the API with custom rules
- Creating a Form that uses the API



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Generate a method in a table
- Modify the API with custom rules
- Create a form that uses the API



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you should have learned how to generate a method in a table, modify the API with custom rules, and then use this API in a form.

Jack added a validation in the `PROJECT_API` package to check if the project end date is greater than the project start date. He also added an initcap computation to the project name.

He also created a Report and Form based on the `PTS_PROJECTS` table and used the `PROJECT_API`.

8

Creating and Using RESTful Web Services

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Using RESTful Services in PTS



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack wants Project Tracking System (PTS) applications to interact with one another over the web in a platform-neutral, language independent environment so that he can use RESTful Web Services to check weather conditions based on the employees' address.

Objectives

After completing this lesson, you should be able to:

- Create a RESTful Web Service
- Create an application that consumes a RESTful Web Service
- Invoke a RESTful Web Service using a Java client



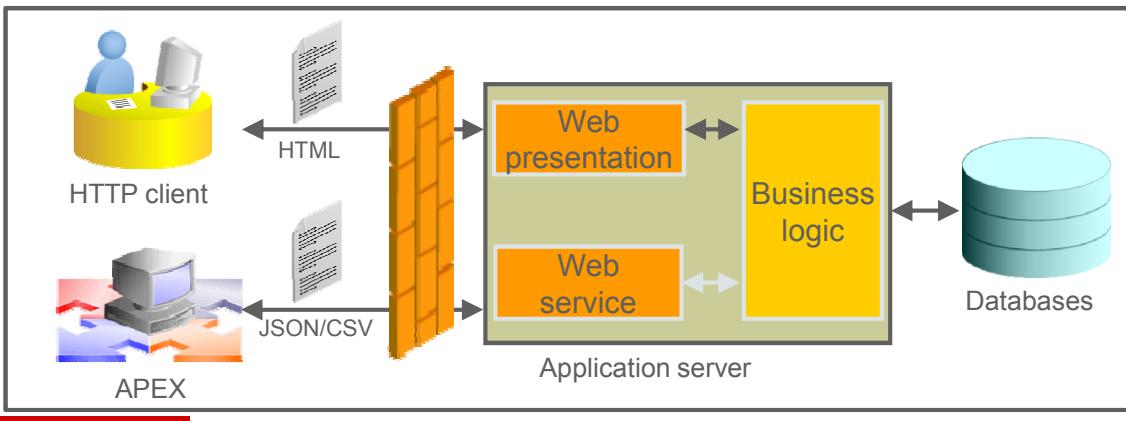
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learn about RESTful Web Services and its various components. You also create a new RESTful Web Service and consume the RESTful Web Service in your application.

What Is a Web Service?

- Web services enable applications to interact with one another on the web.
- The scope of RESTful Web Service is found in the:
 - URI
 - Service method that is described by the HTTP methods: GET, POST, PUT, and DELETE



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Web services enable applications to interact with one another over the web in a platform-neutral, language independent environment. In a typical web services scenario, a business application sends a request to a service at a given URL by using the protocol over HTTP. The service receives the request, processes it, and returns a response. You can incorporate calls with external web services in applications developed in Application Builder.

Web services are typically based on Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) architectures. The scope of the web service is found in the URI and the method of the service is described by the HTTP method that is used, such as GET, POST, PUT, and DELETE. RESTful Web Services are resource oriented.

The advantages of using REST are:

- REST messages are not blocked by firewalls because this protocol uses the HTTP protocol.
- REST requests do not require the overhead of XML, and SOAP envelopes and inputs are typically provided in the URI.

To use web services in Oracle Application Express, you create a web service reference using a wizard. Web service references can be based on a RESTful style.

What Are RESTful Web Services?

- An architecture standard for accessing information
- Built on HTTP
- Commonly used in dynamic languages, such as PHP
- RESTful Web Services have three main characteristics:
 - The services use HTTP methods explicitly.
 - The services are accessible through Uniform Resource Identifiers (URIs).
 - The services are stateless.
- Examples:
 - Creates a native mobile application using the same database as the corresponding APEX web application
 - Integrates with back office operations
 - Provides data persistence for a static single page web application



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

REST is the commonly used name for a set of architectural principals officially known as Representational State Transfer. RESTful services enable an application to publish web services that are accessed by other applications to exchange data. REST is an architecture style for designing networked applications. Rather than using complex mechanisms such as SOAP to connect machines, simple HTTP is used to make those connections.

For example, a RESTful service can be configured to return all employee names for a particular department. The data exchange for a RESTful service follows the REST architectural style.

When you create a RESTful service, you supply the necessary information about the structure of the request and response including:

- A URI identifying the RESTful request
- The HTTP method identifying the method of the web service
- HTTP Headers, if required, that are part of the request
- Parameters expected by the web service
- The type of input expected by the web service
- The format of the response and how to identify the response parameters

Advantages of RESTful Web Services

- Provide a common data access layer that can be accessed from any platform
- Simple
- Based on familiar HTML, GET, POST, PUT, and DELETE methods
- Widely used by Twitter, Netflix, Dropbox, PayPal, Flickr, and Amazon S3
- Stateless, which reduces overhead and complexity on the server
- Support caching and light weight
- Can be called through the web browser



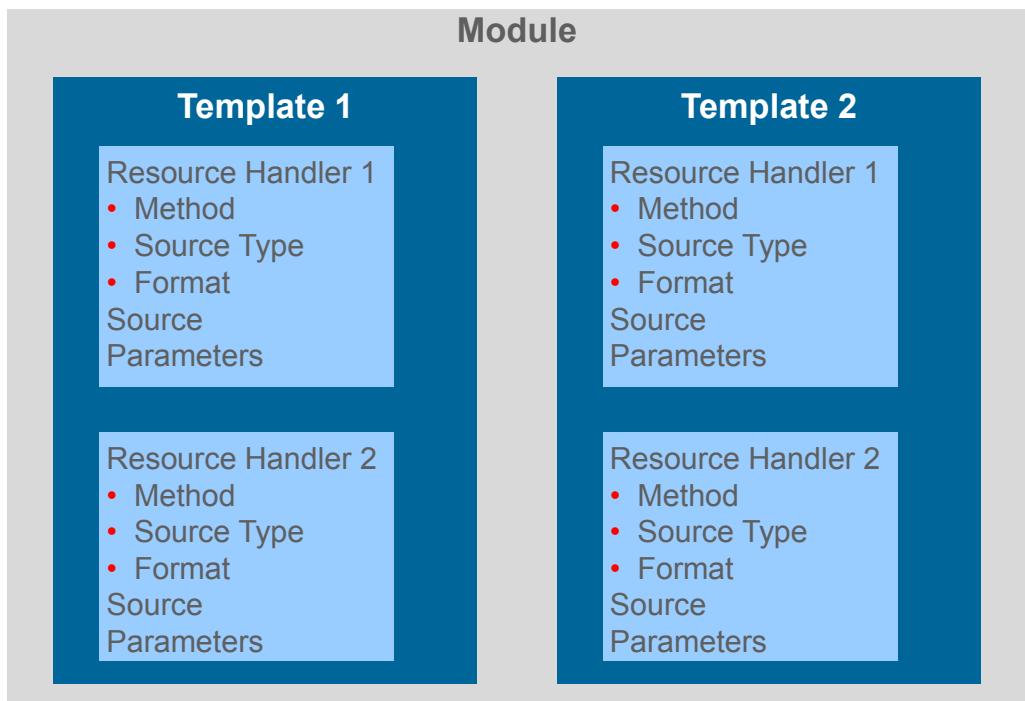
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Earlier, the predominant web service standard was SOAP. It was a widely used, but complicated web service. Requests and response were XML wrapped in SOAP Envelopes, whose structure was defined by one of the very large specifications, either 1.1 or 1.2.

Currently, RESTful Web Services are used everywhere. All the major players on the web like Google, Amazon, Yahoo, Netflix, and others have RESTful API's.

REST provides a powerful yet simple alternative to standards such as SOAP, with connectivity to virtually every language environment without having to install client drivers. This is because it is based on simple HTTP calls, which virtually all language environments support.

RESTful Web Service Components



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A module is a grouping of templates under a common URI prefix. A template can have multiple resource handlers that each provide a HTTP method to GET, POST, PUT, or DELETE data. Each resource handler contains a Source, Source Type, and Format that defines the source implementation for the selected HTTP method. For example, if the Source Type is set to Query and the Format is JSON, a SQL Query (from the Source item) executes and transforms the resultset into JavaScript Object Notation (JSON) representation. This option is only available when the HTTP method GET has been selected.

Method: Identifies the HTTP method to be used for the Resource Handler. The four options are GET, DELETE, POST, and PUT. Only one handler per HTTP method is permitted.

Source Type: Defines the source implementation for the selected HTTP method. Options include:

- **Query:** Executes a SQL Query and transforms the resultset into either a JSON or CSV representation, depending on the format selected. This option is only available when the HTTP method GET has been selected.
- **Query One Row:** Executes a SQL Query returning one row of data into a JSON representation. This option is only available when the HTTP method GET has been selected.

- **PL/SQL:** Executes an anonymous PL/SQL block and transforms any OUT or IN/OUT parameters into a JSON representation.
- **Feed:** Executes a SQL Query and transforms the results into a JSON Feed representation. Each item in the feed contains a summary of a resource and a hyperlink to a full representation of the resource. The first column in each row in the resultset must be a unique identifier for the row and is used to form a hyperlink of the form `path/to/feed/{id}`, with the value of the first column being used as the value for `{id}`. The other columns in the row are assumed to summarize the resource and are included in the feed. A separate Resource Template for the full representation of the resource should also be defined.
- **Media Resource:** Executes a SQL Query conforming to a specific format and turns the resultset into a binary representation with an accompanying HTTP Content-type header identifying the Internet media type of the representation.

Source: Identifies the SQL Query or PL/SQL block responsible for handling the selected HTTP method. Multiple resource handlers can be defined for a resource template. To define additional handlers, edit the newly created RESTful Service Module.

Parameters: Parameters to a resource handler can also be manually defined to bind HTTP headers to the resource handler, or to cast a URI template parameter to a specific data type. For example, a resource handler might need to know the value of the HTTP Accept-Language header to localize the generated representation.

ORDS and RESTful Web Services

ORDS performs the following tasks with respect to RESTful services:

- Request dispatching
- JSON generation for simple GET requests
 - Pagination
 - Lower cases column names
 - Omits null values
 - Generates JSON links
- Simple JSON parsing, form data parsing
- Exception and error handling and responses (HTML)

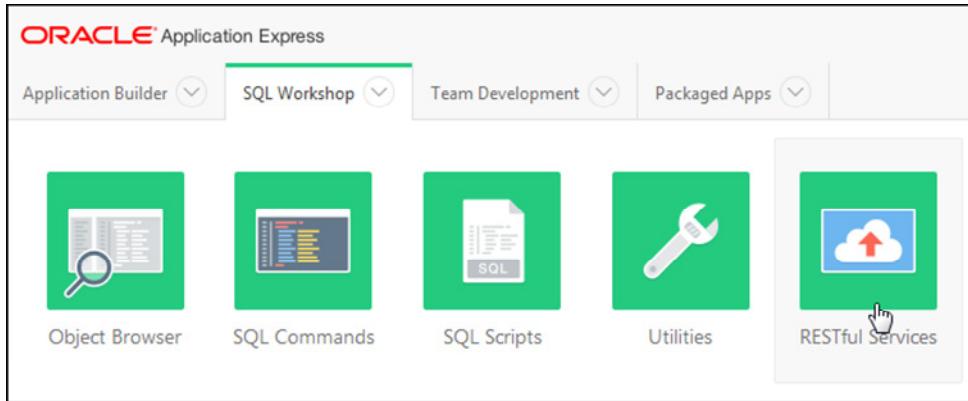


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle REST Data Services (ORDS) make it easy to develop modern REST interfaces for relational data in the Oracle Database and now, with ORDS 3.0, the Oracle Database 12c JSON Document Store and Oracle NoSQL Database. ORDS is available both as an Oracle Database Cloud Service and on premise.

ORDS performs the tasks given in the slide with respect to RESTful services.

Accessing RESTful Services



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can access a RESTful Web Service in APEX by performing the following steps:

1. Log in to the workspace home page
2. Click **SQL Workshop**
3. Click **RESTful Services**

Creating a RESTful Web Service

The screenshot shows the 'RESTful Services Module' configuration dialog. At the top right are 'Cancel' and 'Create Module' buttons. The main area contains several configuration sections:

- Name:** example_ws
- URI Prefix:** (empty)
- Origins Allowed:** (empty)
- Status:** Published
- Pagination Size:** 25
- Required Privilege:** - Assign Privilege -

Below these are sections for adding a resource template and a resource handler:

- Add a Resource Template:**
 - URI Template:** example_ws
 - Priority:** 0
 - Entity Tag:** Secure HASH
- Add a Resource Handler:**
 - Method:** GET
 - Source Type:** Query
 - Format:** JSON
 - Source:** select * from oe_hr_employees;

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You create a RESTful Web Service Module by specifying the Name, adding a resource template by specifying the URI template, and by adding a resource handler.

Creating a RESTful Web Service

The screenshot shows the Oracle Application Express interface for creating a RESTful service. On the left, a sidebar titled 'RESTful Service' lists two items: 'example_ws' (with a blue icon) and 'example_ws' (with a green icon). Below these are two buttons: 'GET' (highlighted in yellow) and '+ Create Handler' and '+ Create Template'. The main panel is titled 'Resource Handler' and contains the following configuration:

- RESTful Service Module:** example_ws
- URI Template:** example_ws
- Method:** GET
- Source Type:** Query
- Format:** JSON
- Requires Secure Access:** No
- Pagination Size:** [empty input field]

Below this is a 'Source' section with a code editor containing the following SQL query:

```
* Source  
select * from oe_hr_employees
```

At the bottom of the panel, there is a note: "To test the behaviour of the RESTful Service Handler, click the Test button below. If the REST parameters, click the Set Bind Variables button to set test values for the parameters. Before I have saved all changes to this page by clicking Apply Changes. For better results, ensure you installed in your browser." Below the note are two buttons: 'Test' and 'Set Bind Variables'.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A Resource Handler is a query or an anonymous PL/SQL block responsible for handling a particular HTTP method. Although multiple resource handlers can be defined for a resource template, only one resource handler per HTTP method is permitted. You specify the source for the resource handler.

Testing the RESTful Web Service

http://<hostname>:<port>/apex/teach/example_ws

```
{  
  - items: [  
    - {  
        employee_id: 198,  
        first_name: "Donald",  
        last_name: "OConnell",  
        email: "DOCONNEL",  
        phone_number: "+650.507.9833",  
        hire_date: "1999-06-21T04:00:00Z",  
        job_id: "SH_CLERK",  
        salary: 2600,  
        manager_id: 124,  
        department_id: 50  
      },  
      - {  
        employee_id: 199,  
        first_name: "Douglas",  
        last_name: "Grant",  
        email: "DGRANT",  
        phone_number: "+650.507.9844",  
        hire_date: "2000-01-13T05:00:00Z",  
        job_id: "SH_CLERK",  
        salary: 2600,  
        manager_id: 124,  
        department_id: 50  
      },  
      - {  
        employee_id: 200,  
        first_name: "Jennifer",  
        last_name: "Whalen",  
        email: "JWHALEN",  
        phone_number: "+515.123.4444",  
        hire_date: "1987-09-17T04:00:00Z",  
        job_id: "AD_ASST",  
        salary: 4400,  
        manager_id: 101,  
        department_id: 10  
      }  
  ]  
}
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You test the behavior of the RESTful Service Handler by clicking the Test button (shown in the previous slide). The slide shows the result in JSON format.

Adding a Bind Variable to the RESTful Web Service

The screenshot shows the Oracle Application Express interface for configuring a RESTful Web Service.

Resource Handler: A resource handler is a query or an anonymous PL/SQL block. Although multiple resource handlers can be defined, one is permitted.

RESTful Service Module: URI Template: example_ws, Method: GET, Source Type: Query, Format: CSV, Requires Secure Access: No, Pagination Size: [empty input field].

Source: * Source: select * from oechr_employees where department_id = :DEP

Resource Handler Parameter: Parameters to a resource handler can also be manually defined to bind HTTP headers to the resource handler or a URI template parameter to a specific data type. For example, a resource handler might need to know the value of the HTTP Accept-Language header in order to localize the generated representation.

RESTful Service Module: example_ws, Resource Handler: GET, Handler Source: select * from oechr_employees where department_id = :DEP.

Bind Variable Name: DEP, Access Method: IN, Source Type: HTTP Header, Parameter Type: String.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can add parameters to a resource handler to bind HTTP headers to the resource handler, or to cast a URI template parameter to a specific data type. A Bind Variable name identifies the parameter bind variable name used within SQL or PL/SQL.

RESTful Web Service: Examples

- Simple Query**
- Query with Parameter**
- PL/SQL**
- Feed**



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The examples mentioned in the slide are some of the source implementation types that can be used for an HTTP method type.

Jack creates an employee module that implements several methods to retrieve and display employee information from the employees' table. The employee module consists of several resource templates. Each resource template demonstrates a different way of retrieving information and formatting the returned results.

RESTful Web Service: Example

Simple Query

The screenshot shows the Oracle Application Express interface for creating a RESTful service. On the left, a sidebar lists three resource handlers under the 'Employees' module:

- Employees**: Contains a GET handler.
- employees/{id}**: Contains a GET handler.
- employeesfeed/**: Contains a GET handler.

On the right, a detailed configuration window for the 'employeesfeed/' resource is open:

- Resource Handler: GET** (Title)
- RESTful Service Module:** Employees
- URI Template:** Employees
- Method:** GET
- Source Type:** Query
- Format:** JSON
- Requires Secure Access:** No
- Pagination Size:** (empty input field)
- Source:** (Panel containing a code editor with the SQL query: `1 select * from employees`)

At the bottom of the slide, there is a red banner with the word "ORACLE".

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The simple query example shown in the slide executes a SQL Query and transforms the resultset into a JSON representation. This option is available for the GET method.

RESTful Web Service: Example

Simple Query JSON Results

<http://<hostname>:<port>/apex/teach/hr/empinfo/>

```
{
  "next": {"$ref": "https://apex.oracle.com/pls/apex/apexws2/Employees?page=1"}, "items": [
    {"employee_id": 505, "first_name": "Fiorello", "last_name": "LaGuardia", "email": "fiorello.laguardia@pts.com", "phone_number": "2125553923", "mobile_number": "1235342653", "address": "Hangar Center, Third Floor, Flushing, NY", "designation": "Senior Manager", "salary": 240000, "hire_date": "2014-06-08T00:00:00Z"}, {"employee_id": 504, "first_name": "Frank", "last_name": "O'Hare", "email": "frank.charie@pts.com", "phone_number": "773557693", "mobile_number": "3157862405", "address": "10000 West OHare, Chicago, IL", "designation": "Manager", "salary": 180000, "manager_id": 505, "hire_date": "2003-06-06T00:00:00Z"}, {"employee_id": 518, "first_name": "Turner", "last_name": "Thomas", "email": "turner.thomas@pts.com", "phone_number": "7642788982", "mobile_number": "1238767344", "address": "1234 Sacramento, CA", "designation": "Manager", "salary": 180000, "manager_id": 505, "hire_date": "2014-04-06T00:00:00Z"}, {"employee_id": 520, "first_name": "Rebecca", "last_name": "Mary", "email": "rebecca.mary@pts.com", "phone_number": "3157862401", "mobile_number": "3157862401", "address": "5623 W University Dr, Tempe, AZ", "designation": "Manager", "salary": 180000, "manager_id": 505, "hire_date": "2014-09-06T00:00:00Z"}, {"employee_id": 501, "first_name": "John", "last_name": "Dullies", "email": "john.dullies@pts.com", "phone_number": "703552143", "mobile_number": "7035558967", "address": "45020 Aviation Drive, Sterling, VA", "designation": "Developer", "salary": 100000, "manager_id": 504, "hire_date": "2009-01-01T00:00:00Z"}, {"employee_id": 502, "first_name": "William", "last_name": "Hartsfield", "email": "william.harstfield@pts.com", "phone_number": "4045553285", "mobile_number": "9873567899", "address": "6000 North Terminal Parkway, Atlanta, GA", "designation": "Senior Developer", "salary": 140000, "manager_id": 504, "hire_date": "2013-06-06T00:00:00Z"}, {"employee_id": 503, "first_name": "Edward", "last_name": "Logan", "email": "edward.logan@pts.com", "phone_number": "5175553295", "mobile_number": "5673459876", "address": "1 HarborSide Drive, East Boston, MA", "designation": "QA Engineer", "salary": 90000, "manager_id": 504, "hire_date": "2000-03-05T00:00:00Z"}, {"employee_id": 506, "first_name": "Albert", "last_name": "Lambert", "email": "albert.lambert@pts.com", "phone_number": "3145554022", "mobile_number": "3157862399", "address": "10701 Lambert International Blvd., St. Louis, MO", "designation": "Trainee", "salary": 50000, "manager_id": 518, "hire_date": "2015-01-01T00:00:00Z"}, {"employee_id": 507, "first_name": "Eugene", "last_name": "Bradley", "email": "eugene.bradley@pts.com", "phone_number": "3605551835", "mobile_number": "2349872345", "address": "Schoephoester Road, Windsor Locks, CT", "designation": "Trainee", "salary": 50000, "manager_id": 518, "hire_date": "2015-02-01T00:00:00Z"}, {"employee_id": 508, "first_name": "King", "last_name": "John", "email": "king.john@pts.com", "phone_number": "3242597586", "mobile_number": "3148762983", "address": "602 Scottsdale, AZ", "designation": "Developer", "salary": 100000, "manager_id": 518, "hire_date": "2013-10-02T00:00:00Z"}, {"employee_id": 509, "first_name": "Blaker", "last_name": "Joseph", "email": "blake.joseph@pts.com", "phone_number": "5231148976", "mobile_number": "2345432345", "address": "89 Camelback Road, Scottsdale, AZ", "designation": "Trainee", "salary": 50000, "manager_id": 518, "hire_date": "2005-03-11T00:00:00Z"}, {"employee_id": 510, "first_name": "Clark", "last_name": "James", "email": "clark.james@pts.com", "phone_number": "5672878364", "mobile_number": "9874569834", "address": "56234 New Blvd, NJ", "designation": "Developer", "salary": 100000, "manager_id": 518, "hire_date": "2010-08-08T00:00:00Z"}, {"employee_id": 511, "first_name": "Jones", "last_name": "Thomas", "email": "jones.thomas@pts.com", "phone_number": "1232347865", "mobile_number": "7896438734", "address": "Super Center, Second Floor, Tampa, FL", "designation": "Senior Developer", "salary": 140000, "manager_id": 520, "hire_date": "2014-06-09T00:00:00Z"}, {"employee_id": 512, "first_name": "Scott", "last_name": "Bren", "email": "scott.bren@pts.com", "phone_number": "3157862403", "mobile_number": "3157862403", "address": "2000 North Terminal Parkway, Atlanta, GA", "designation": "Trainee", "salary": 50000, "manager_id": 518, "hire_date": "2015-01-01T00:00:00Z"}]
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The slide shows the resultset when the Query Source Type and the JSON format option are selected.

RESTful Web Service: Example Query with a Parameter

The screenshot shows the Oracle Application Express interface for configuring a RESTful service module. On the left, there's a navigation tree under 'Employees' with various resource handlers like 'employees', 'employees/(id)', and 'employeesfeed/'. A specific 'Handler: GET' for 'employees/(id)' is selected. On the right, the 'Resource Handler: GET' configuration page is displayed. It shows the URI Template as 'employees/(id)', Method as 'GET', Source Type as 'Query One Row', and Requires Secure Access as 'No'. The Source section contains a SQL query: `select * from employees where employee_id = :id`. Below this, a 'Parameters' table is shown with one entry:

Name	Bind Variable Name	Access Method	Source Type	Parameter Type
<code>id</code>	<code>id</code>	IN	URI	String

. The Oracle logo is at the bottom left, and a copyright notice is at the bottom right.

The slide shows an example of a resource handler that uses Query One Row as a source type and passes a parameter called `id` to the resource handler.

RESTful Web Service: Example Query with Parameter Results

The screenshot shows a RESTful Web Service configuration in Oracle Application Express. The 'Source' tab contains the SQL query: 'select * from employees where employee_id = :id'. A 'Bind Variable' table below it shows a row for ':ID' with the value '505'. To the right, the results of the query are displayed as a JSON object:

```
{"employee_id":505,"first_name":"Fiorello","last_name":"LaGuardia","email":"fiorello.laguardia@pt.com","phone_number":"2125553923","mobile_number":"1235342653","address":"Hangar Center, Third Floor, Flushing, NY","designation":"Senior Manager","salary":24000,"hire_date":"2014-06-08T00:00:00Z"}
```

At the bottom left is the Oracle logo, and at the bottom right is the copyright notice: 'Copyright © 2015, Oracle and/or its affiliates. All rights reserved.'

The slide shows the resultset when a bind variable is passed to set values for the parameters.

RESTful Web Service: Example PL/SQL

The screenshot shows the Oracle Application Express interface. On the left, there's a sidebar with a tree view under 'Employees'. The 'employees' node has a 'POST' method selected. On the right, a 'Resource Handler:' dialog is open. It shows the configuration for handling a POST method on the 'Employees' resource template. The source code for the handler is a PL/SQL block:

```
1 declare
2   id employees.employee_id%TYPE;
3 begin
4   id := employees_seq.nextval;
5 end;
```

At the bottom of the dialog, it says 'Copyright © 2015, Oracle and/or its affiliates. All rights reserved.'

The PL/SQL example shown in the slide inserts a new row into a table and returns the newly created employee's ID in a bind variable. The employee ID is returned in the HTTP header. This Resource Handler uses the POST method.

RESTful Web Service: Example Feed

The screenshot shows the Oracle Application Express interface. On the left, there's a navigation tree under 'Employees' with items like 'employees', 'employees/{id}', and 'employeesfeed/'. The 'employeesfeed/' item has a 'GET' button highlighted with a cursor. On the right, a 'Resource Handler' configuration window is open. It shows the 'URI Template' as 'employeesfeed/' and the 'Method' as 'GET'. The 'Source Type' is set to 'Feed', and the 'Source' code is a SQL query: 'select employee_id, first_name from employees order by employee_id, first_name'. The Oracle logo is at the bottom left.

The slide shows an example of when feed is selected as the source type. It executes a SQL Query and transforms the results into a JSON Feed representation. Each item in the feed contains a summary of a resource and a hyperlink to a full representation of the resource.

RESTful Web Service: Example Feed Results

<http://<hostname>:<port>/apex/teach/hr/employeefeed/>

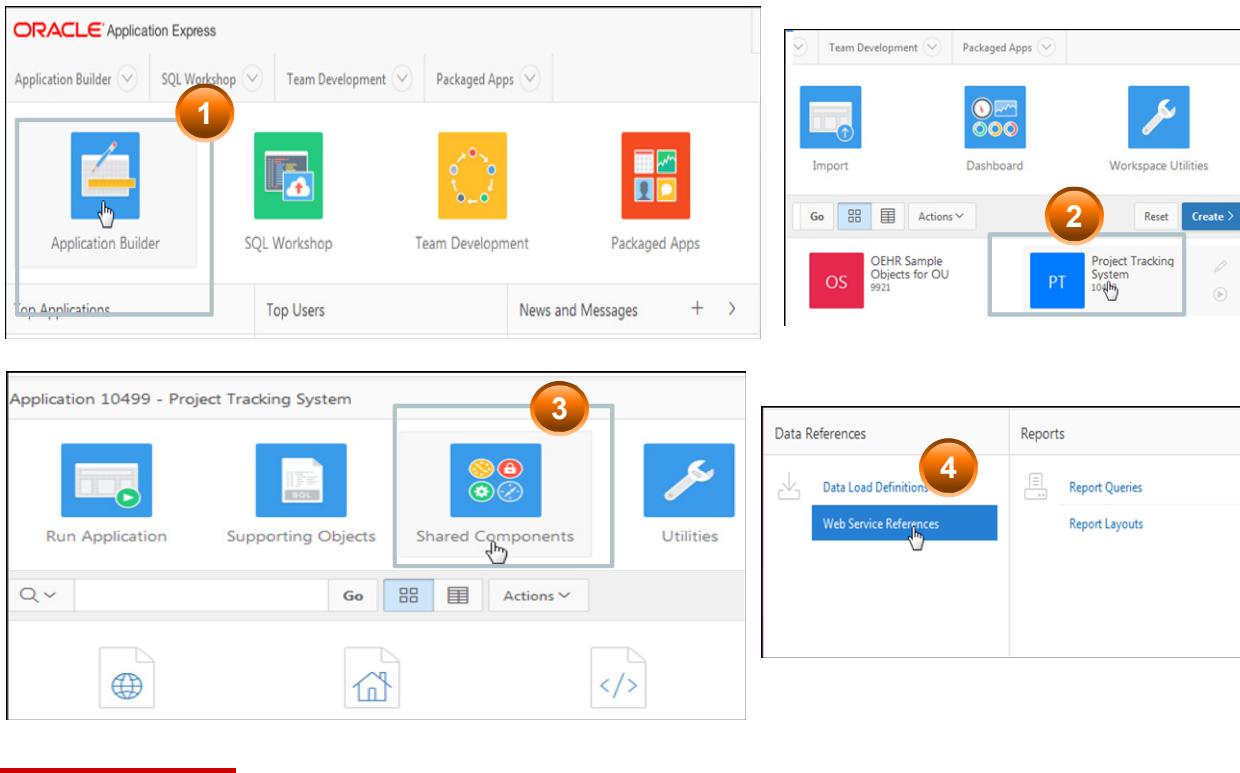
```
{"next":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/?page=1"},"items":[{"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/501"}, "employee_id":501, "first_name":"John"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/502"}, "employee_id":502, "first_name":"William"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/503"}, "employee_id":503, "first_name":"Edward"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/504"}, "employee_id":504, "first_name":"Frank"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/505"}, "employee_id":505, "first_name":"Fiorello"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/506"}, "employee_id":506, "first_name":"Albert"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/507"}, "employee_id":507, "first_name":"Eugene"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/508"}, "employee_id":508, "first_name":"King"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/509"}, "employee_id":509, "first_name":"Blake"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/510"}, "employee_id":510, "first_name":"Clark"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/511"}, "employee_id":511, "first_name":"Jones"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/512"}, "employee_id":512, "first_name":"Scott"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/513"}, "employee_id":513, "first_name":"Ford"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/514"}, "employee_id":514, "first_name":"Smith"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/515"}, "employee_id":515, "first_name":"Allen"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/516"}, "employee_id":516, "first_name":"Ward"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/517"}, "employee_id":517, "first_name":"Martin"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/518"}, "employee_id":518, "first_name":"Turner"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/519"}, "employee_id":519, "first_name":"Adams"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/520"}, "employee_id":520, "first_name":"Rebecca"}, {"uri":{"$ref":"https://apex.oracle.com/pls/apex/apexws2/employeesfeed/521"}, "employee_id":521, "first_name":"Miller"}]}
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This slide shows an example of feed results.

Accessing the Web Service Referencing Page



You manage web service references on the Web Service References page.

To access the Web Service References page:

1. On the workspace home page, click the **Application Builder** icon.
2. Select your application.
3. Click **Shared Components**.
4. Under **Data References**, click **Web Service References**.

Steps to Create and Consume a RESTful Web Service

1. Create a Database Application.
2. Create a Web Service Reference in your application.
3. Create a Form and Report page that uses the web service.
4. Test your application.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This slide shows you the steps to create and consume a RESTful Web Service.

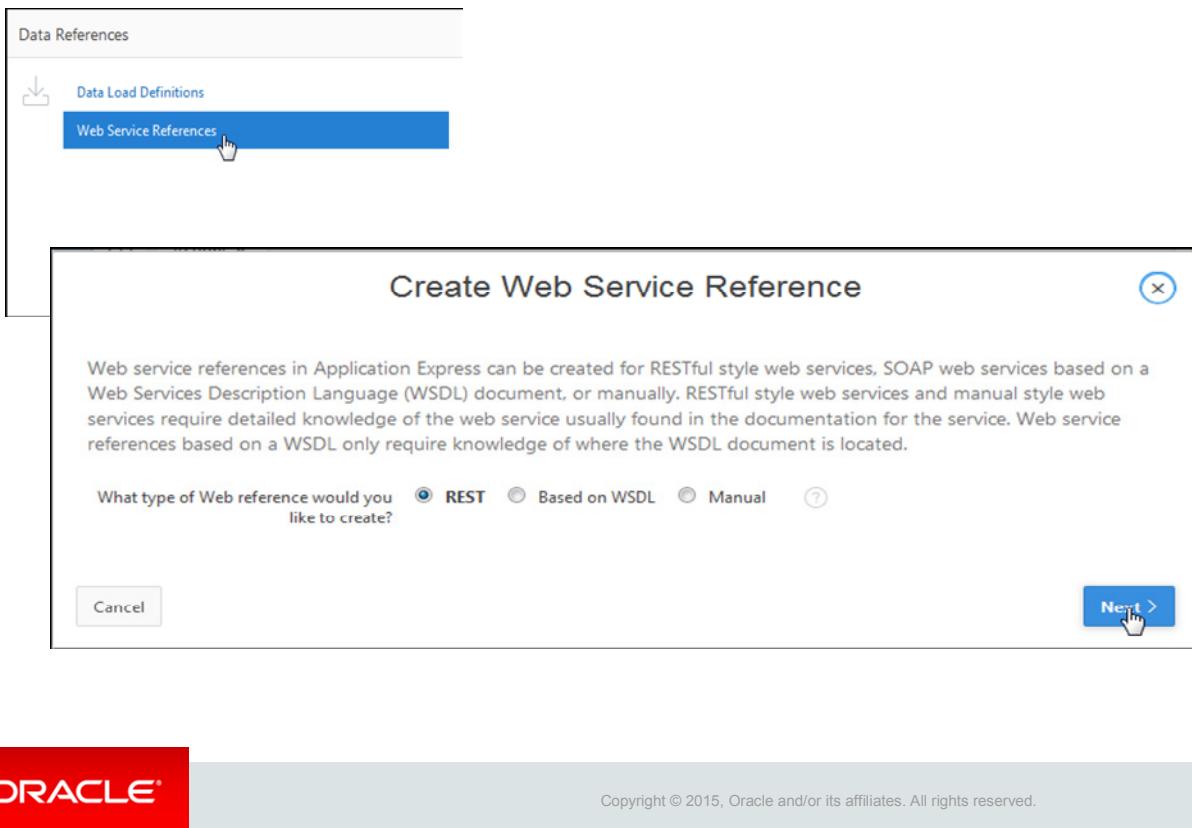
Jack will create a RESTful Web Service reference in the PTS application which returns weather condition results based on the location of the employees.

Step 1: Creating a Database Application

The screenshot shows the Oracle Application Express (APEX) interface. At the top, it displays "Application 102" and "Application 102 - PROJECT TRACKING SYSTEM". There are several navigation icons at the top right, including a plus sign, a magnifying glass, a house, a checkmark, and a refresh symbol. Below the title, there are five main buttons: "Run Application", "Supporting Objects", "Shared Components", "Utilities", and "Export / Import". Underneath these buttons is a search bar with a magnifying glass icon and a dropdown arrow, followed by a "Go" button and some other icons. To the right of the search bar is a "Create Page >" button. Below this row, there is a grid of page thumbnails. The first three thumbnails are labeled: "0 - Global Page - Mo...", "1 - Home", and "2 - Home". The fourth thumbnail starts with "3 - Project Status Re...". To the right of the thumbnails is a "Tasks" sidebar with a list of items, many of which are numbered (e.g., 14. Project Details, 13. Projects_List, 41. Enter Key Points, etc.). At the bottom of the interface, there is a red banner with the word "ORACLE" and a copyright notice: "Copyright © 2015, Oracle and/or its affiliates. All rights reserved."

PTS database application consumes the web service.

Step 2: Creating a Web Service Reference



You can then create a RESTful Web Service reference in the PTS application. This is done by selecting Web Service References in your application.

Step 2: Creating a Web Service Reference

Create REST Web Reference

REST Details

RESTful Web services rely on a simple resource-oriented architecture. The resource is identified by the URL and the method is described by the HTTP method. Inputs to the service are sometimes contained in the URL itself or in the HTTP payload. Inputs can also be read from HTTP headers sent with the request.

Application: 51492 Project Tracking System [?](#)

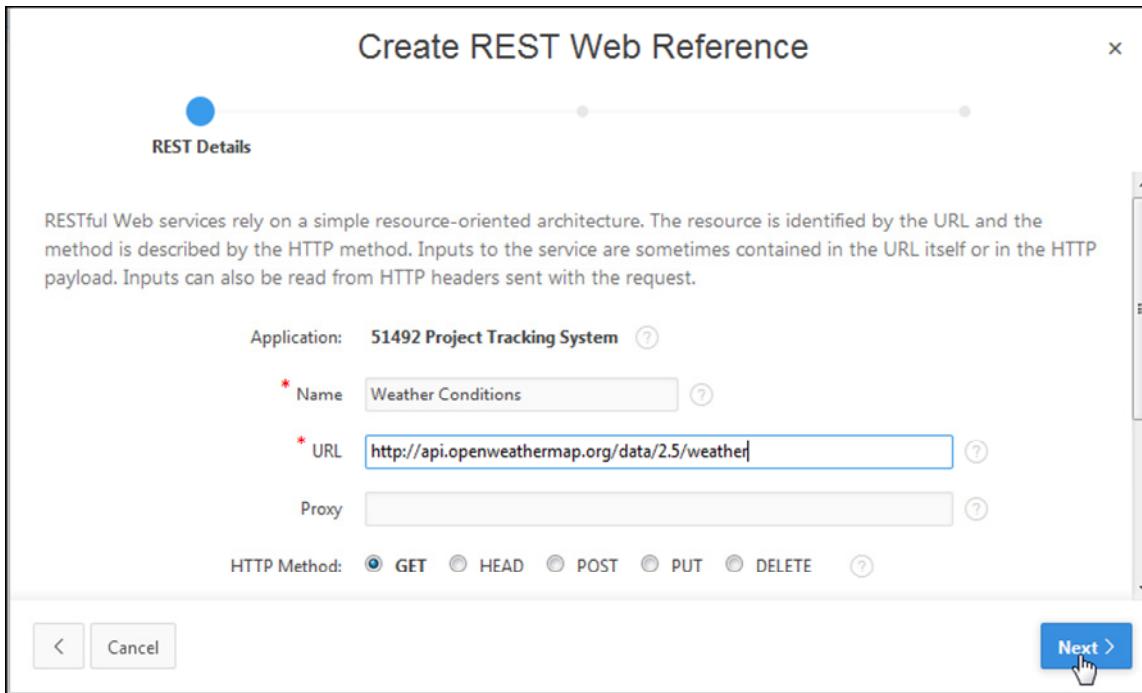
* Name: Weather Conditions [?](#)

* URL: <http://api.openweathermap.org/data/2.5/weather> [?](#)

Proxy: [?](#)

HTTP Method: GET HEAD POST PUT DELETE [?](#)

[<](#) [Cancel](#) [Next >](#)

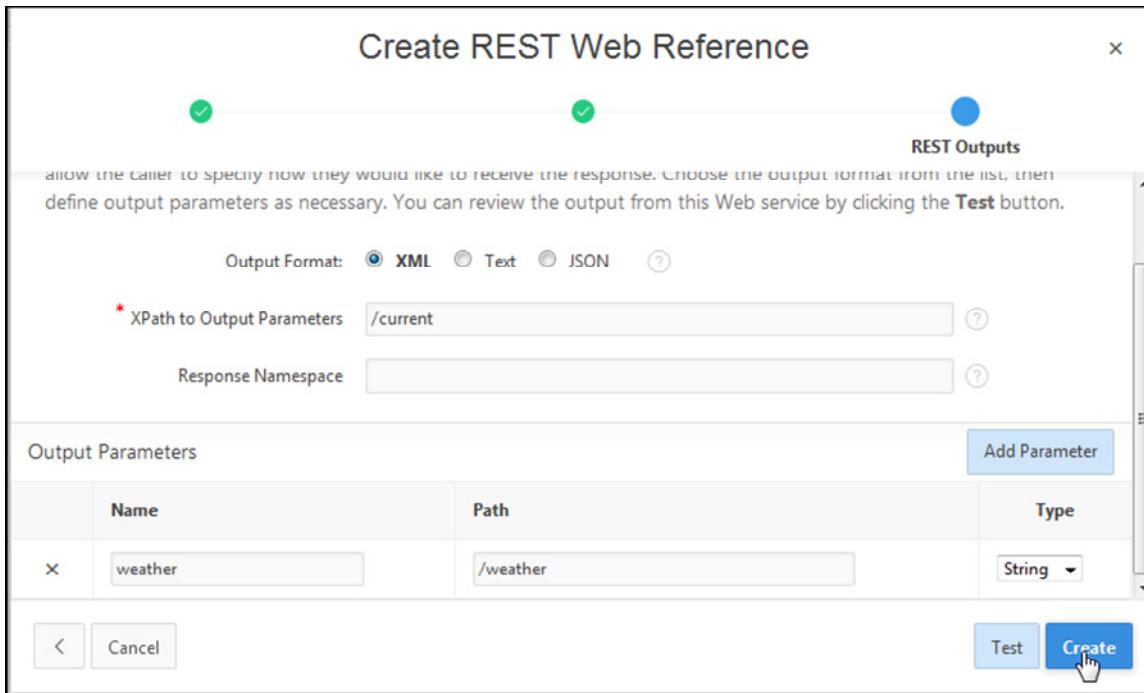


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can add a web service reference in your application by specifying the Name and URL. You can also define the HTTP header name that must be sent with the request.

Jack specifies the name as Weather Conditions and enters the URL as <http://api.openweathermap.org/data/2.5/weather>.

Step 2: Creating a Web Service Reference

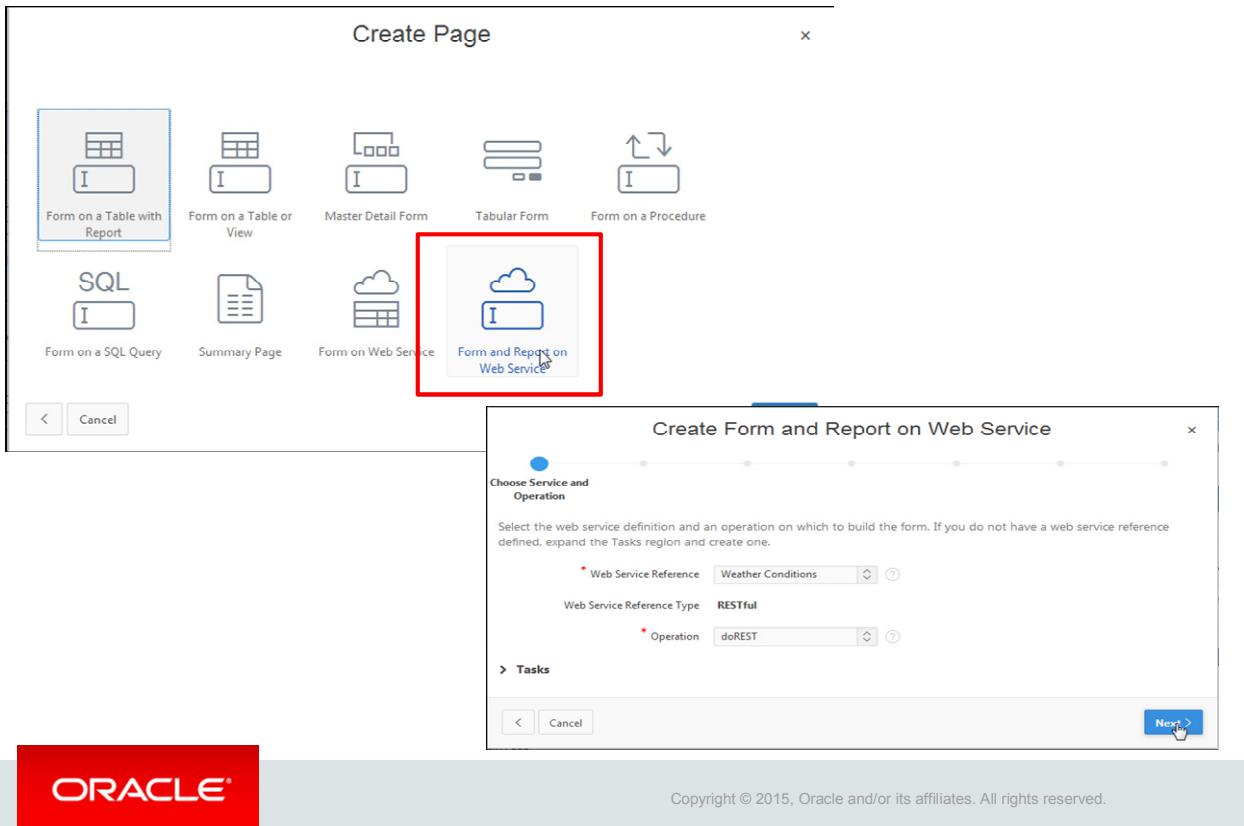


ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You specify the output format for the response. Select XML if you want the output type to be in XML format by specifying the Xpath. If you select Text, enter the name and path for the response parameters. The path is a number if the response is text.

Step 3: Creating a Form and Report on the Web Service



After creating the Web Service Reference to your application, you create a form and report on the web service. You select the web service reference and an operation on which you can build the form.

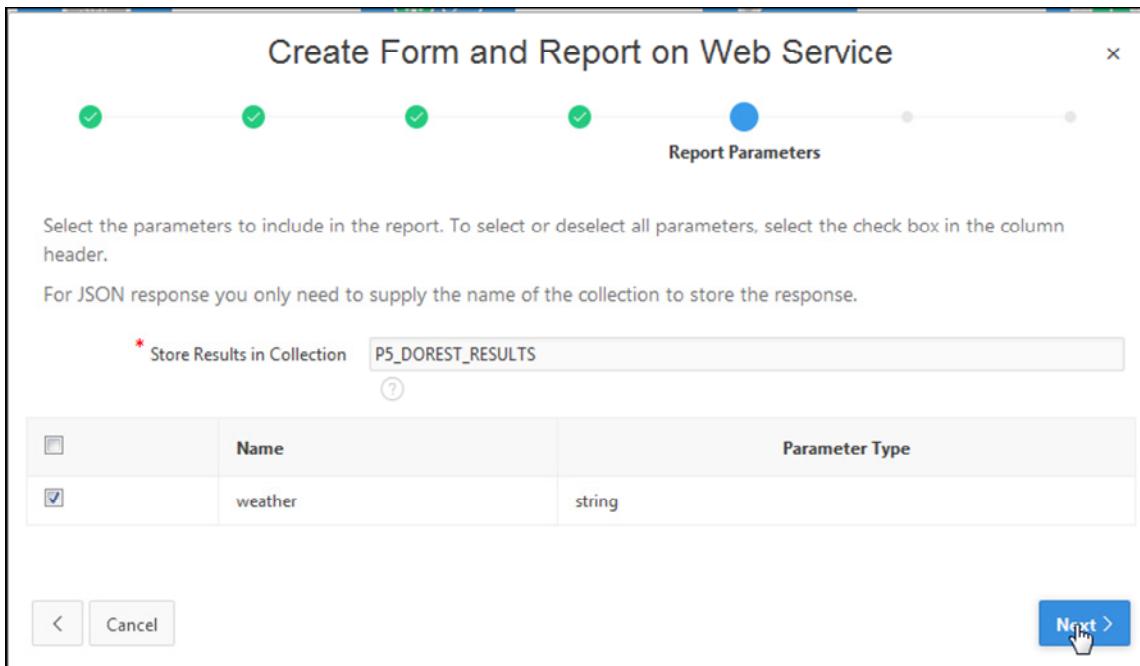
Step 3: Creating a Form and Report on the Web Service

The screenshot shows the 'Create Form and Report on Web Service' wizard in progress. The main window is titled 'Page and Region Attributes' and shows the configuration for a 'Weather Conditions' web service. The 'Operation' is set to 'doREST'. The 'Page' is '5', 'Page Name' is 'Weather Conditions', and 'Page Mode' is 'Normal'. The 'Form Region Title' is 'Employees' and the 'Report Region Title' is 'Results'. Both 'Form Region Template' and 'Report Region Template' are set to 'Standard'. A modal dialog box titled 'Input Items' is open, prompting the user to identify names and labels for input items based on service parameters. It lists two items: 'q' (string type, item name P5_Q, item label Location, Create Yes) and 'mode' (string type, item name P5_MODE, item label Mode, Create Yes). The 'Next >' button at the bottom right of the modal is highlighted with a mouse cursor.

Name	Type	Item Name	Item Label	Create
q	string	P5_Q	Location	Yes
mode	string	P5_MODE	Mode	Yes

Specify the page and region information. Identify the names and labels for items created based on the service's input parameters.

Step 3: Creating a Form and Report on the Web Service



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Select the parameters that you want to include in the report by selecting the check box.

Jack selects the weather parameter.

Step 4: Testing Your Application

The screenshot shows two views of the Project Tracking System. The top view is the 'Employees' page with fields for 'Location' (San Francisco) and 'Mode' (xml), and a 'Submit' button being clicked. The bottom view shows the results page with a 'Weather' section containing the text 'scattered clouds', which is highlighted with a red box. The Oracle logo is at the bottom left, and a copyright notice is at the bottom right.

Test your application by entering a value for the parameter if you have included a parameter in your web service.

Jack tests the application by entering the location as San Francisco and gets the weather condition.

Consuming Your APEX Web Service Using Java

The image shows two terminal windows side-by-side. Both windows have a title bar 'oracle@EDT3R28P0:~/labs/RESTemp' and a menu bar 'File Edit View Search Terminal Help'.
The left terminal window is labeled 'GET method' below it. It contains the following text:

```
[oracle@EDT3R28P0 Desktop]$ cd /home/oracle/labs/RESTemp  
[oracle@EDT3R28P0 RESTemp]$ ./run.sh 5 106  
The URI to your RESTful Web Service is http://10.150.21.221:8083/apex/ora01/.  
The request prior to the call: http://10.150.21.221:8083/apex/ora01/employees/10  
6  
HTTP/1.1 200 OK  
first_name : Valli  
hire_date : 1998-02-05T08:00:00Z  
phone_number : 590.423.4560  
email : VPATABAL  
manager_id : 103  
department_id : 60  
last_name : Pataballa  
salary : 4800  
employee_id : 106  
job_id : IT_PROG  
[oracle@EDT3R28P0 RESTemp]$
```


The right terminal window is labeled 'POST method' below it. It contains the following text:

```
[oracle@EDT3R28P0 RESTemp]$ ./run.sh 1  
The URI to your RESTful Web Service is http://10.150.21.221:8083/apex/ora01/.  
The request prior to the call: http://10.150.21.221:8083/apex/ora01/employees  
6  
HTTP/1.1 200 OK  
first_name : Valli  
hire_date : 1998-02-05T08:00:00Z  
phone_number : 590.423.4560  
email : VPATABAL  
manager_id : 103  
department_id : 60  
last_name : Pataballa  
salary : 4800  
employee_id : 106  
job_id : IT_PROG  
[oracle@EDT3R28P0 RESTemp]$  
Enter the first name: a  
Enter the last name: b  
Enter the email address: a@b.com  
Enter the hiredate (DD-MM-YYYY): 16-06-2013  
Enter the job id: SA REP  
HTTP/1.1 200 OK  
Employee id of new employee is 207  
[oracle@EDT3R28P0 RESTemp]$
```

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can also consume your APEX Web Service using a Java client. In this lesson, you run a Java program that can invoke the GET method to select an employee's details, or invoke the POST method to insert a new employee into the database.

Scenario 1: E-Business Suite

Have you ever wanted to:

- Exchange data between EBS and other custom systems within your organization
- Build a mobile application using data from your EBS system

What technology would you recommend to perform the following tasks?



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ERP system is typically the pivot point for information across your entire enterprise. Sometimes, it is the single source for Customer, Supplier, Item Master and Financial data. Most times, a database is hard to access for organizations, as a result of which redundancy of data is observed. These files are stored in other systems locally, leading to leakage of data. To safeguard your data, you can enable web services, which will directly lead to the truth and reduce data redundancy.

Scenario 2: Accessing the PTS Data

Jack has created a security application in his workspace using OEHR schema. Now he has a requirement to access the employees' data from the PTS database and wants to create a page in the security application which has the following requirements:

- It should display information about all employees in the PTS system.
- For each employee, it should display information such as employee ID, name, and login time.
- It should be able to create or update the employee information.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the scenario described in the slide, Jack wants to access employees' data from the PTS database and display employees' information in the security application. The security application which he has created is based on the OEHR schema.

He needs to display, create, and update employees' details in the security application.

To achieve this, he can create a RESTful service module called employees and then create a web service in the security application. After the web service is created, he can create a form and report on a web service to consume the web service.

Practice 8 Overview: Creating and Using RESTful Web Services

This practice covers the following topics:

- Creating a RESTful Web Service
- Creating an application that consumes the RESTful Web Service



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Create a RESTful Web Service
- Create an application that consumes a RESTful Web Service
- Invoke a RESTful Web Service using a Java client



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learned how to create a RESTful Web Service and consume it in an application.

Jack has used the RESTful Web Service reference in the PTS application which returns weather condition results based on the location of employees.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

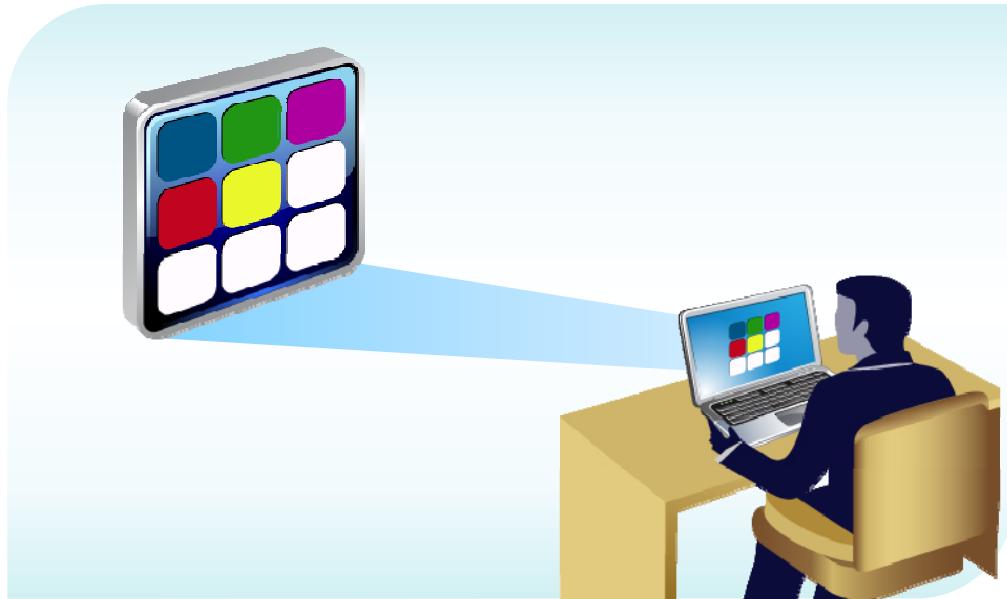
9

Using Templates and Themes

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Using Templates and Themes in PTS



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack has almost completed creating the PTS application. Now he wants to change the look and feel of the application. He explores the various themes available in APEX that he can use to enhance the user experience of the PTS application.

Though there are various themes available in APEX, he wants to change some specific components of those themes and templates, and then use it for his application. Jack decides to explore how he can modify the existing themes and templates for his application.

Objectives

After completing this lesson, you should be able to:

- Explain the difference between the various types of applications (Desktop, Mobile, and Responsive Design)
- Explain how substitution strings work within a template
- Create a new theme and use it in an application



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learn about the different types of applications that can be developed in APEX. You learn about the various theme types supplied by APEX and the templates that can be used within each application type. You understand why substitution strings are used in a template and understand the various substitution strings that are used within a template. You also learn about creating a customized theme and using it in an application, which will then be called a master application.

Types of Applications

- Desktop
- Mobile
- Responsive design



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Optimal user interface design principles have continually changed and evolved as new architectures are introduced. The old mainframe applications screen design was based on keyboard input only and the order of items on a page was critically important. However, with client/server applications, user interface design generally placed multiple items on a single line and everything had to fit into a single screen with no scrolling. With web applications, the principle design changed again to very few items on a single line, instead using vertical scrolling as required. Now, with mobile and tablet applications, the user interface principles have changed again to display minimal information with very easy to navigate screens that use touch instead of pointing devices.

Mobile applications are those designed to run specifically on smartphone devices. To facilitate this, applications need to be built with extremely minimal, semantic HTML that is optimized for mobile connections. Oracle Application Express 5.0 incorporates jQuery Mobile to allow developers to rapidly build mobile applications. jQuery Mobile-based web pages use a framework that was specifically built for mobile devices. jQuery Mobile is a lightweight framework that enables you to create compact and minimal websites that include only a few images and CSS files.

jQuery Mobile is also aware of and able to respond to mobile device-specific events, such as orientation change and touch events. The look and feel of jQuery Mobile-based applications can be easily modified via CSS using tools such as Theme Roller, thanks to its relatively static HTML structure.

Mobile applications developed with Application Express are browser-based applications that run inside the browser on the mobile device. Therefore, these applications must have a connection to communicate with the Oracle Database and cannot operate in a disconnected environment. Using jQuery Mobile, these applications can run on any mobile operating system, including iOS, Android, Blackberry, and Windows. Mobile devices that have HTML5 capabilities will be able to use all of the capabilities that can be built into the applications, including HTML5 date-pickers, sub-types that display different keypads based on field definition, and so on. Older devices will still render the application but with less advanced features. The major advantage of developing browser-based applications is that you only need to develop them once for the majority of mobile devices. However, one major limitation is accessing on-device features, such as contact lists. This limitation can be alleviated by integrating with solutions such as PhoneGap that allow for creating hybrid solutions, which use a native application wrapper to display the web application.

One of the recent user interface design principles being adopted is responsive design. Responsive design is a way to design websites so that the layout fits the available space on differently sized desktop or laptop browsers, as well as tablets and smartphones. On larger screens, the user gets the full experience. On smartphones and tablets, the layout adapts itself to the size of the device's screen. This is done by having certain elements shift position, resize, or become hidden entirely. The goal is to make all essential content available in a user-friendly and pleasing way on any device. Oracle Application Express 5.0 introduces a responsive user interface theme--Theme 42. This theme provides the building blocks and templates to achieve a responsive layout. However, it is up to the developer to use those templates and arrange the page content in such a way that the end result is truly responsive. It is not just a matter of picking one set of templates versus another set of templates. Also note that converting an existing application to Theme 25 requires you to review the pages of the application to ensure that the appropriate template and layout are defined.

User Interface

- Default
- Auto Detect
- Global Page

The screenshot shows the 'User Interfaces' tab selected in the top navigation bar of the Oracle Application Express interface. The main content area displays a table of user interfaces for 'Application 102'. The table has columns for Name, Type, Default, Auto Detect, Global Page, Theme, and Theme Style. Two rows are present: one for 'Desktop' (Type: monitor icon, Default checked, Global Page: 15, Theme: Universal Theme - 42, Theme Style: Vita) and one for 'Mobile' (Type: smartphone icon, Default checked, Global Page: 0, Theme: Mobile - 51, Theme Style: Default). Buttons for 'Cancel' and 'Apply Changes' are visible at the top right.

Name	Type	Default	Auto Detect	Global Page	Theme	Theme Style
Desktop	monitor icon	✓		15	Universal Theme - 42	Vita
Mobile	smartphone icon	✓		0	Mobile - 51	Default

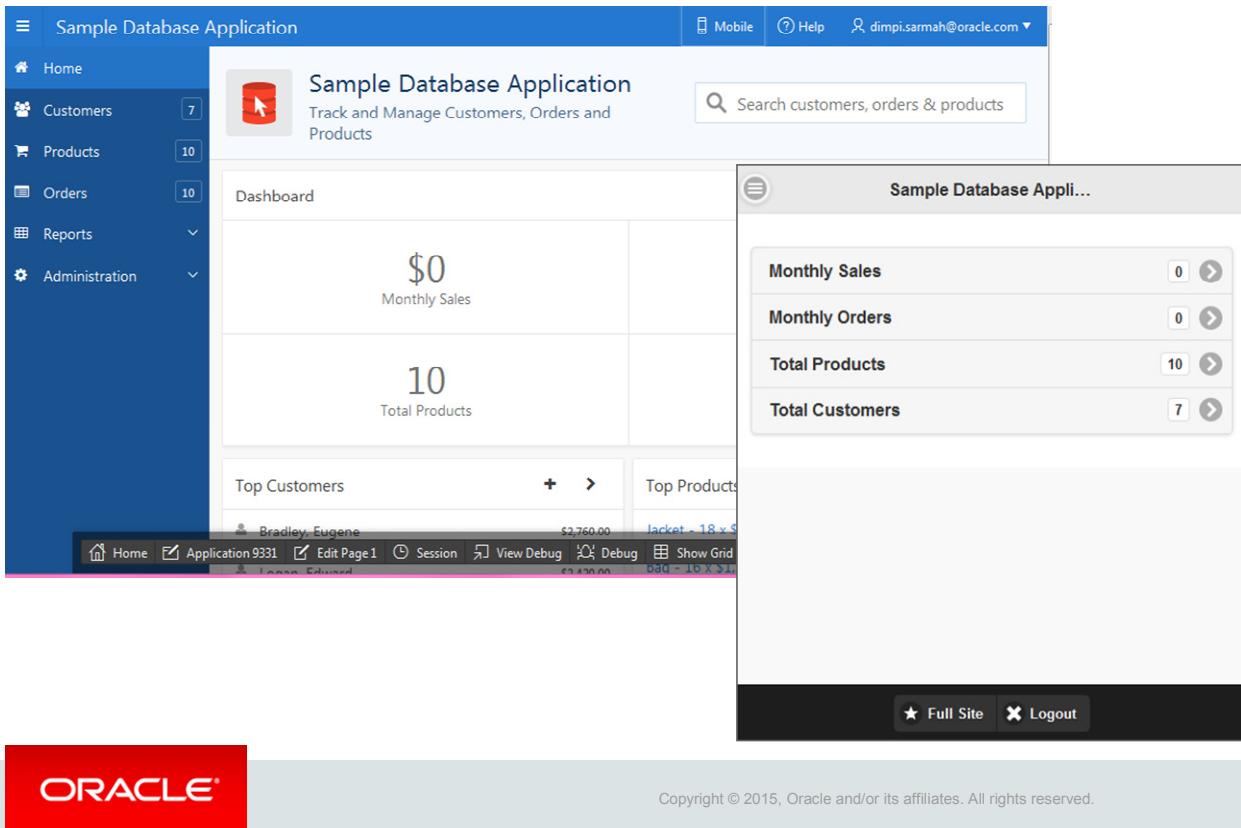
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Application Express 5.0 introduces the ability to declaratively define mobile applications using jQuery Mobile. The Application Builder has been enhanced to support multiple user interfaces, both desktop and mobile, within a single application. Therefore, developers can choose to build an application only for desktop, mobile, or for both. Each user interface defined for an application has its own theme, login page, home page, and global page (formerly Page 0). When defining a page within an application, you must specify which user interface is associated with that page. Based on the user interface selected, the available page and region types vary, and the options available through the create page and region wizards vary accordingly. For example, maps, data loading, and other page types and interactive report regions are only supported on desktop pages. Similarly, on mobile pages, the most common construct is list and form, instead of report and form.

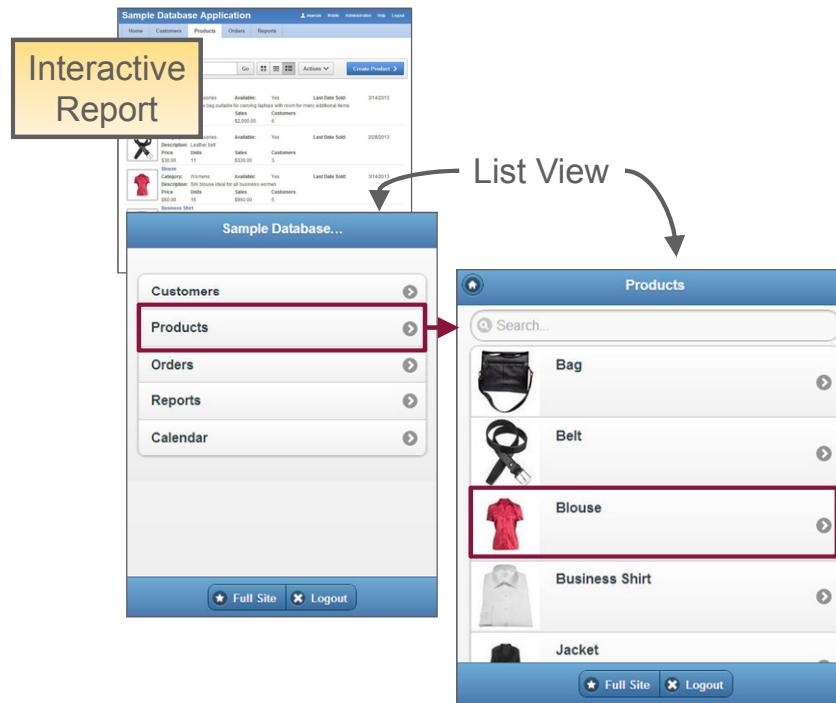
Desktop Versus Mobile Applications

Home Page



In the Sample Database Application, you can view the Home page of the Desktop UI and also click the Mobile link in the Navigation Bar to see the jQuery Mobile UI version. Notice that the user interface looks completely different.

Desktop Versus Mobile Applications Reports



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the Sample Database Application, on the Products tab, you see a detailed view of Interactive Report on the desktop. If you review the equivalent page in the mobile version, you see a List View of products that you can then drill down to another list view to see the details.

Desktop Versus Mobile Applications

Forms

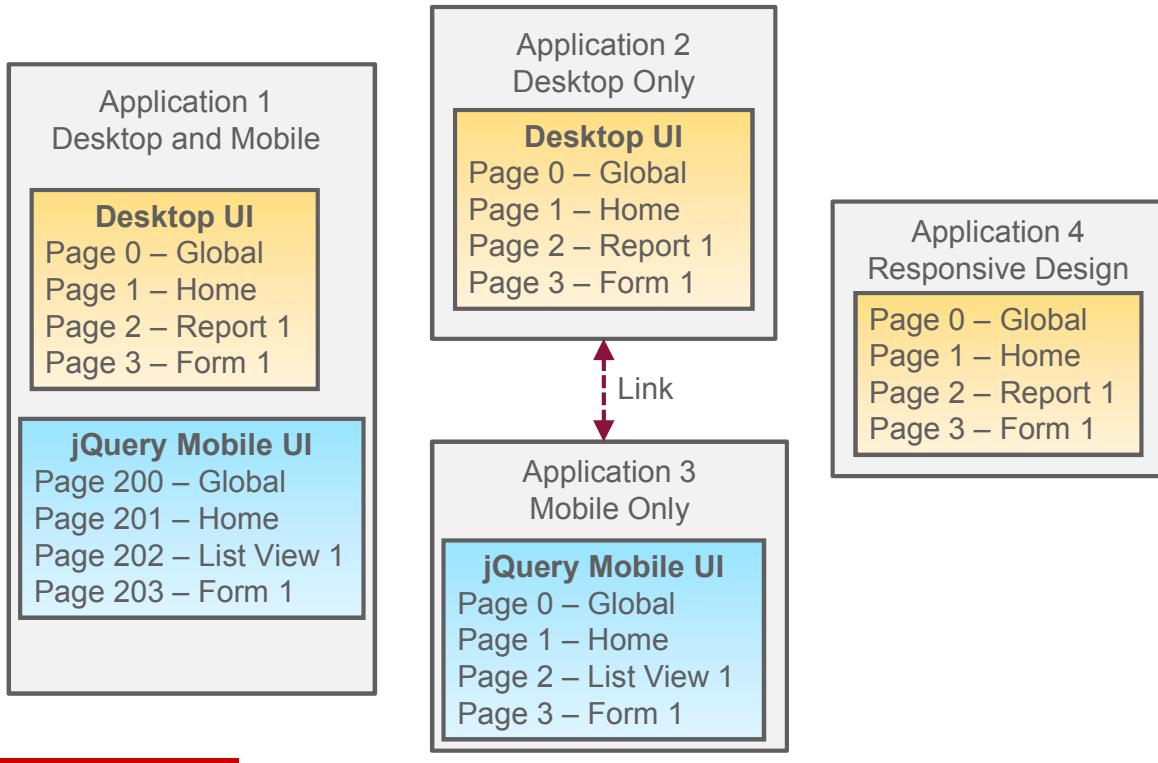
The image shows two side-by-side forms for maintaining a product. On the left is a desktop form titled 'Product Details' with fields for Product Name (Bag), Product Description (Unisex bag suitable for carrying laptops with room for many additional items), Category (Accessories), Product Available (Yes), List Price (125), Tags, and a Product Image preview. On the right is a mobile form titled 'Maintain Product' with similar fields, but the select list for Category is shown as a scrollable list instead of a standard dropdown. Both forms include 'Delete' and 'Apply Changes' buttons at the bottom.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this case, the forms are only slightly different. The mobile form uses jQuery Mobile item types and will use the mobile widget whenever possible. For example, for a select list, it will use the select list widget to show the list instead of a drop-down list as in the desktop version.

Application Options



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are many options when determining which type of application to create. Some of the possible ones are:

- Creating one application that contains both user interfaces: Desktop and jQuery Mobile. For this type of application, it is recommended that you separate the page types by using page numbers. For example, all the Desktop UI pages start with the Global page (0) and the Global page for the jQuery Mobile pages start with page 200.
- Create a Desktop Only application with a link to the jQuery Mobile Only application.
- Create a jQuery Mobile Only application with a link to the Desktop Only application.
- Create a Desktop UI application that uses Theme 25. Both Mobile and Desktop use the same application. Using Grid Layout and CSS3 Media Queries, the application will adjust automatically from one page.

Types of Themes

- Universal Theme - 42
- Mobile Theme - 51
- Custom: Themes you create
- Legacy: Old themes before APEX 4.0

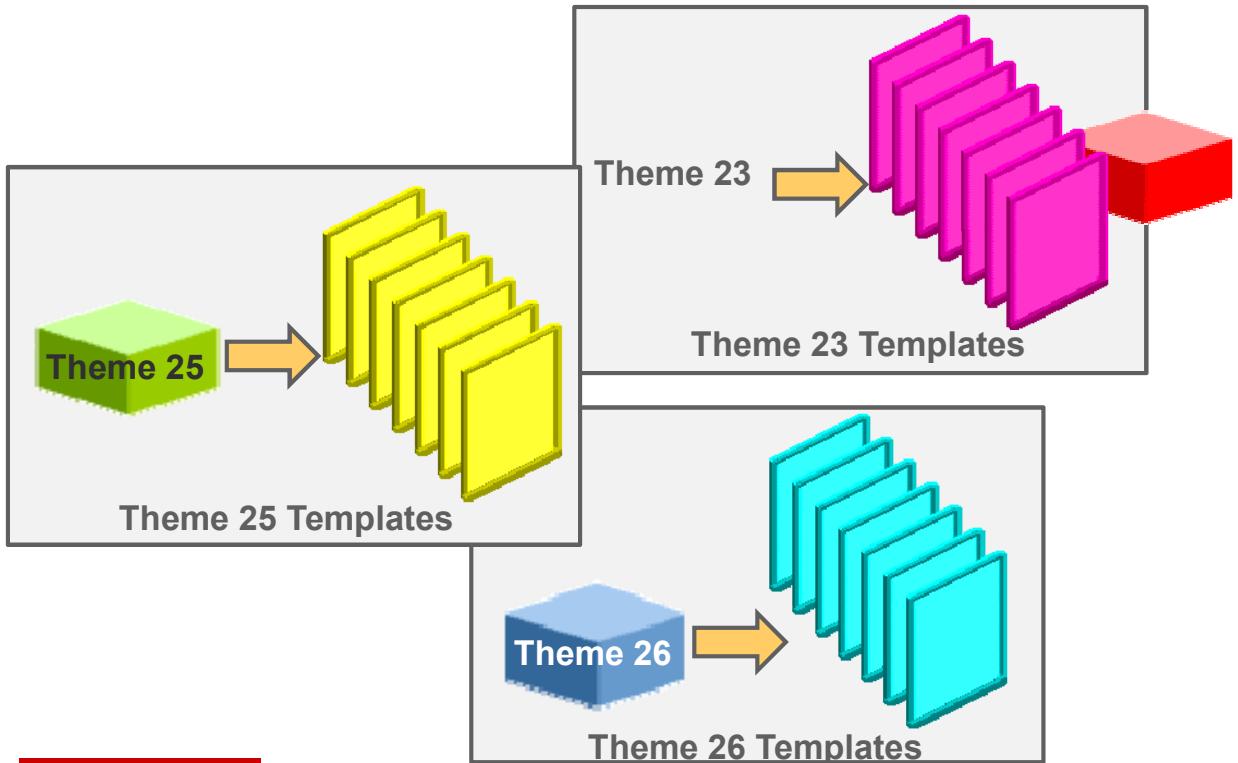


ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

With APEX 5.0, it is common to use one of the standard themes: 20-26. Theme 21 and 22 are older themes, and Theme 23, 24, and 26 are HTML5 based. Theme 25 is also HTML5 based and is the responsive design theme. Mobile Theme 51 is used when creating a Mobile page. In addition, you can also create your own themes, discussed later in this lesson. The themes (1-20) provided in previous releases of APEX are given. However, you should migrate your application to a later theme wherever possible to take advantage of the latest capabilities in APEX.

APEX-supplied Themes and Templates



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are 26 desktop themes supplied with Oracle APEX. A theme consists of several templates that determine how your page will look.

What Are Templates?

- Templates define how the pages or the page components of an application are displayed.
- You can use the following templates:
 - Page
 - Region
 - Report
 - Label
 - List
 - Button
 - Breadcrumb
 - Calendar
 - Pop-up



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When you build applications by using Oracle APEX, you use the default or predefined templates to build application components, such as pages, regions, reports, items, menus, and lists. Apart from using the available templates, you can also customize the look and feel of the application by modifying the existing templates or creating new templates by using HTML and cascading style sheets (CSS).

Templates facilitate the separation of business logic from the user interface. The developers of your organization can focus on the code for the business logic, while the graphic artists can concentrate on the look and feel.

The advantages of using templates are:

- Multiple components of your application can use the templates.
- To incorporate any change in the component, a single change to the template is sufficient.

Types of Templates

The screenshot shows the Oracle APEX application builder interface. The top navigation bar includes Application Builder, SQL Workshop, and Team Development. Below this is a breadcrumb trail: Application 102 > Shared Components > Templates. The main content area has tabs for Templates, Subscription, Publish, Utilization, and History, with Templates selected. There is a search bar and a 'Go' button. The main table lists various template types: Theme, Breadcrumb, Button, Label, Legacy Calendar, List, and Page. Each row includes columns for Name, Subscribed From, Subscribers, References, Updated, Updated By, Default, Preview, and Theme. The 'Page' row is currently selected, showing it is a 'Button' type with 'Button, Alternative 2' as its name. The Oracle logo is at the bottom left, and a copyright notice is at the bottom right.

Type	Name	Subscribed From	Subscribers	References	Updated	Updated By	Default	Preview	Theme
Theme				4					4
Theme				1					1
Breadcrumb		Theme		1					
Button		Theme		0					
Label		Theme		1					
Legacy Calendar		Theme		0					
List		Theme		0					
Page	Button, Alternative 2	Theme		0					
	Button,	Theme		0					

Oracle APEX offers the following types of templates:

- **Page:** Each page in your application is rendered using a page template. Page templates control the appearance of navigation bars, parent tabs, and standard tabs.
- **Region:** Region templates control the appearance of regions. Region templates can also be used to position buttons and region titles.
- **Label:** Label templates control the appearance of item labels (for example, field labels).
- **List:** List templates control the appearance of lists. For example, you can create a list to add a list of icons on a home page or a third-level tab, or to include a progress indicator.
- **Pop-up List of Values:** Pop-up templates control the display of pop-ups.
- **Calendar:** Calendar templates control the display of calendars.
- **Breadcrumb:** Breadcrumb templates control the display of breadcrumbs.
- **Button:** Button templates enable application developers to customize the look and feel of a button.
- **Report:** Report templates control the format of the report results.

Using Substitution Strings in Templates

A substitution string:

- Is a defined character string
- Is replaced by an object at run time
- Must be in uppercase
- Begins and ends with a pound (#) symbol

Example: #TITLE# is a substitution string that is replaced with the title text at run time.

Definition
<pre>Template <section class="uRegion #REGION_CSS_CLASSES# clearfix" id="#REGION_STATIC_ID#" #REGION_ATTRIBUTES> <div class="RegionHeading"> <h1>#TITLE#</h1> #CLOSE##PREVIOUS##NEXT##DELETE##EDIT##CHANGE##CREATE##CREATE2##EXPAND##COPY##HELP# </div> <div class="uRegionContent clearfix"> #BODY# </div> </section></pre>

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A substitution string is a defined character string that is replaced by an object at run time. Substitution strings used within a template must be in uppercase and begin and end with a pound (#) symbol. For example, in a region template, the #TITLE# substitution string is replaced with the title of the region, and the #BODY# substitution string is replaced with the region source at run time. The region source can be static HTML, a report, or form fields. At run time, the Oracle APEX engine replaces these strings with values, other objects, or null values.

If you are familiar with HTML, you can use HTML and, optionally, define some style definitions to customize your reports.

Page Template: <head>

```
<!DOCTYPE html>
<meta http-equiv="x-ua-compatible" content="IE=edge" />

<!--[if lt IE 7]><html class="no-js lt-ie10 lt-ie9 lt-ie8 lt-ie7" lang=&#34;&BROWSER_LANGUAGE.&quot;> <![endif]-->
<!--[if IE 7]><html class="no-js lt-ie10 lt-ie9 lt-ie8" lang=&#34;&BROWSER_LANGUAGE.&quot;> <![endif]-->
<!--[if IE 8]><html class="no-js lt-ie10 lt-ie9" lang=&#34;&BROWSER_LANGUAGE.&quot;> <![endif]-->
<!--[if IE 9]><html class="no-js lt-ie10" lang=&#34;&BROWSER_LANGUAGE.&quot;> <![endif]-->
<!--[if gt IE 9]><!--> <html class="no-js" lang=&#34;&BROWSER_LANGUAGE.&quot;> <!--<![endif]-->
<head>
    <meta charset="utf-8">
    <title>#TITLE#</title>
    #APEX_CSS#
    #THEME_CSS#
    #TEMPLATE_CSS#
    #THEME_STYLE_CSS#
    #APPLICATION_CSS#
    #PAGE_CSS#
    #FAVICONS#
    #HEAD#
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no"/>
</head>
<body class="t-PageBody t-PageBody--hideLeft t-PageBody--hideActions no-anim #PAGE_CSS_CLASSES#" #ONLOAD#
id="t_PageBody">
#FORM_OPEN#
<header class="t-Header" id="t_Header">
    #REGION_POSITION_07#
    <div class="t-Header-branding">
        <div class="t-Header-controls">
            <button class="t-Button t-Button--icon t-Button--header t-Button--headerTree" id="t_Button_navControl"
type="button"><span class="t-Icon fa-bars"></span></button>
        </div>
    ...

```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the Header area of the Page Template, you see the HTML tags that are used when presenting the Header of the HTML page. In addition, you see a series of substitution strings that get resolved during run time. Typically, you never change the header template but it may be useful to know that there are certain substitution strings that are changed as a result of what you enter in your application. For example, if you insert CSS in the CSS area in Page Definition, the CSS will be inserted in the #PAGE_CSS# area in the template when executed. You examine how this works later in this lesson.

Page Source: <head>

```

<!-- [if HTML5] ><! [endif] -->
<!doctype html>
<!-- [if lt IE 7 ]> <html class="ie6 no-css3 no-js" lang="en"> <! [endif] -->
<!-- [if IE 7 ]>      <html class="ie7 no-css3 no-js" lang="en"> <! [endif] -->
<!-- [if IE 8 ]>      <html class="ie8 no-css3 no-js" lang="en"> <! [endif] -->
<!-- [if IE 9 ]>      <html class="ie9" lang="en"> <!z [endif] -->
<!-- [if (gt IE 9) | !(IE)]><! --> <html class="no-js" lang="en"> <! --<! [endif] -->
</head>
<!-- [if !HTML5] >
    <meta http-equiv="X-UA-Compatible" content="#TITLE#" = "IE=edge,chrome=1">
<! [endif] -->
    <meta charset="UTF-8">
    <title>Sample Database Application</title>
    <link rel="icon" href="/i/favicon.ico" type="image/x-icon">
    <link rel="shortcut icon" href="/i/favicon.ico" type="image/x-icon">
<link rel="stylesheet" href="/i/css/apex.min.css?v=4.2.1.00.08" type="text/css" />
<!-- [if IE]><link rel="stylesheet" href="/i/css/apex_ie.min.css?v=4.2.1.00.08" type="text/css" /><! [endif] -->
<link rel="stylesheet" href="/i/libraries/jquery-ui/1.8.22/themes/base/jquery-ui.min.css?v=4.2.1.00.08" type="text/css" />
<link rel="stylesheet" href="/i/themes/theme_25/css/4_2.css?v=4.2.1.00.08" type="text/css" />
<link rel="stylesheet" href="/i/themes/theme_25/css/responsive_grid.css?v=4.2.1.00.08" type="text/css" />

<script type="text/javascript">
var apex_img_dir = "/i/", htmlDb_Img_Dir = apex_img_dir;
</script>
<script src="/i/libraries/apex/minified/desktop_all.min.js?v=4.2.1.00.08" type="text/javascript"></script>

<script src="/i/libraries/apex/minified/legacy.min.js?v=4.2.1.00.08" type="text/javascript"></script>

<script type="text/javascript" src="/i/libraries/modernizr/2.5.3/modernizr.min.js?v=4.2.1.00.08"></script>
<!-- [if lt IE 9]><script type="text/javascript" src="/i/libraries/respond.js/1.1.0/respond.min.js?v=4.2.1.00.08"></script><! [endif] -->

<script type="text/javascript" src="/i/themes/theme_25/js/4_2.min.js?v=4.2.1.00.08"></script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0" />
</head>

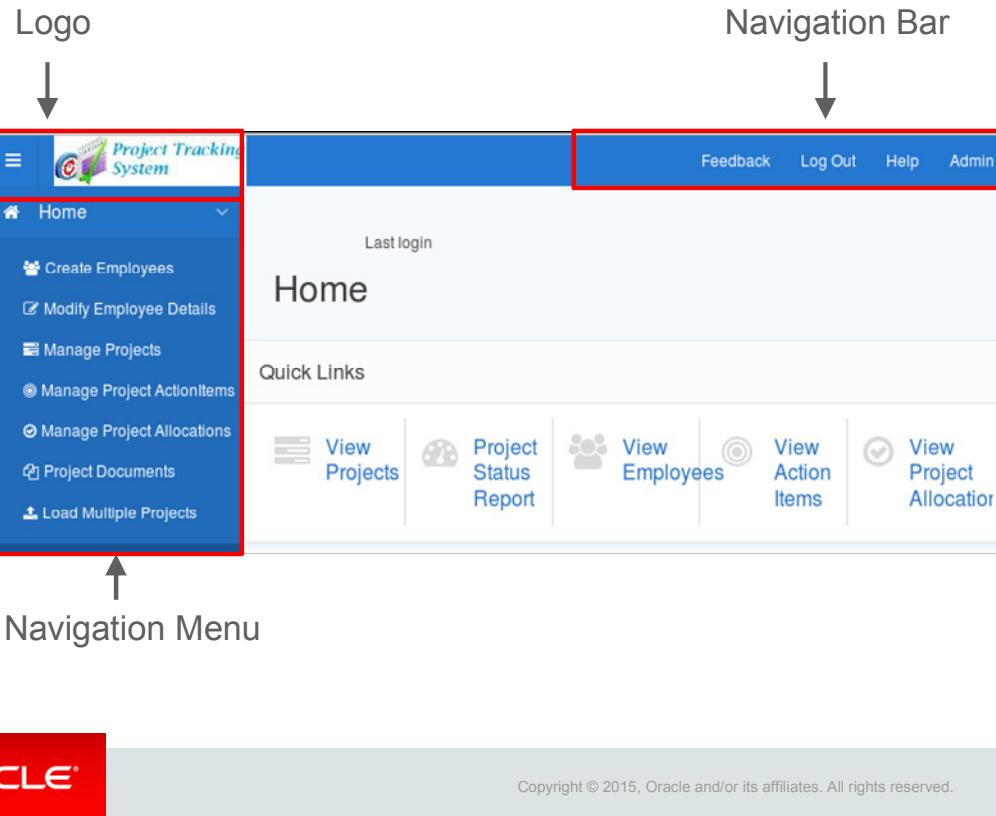
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If you review the Page Source, you will see how the substitution strings are resolved when the page is executed for the header.

Application Header



A screenshot of the Header area is in the slide. In the next set of slides, you see how the template inserts the correct HTML in the appropriate area in the header.

Page Template: <header>

```
<header class="t-Header" id="t_Header">
  #REGION_POSITION_07#
  <div class="t-Header-branding">
    <div class="t-Header-controls">
      <button class="t-Button t-Button--icon t-Button--header t-Button--headerTree"
id="t_Button_navControl" type="button"><span class="t-Icon fa-bars"></span></button>
    </div>
    <div class="t-Header-logo">
      <a href="#HOME_LINK#" class="t-Header-logo-link">#LOGO#</a>
    </div>
    <div class="t-Header-navBar">
      #NAVIGATION_BAR#
    </div>
  </div>
  <div class="t-Header-nav">
    #TOP_GLOBAL_NAVIGATION_LIST#
    #REGION_POSITION_06#
  </div>
</header>
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The code in the slide shows the HTML and the substitution strings in the Page Template for One Level Tabs–No Sidebar. Notice there are substitution strings for #LOGO#, #NAVIGATION_BAR#, and #TOP_GLOBAL_NAVIGATION_LIST#. In the next two slides, you see the page source with the substitution strings resolved.

Page Source: <header>

```
<div class="t-Header-logo">
    <a href="f?p=101:1:14633319515212" class="t-Header-logo-link"></a>
</div>
<div class="t-Header-navBar">
    <ul class="t-NavigationBar" id="2028263703523578821"><li class="t-NavigationBar-item">
        <a class="t-Button t-Button--icon t-Button--header t-Button--navBar"
href="javascript:apex.navigation.dialog('f?p=101:19:14633319515212::NO::19:P19_APPLICATION_ID,P19_PAGE_ID:101,1\0026p_dialog Cs=HHQXqmbvCnmjQGkkXCwlnF1b8i-HcpOUnldC9y-FhGq4DkkQB_Hk9hVvHsSUEb-CP4yfRERcx1o74zaof4BrXQ',{title:'Feedback',height:'500',width:'720',maxWidth:'960',modal:false,dialog: null},'t-Dialog--standard',apex.jQuery('#R'));" role="button">
            <span class="t-Icon "></span><span class="t-Button-label">Feedback</span><span class="t-Button-badge"></span>
        </a>
    </li><li class="t-NavigationBar-item">
        <a class="t-Button t-Button--icon t-Button--header t-Button--navBar"
href="apex_authentication.logout?p_app_id=101&p_session_id=14633319515212" role="button">
            <span class="t-Icon "></span><span class="t-Button-label">Log Out</span><span class="t-Button-badge"></span>
        </a>
    </li><li class="t-NavigationBar-item">
        <a class="t-Button t-Button--icon t-Button--header t-Button--navBar"
href="f?p=101:17:14633319515212:1:NO:::" role="button">
            <span class="t-Icon "></span><span class="t-Button-label">Help</span><span class="t-Button-badge"></span>
        </a>
    </li>
... </div>
</header>
...

```

#LOGO#

#NAVIGATION_BAR#

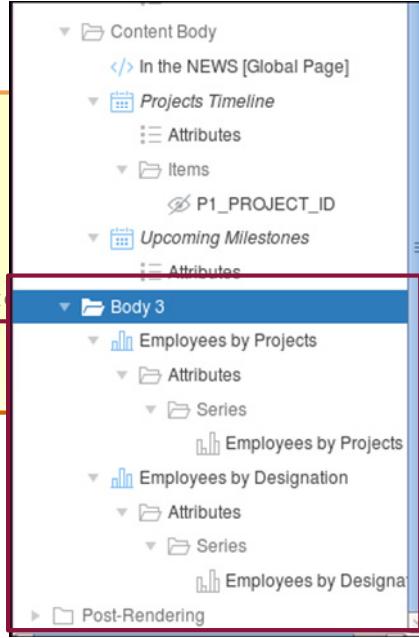


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Review the data that is inserted (per resolution of the substitution strings) when the page is executed for #LOGO# and #NAVIGATION_BAR#. This page source displays what is displayed in the slide.

Page Template: Body

```
<><div class="t-Body">
  #SIDE_GLOBAL_NAVIGATION_LIST#
<div class="t-Body-main">
  <div class="t-Body-title" id="t_Body_title">
    #REGION_POSITION_01#
  </div>
  <div class="t-Body-content" id="t_Body_content">
    #SUCCESS_MESSAGE##NOTIFICATION_MESSAGE##GLOBAL_NOTIFICATION#
    <div class="t-Body-contentInner">
      #BODY#
    </div>
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The regions located in the Body (3) area will be inserted in the #BODY# area in the page template.

Page Template: Position

The screenshot illustrates the Oracle Application Express interface for managing page templates. On the left, the 'Position' section of the 'Page Template: Position' configuration is shown, with a red arrow pointing to the 'Quick Pick: Position' button next to the 'Content Body' field. This button is used to quickly select a position for the region. On the right, the resulting page structure is displayed, featuring three main content blocks: 'Employees by Projects', 'Projects Timeline', and 'Upcoming Milestones', each with its own edit controls and sub-regions.

To see the position for the Page Template, you can click the Quick Pick position icon for the position in the Region Definition.

Page Template

The screenshot shows a web application interface for a 'Project Tracking System'. At the top, there's a blue header bar with the system name and navigation links for Feedback, Log Out, Help, and Admin. Below the header is a sidebar with a 'Home' link and a 'Last login' message. A 'Quick Links' section contains four items: 'View Projects', 'Project Status Report', 'View Page Template Body (3)' (which is highlighted with a yellow box), and 'View Project Allocations'. The main content area features two charts: 'Employees by Projects' (a bar chart) and 'Employees by Designation' (a pie chart). A red box highlights the entire content area. At the bottom, there's a toolbar with various icons and a footer with the Oracle logo and copyright information.

The slide shows an example of content displayed in the Page Template Body (3) location.

Using Button Templates

The screenshot illustrates the Oracle Application Express (APEX) interface for creating and configuring a button template. It consists of three main panels:

- Left Panel (Region Structure):** Shows the hierarchical structure of the page. A red box highlights the "CREATE" button under the "Region Buttons" section of the "Projects1" region.
- Middle Panel (Button Properties):** A modal window titled "Button" displays properties for the selected button:
 - Identification:** Button Name: CREATE, Label: Create.
 - Layout:** Sequence: 10, Region: Projects1, Button Position: Right of Interactive Report Search Bar.
- Right Panel (Appearance Properties):** A modal window titled "Appearance" displays properties for the button template:
 - Button Template:** Text.
 - Hot:** Yes (selected).
 - Template Options:** Use Template Defaults (selected).
 - CSS Classes:** (empty)

Below these panels is a preview of the "PROJECT TRACKING SYSTEM" application. It shows a table with project details and a "Create" button in the header. The "CREATE" button from the structure panel is highlighted with a red box. The footer of the application includes the "ORACLE" logo.

Each button has a button position and template. In the example in the slide, you see that because this is a button on an interactive report, you should specify the Button Position as Right of Interactive Report Search Bar. For the Button style, you want to use a template-based button. In this case, you want the button to be displayed with a right arrow so you need to specify the Button Template as Text with Icon. Lastly, you want to highlight the button so you want to select Yes for Hot. Here, for the Template Options, Use Template Defaults is selected.

Copying and Modifying a Template

The screenshot shows two side-by-side views of a web application interface for a "PROJECT TRACKING SYSTEM". Both views have a blue header bar with the title, a navigation menu on the left, and a search/filter bar at the top of the main content area.

Original - Hot View:

- Header:** PROJECT TRACKING SYSTEM, Feedback, Log Out, Help, Home.
- Left Menu:** Home, Create Employees, Modify Employee Details, Manage Projects, Manage Project Action Items, Project Documents, Project Charts, Projects_List.
- Table Headers:** Project Id, Project Name, Project Type, Project Description, Project Status, Project Planned Start Date, Project Start Date.
- Data Rows:**
 - Project Id: 612, Project Name: MFG Petrol Industry, Project Type: 304, Project Description: Engineering Design Capabilities in the Petrol Industry, Project Status: 101, Project Planned Start Date: 19-JUN-15, Project Start Date: 19-JUN-15, Priority: 0.
 - Project Id: 614, Project Name: NoSQL Course Testing, Project Type: 302, Project Description: Testing Course Lessons for NoSQL, Project Status: 101, Project Planned Start Date: 01-MAY-15, Project Start Date: 01-MAY-15, Priority: 1.

Custom - Hot View:

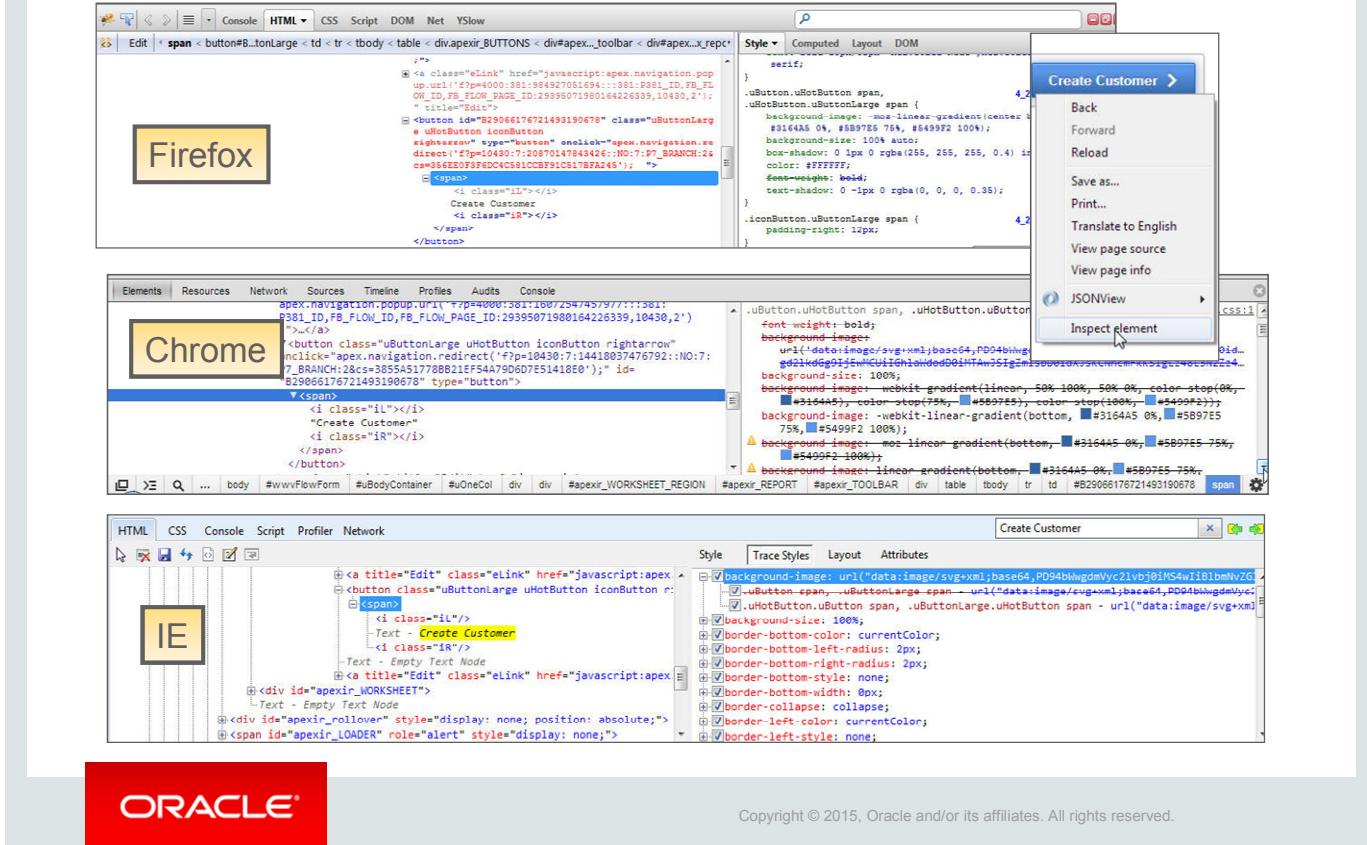
- Header:** PROJECT TRACKING SYSTEM, Feedback, Log Out, Help, Home.
- Left Menu:** Home, Create Employees, Modify Employee Details, Manage Projects, Manage Project Action Items, Project Documents, Project Charts.
- Table Headers:** Project Id, Project Name, Project Type, Project Description, Project Status, Project Planned Start Date, Project Start Date.
- Data Rows:**
 - Project Id: 612, Project Name: MFG Petrol Industry, Project Type: 304, Project Description: Engineering Design Capabilities in the Petrol Industry, Project Status: 101, Project Planned Start Date: 19-JUN-15, Project Start Date: 19-JUN-15, Priority: 0.

Bottom Elements:

- A red button labeled "ORACLE" on the left.
- A copyright notice: Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You may want to customize an existing template supplied by Oracle Application Express. For example, you want to change the font color of the Hot button. In the next set of slides, you create a custom Button Template and then change the template to include the changes you want. In this case, you will also create some styles to change the color in the page itself. You want to leave the color of gray when the button is set to Normal.

Inspecting Browser Elements



To understand which styles are used on your page, you can inspect the elements. To access the inspector, right-click the object you want to inspect and select Inspect element. Each browser has a different way of inspecting. With Firefox, you can install Firebug, which is a free add-in. The other browsers, such as Chrome and IE have their own software included with the browser. In the slide are screenshots of what each one looks like. To learn more about how each of these can be used, review the browser documentation.

Copying a Template

It is best practice to copy a template and change it rather than changing the Oracle APEX-supplied templates.

The screenshot shows a table of templates and a modal dialog for copying a template.

Table Headers: Name, Subscribed From, Subscribers, References, Updated, Default, Preview, Theme, Copy.

Table Data:

Name	Subscribed From	Subscribers	References	Updated	Default	Preview	Theme	Copy
Breadcrumb	Theme	1	5		✓		42	
Breadcrumb	Theme		1	12 days ago	✓		125	
Standard Button	Theme		4		✓		51	
Text	Theme	1	20		✓		42	
Text	Theme		1	12 days ago	✓		125	
Text with Icon	Theme	1	0				42	

Copy Template Dialog:

- Template: Text
- New Template Name: Large Text
- Copy button (highlighted with a red box)

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If you need to change one or a few existing templates (from the templates supplied by Oracle APEX), it is best to make a copy of the template, by assigning it a different name, and then modify the copied template. Then, associate the copied template to the desired page. The reason you copy a template is because you always have the original template to go back to or use in a different application.

To copy a template:

1. On the page, go to **Shared Components > Templates**.
2. For the template you want to copy, click the **Copy** icon.
3. Enter a new template name and click **Copy**.

Modifying a Template

The screenshot shows the Oracle Application Express interface for modifying a template. At the top, there is a 'Normal Template' editor window containing the original template code for a large red button with icons. Below it is a 'Hot Template' editor window where the 'redButton' class has been added to the button's definition. To the right, a 'Page Definition - CSS' panel displays the inline CSS styles for the 'redButton' class, setting the background color to red and the border to black. The Oracle logo is visible at the bottom left, and a copyright notice is at the bottom right.

[Custom] Large Red Button - Icon

```
<button class="t-Button t-Button--icon #BUTTON_CSS_CLASSES# #BUTTON_ATTRIBUTES# onclick="#JAVASCRIPT#" type="button" id="#BUTTON_ID#"><span class="t-Icon t-Icon--left #ICON_CSS_CLASSES#" aria-hidden="true"></span><span class="t-Button-label">#LABEL#</span><span class="t-Icon t-Icon--right #ICON_CSS_CLASSES#" aria-hidden="true"></span></button>
```

```
<button value="#LABEL#" onclick="#JAVASCRIPT#" class="redButton" type="button" #BUTTON_ATTRIBUTES# id="#BUTTON_ID#"><span width="100%">#LABEL#</span></button>
```

Page Definition - CSS

```
redButton {  
background-color: #FF0000;  
border: 1px solid black;  
}
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

After the template is copied, you can modify the template. In this example, you add a class to the Hot Template called redButton. Then, you add two CSS styles to the Page Definition that will set the color to red when the class is redButton. The second CSS style is specified to make sure the border is black. You can also add these CSS styles to the page template which will then make the CSS available to all pages that use that template.

Associating Template with Button

The screenshot illustrates the process of associating a template with a button in Oracle Application Express. On the left, the 'Button' definition page is shown, where the 'Button Name' is set to 'CREATE' and the 'Label' is 'Create'. In the 'Appearance' section, the 'Button Template' dropdown is set to 'Custom - Red' and is highlighted with a red box. On the right, the 'PROJECT TRACKING SYSTEM' application is displayed, showing a table of projects. A red box highlights the 'Create' button in the 'Actions' column of the first row. The Oracle logo is visible at the bottom left of the page.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The last step needed to use the modified template is to change the Button Template in the Button Definition.

Quiz



If you must create a new custom template, it is generally easier to start from the beginning.

- a. True
- b. False

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: b

Creating a Custom Theme

To create a custom theme:

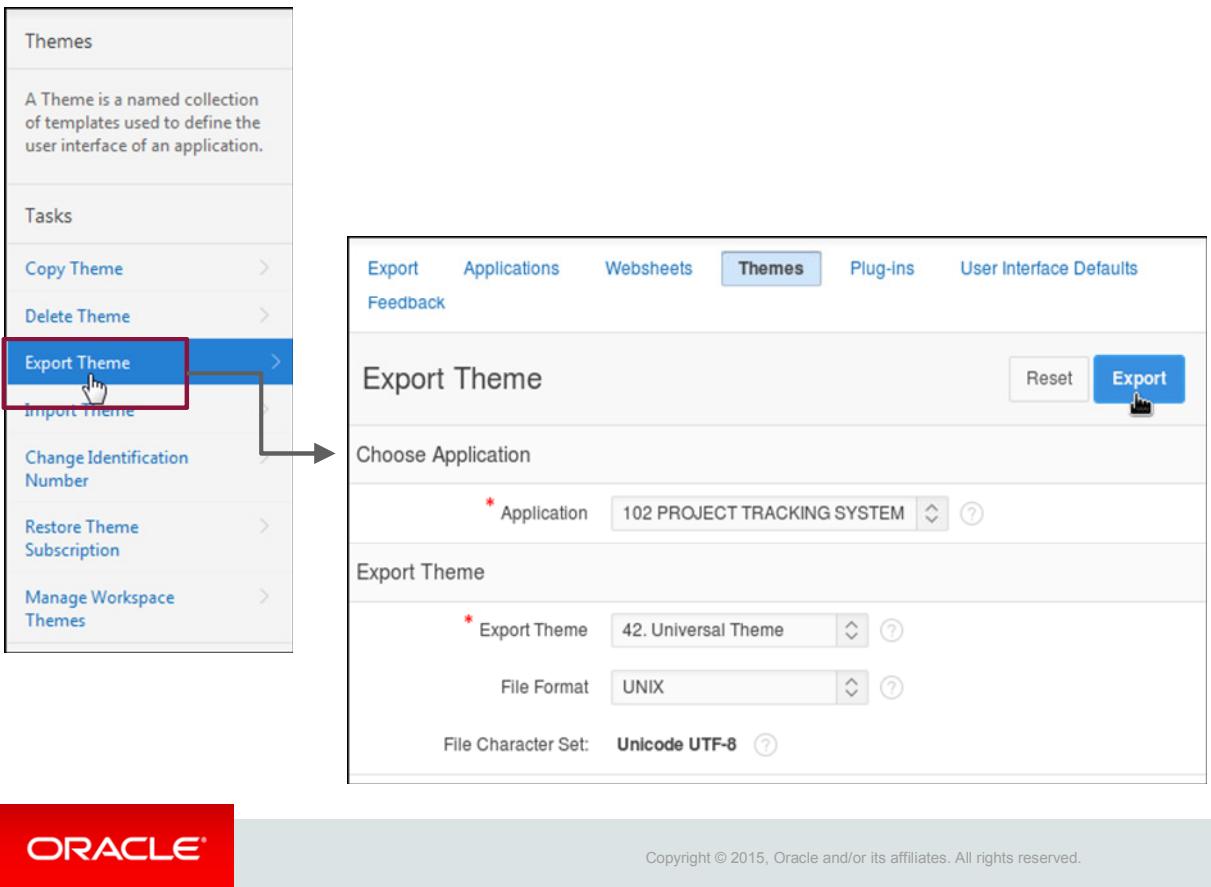
1. Export your theme
2. Copy your theme
3. Edit your theme
4. Manage your workspace themes



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

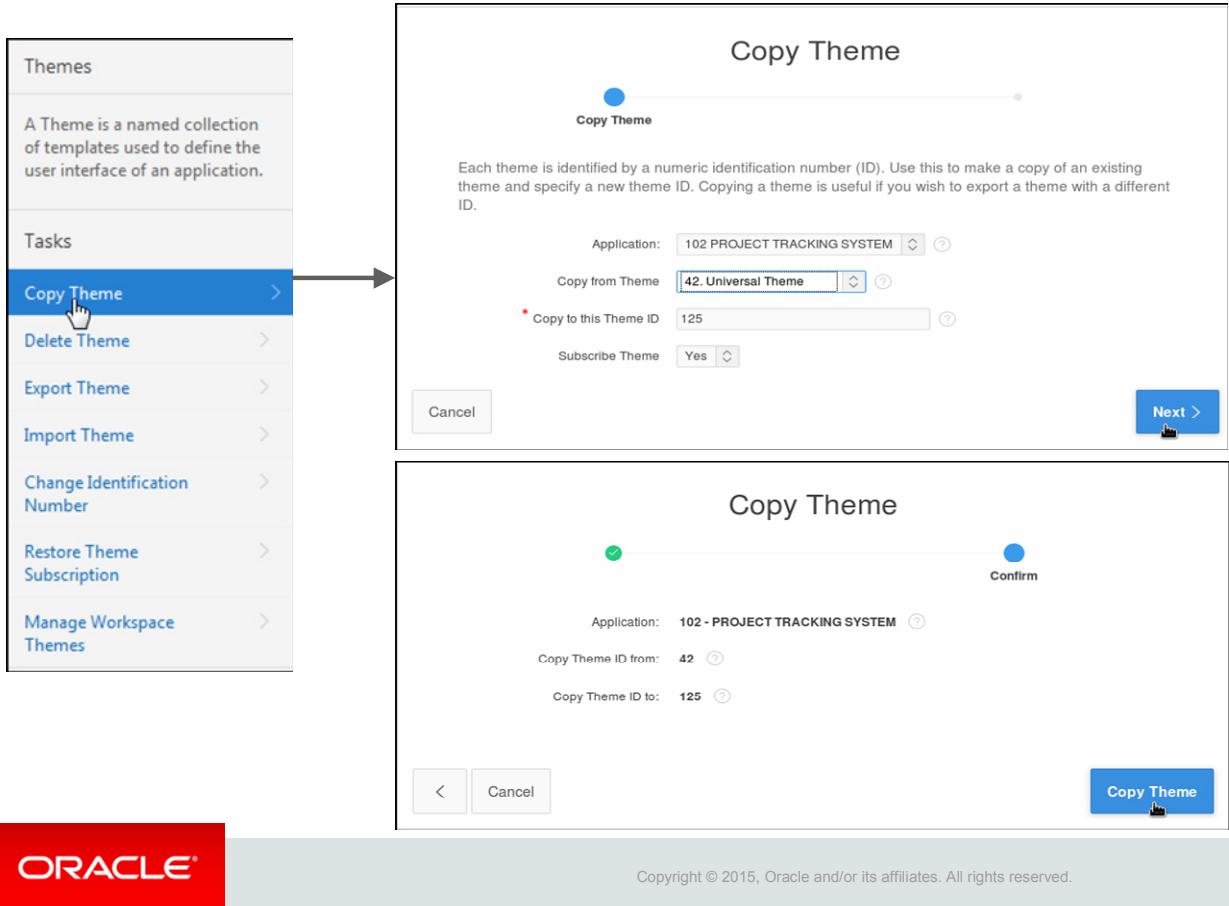
You may want to make the changes you made to the templates available to other applications. To create a custom theme, follow the steps in the slide.

Exporting Your Theme



To export your theme, navigate to **Shared Components > Themes**, and select **Export Theme** from the Tasks list on the right side of the page. Make sure the application is selected as well as the theme you want to export. There are two ways to add a custom theme to your new application. One way is to import the theme. You first need to export the theme before importing it.

Copying Your Theme



To copy a theme, from the Tasks list, select Copy Theme. Enter a number in the 100 series, in this case 125, and click Next. Then click Copy Theme.

Editing Your Theme

The screenshot shows the Oracle Application Express interface. On the left, there's a sidebar titled "User Interface" with a "Themes" section highlighted. The main area is titled "Theme" and shows a table of themes. One theme, "Universal Theme" (Number 42), is selected and shown in detail on the right. The "Create >" button is visible at the top of the theme list.

Number	Name	User Interface	Is Current	Subscribed From	Subscribers	Templates	Page Templates	Region Templates	Button Templates	List Templates
42	Universal Theme	Desktop	✓	Nonexistent Master		52	9	13	3	11
51	Mobile + 51 *	Mobile	✓	Nonexistent Master		30	3	14	2	5

On the right, the "Create / Edit Theme" dialog is open. It shows the following details:

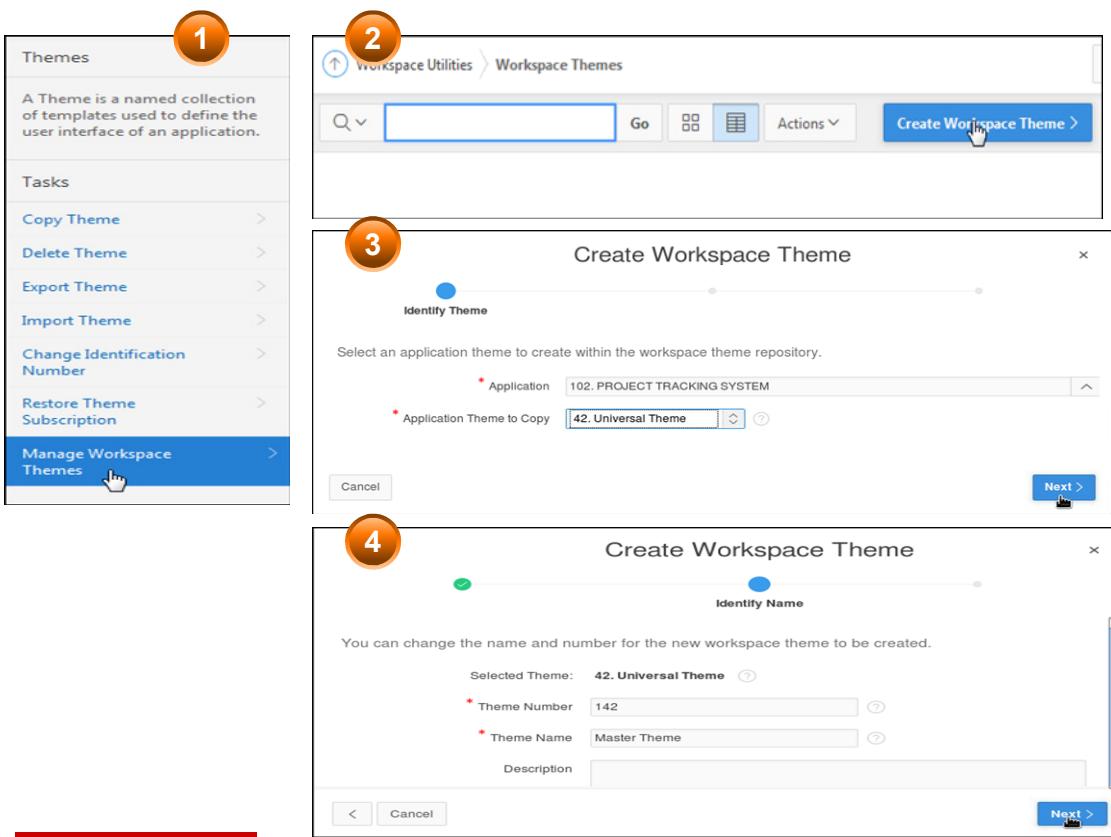
- Application: 102
- Theme Number: 42
- Name: Universal Theme
- User Interface: Desktop
- Navigation Type: List
- Description: (empty)

At the bottom of the dialog are buttons for "Verify", "Unsubscribe", "Refresh Theme", and "Publish Theme".

You need to edit your theme to make sure it has a distinguishable name and ensure that all the defaults have templates.

1. Navigate to the **Themes** page.
2. Click the **theme name**.
3. The **Create/Edit Theme** page appears.
4. Select the Theme name link for the theme you just copied.
5. Change the name to something different to distinguish it from another theme.
6. Make sure that there is a default theme for all the components and click **Apply Changes**.

Managing Workspace Themes

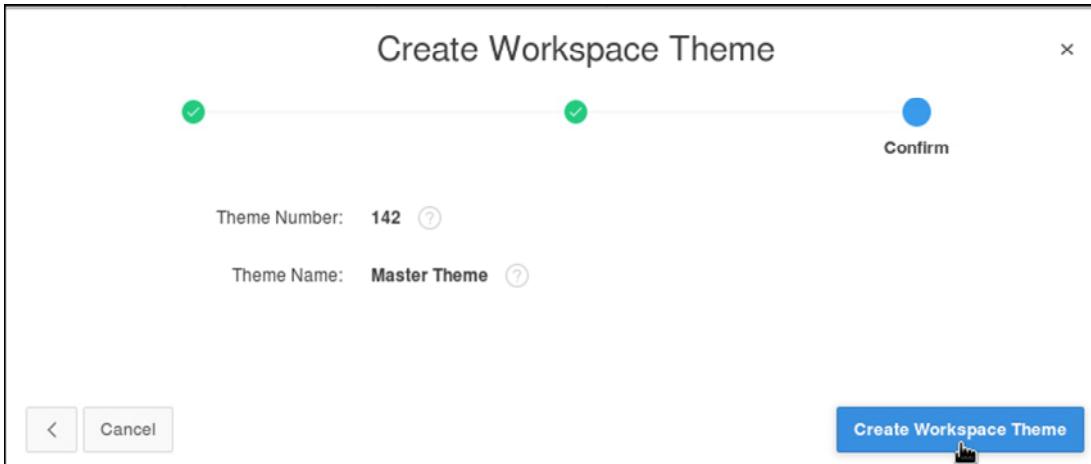


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You may want to make this custom theme available to any application you create. For this, you will need to add it as a Workspace Theme. To do so, perform the following steps:

1. From the Tasks List, select **Manage Workspace Themes**.
2. Select Create and click **Next**.
3. Select your custom theme and click **Next**.
4. Enter a Theme Name and click **Next**.

Managing Workspace Themes



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

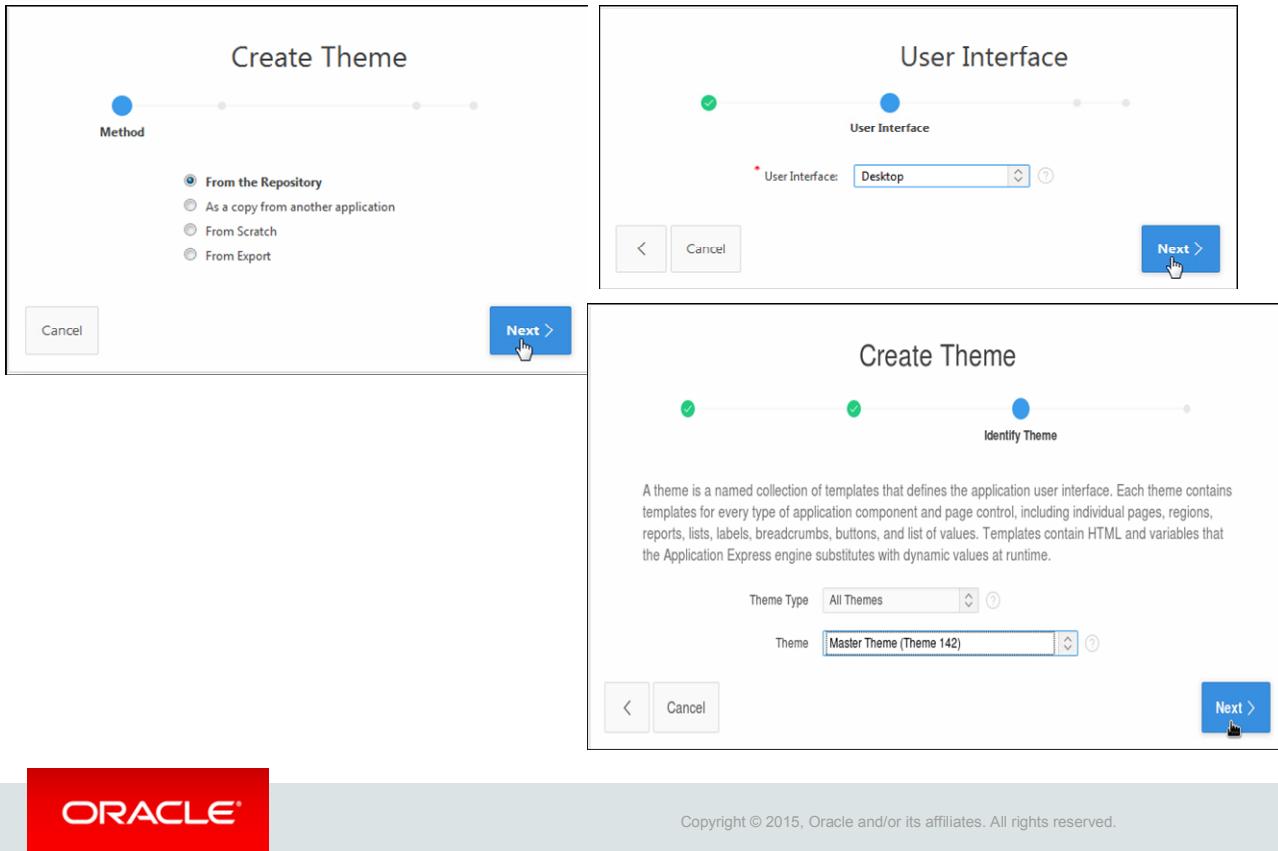
5. Click **Create Workspace Theme**.

Creating a Master Application

The screenshot shows the Oracle Application Express interface for creating a new application. At the top, it says "Application 102" and "Edit Application Properties". Below that, there are five main categories: "Run Application", "Supporting Objects", "Shared Components", "Utilities", and "Export / Import". A search bar and a toolbar with "Create Page >" are at the top of the page list. The page list shows four pages: "0 - Global Page - Mobile", "1 - Home", "2 - Home", and "3 - Project Status Report". The Oracle logo is at the bottom left, and a copyright notice is at the bottom right.

You can create an application with the custom theme that all groups will subscribe to. By doing this, you can change any template in the Master Application and all applications that are subscribed to the Master will automatically have the changes. Create an application with just Login and Home pages.

Adding a Custom Theme to the Master Application



Workspace themes are not available in the Create Application wizard, but they are contained in the repository. To add the custom theme to the master application:

1. From Shared Components, select **Themes**.
2. Select From the Repository and click **Next**.
3. Make sure Desktop is selected and click **Next**.
4. Select **Custom Themes** for Theme Type, select your custom theme, and click **Next**.
5. Click **Create**.

Switching the Current Theme

The screenshot shows the Oracle Application Builder interface. At the top, there's a navigation bar with 'Go', 'Actions', 'Reset', 'Switch Theme', and 'Create >'. Below it is a grid table with columns: Number, Name, User Interface, Is Current, Subscribed From, Subscribers, Templates, Page Templates, Region Templates, Button Templates, and List Templates. Three rows are listed:

Number	Name	User Interface	Is Current	Subscribed From	Subscribers	Templates	Page Templates	Region Templates	Button Templates	List Templates
42	Universal Theme - 42 "	Desktop	✓	Nonexistent Master		52	9	13	3	11
51	Mobile - 51 "	Mobile	✓	Nonexistent Master		30	3	14	2	5
142	Master Theme - 142	Desktop		Theme Repository		52	9	13	3	11

Below the grid is a 'Switch Theme' dialog box. It has two tabs: 'Switch Theme' (selected) and 'Verify Compatibility'. The 'Switch Theme' tab shows the application is '102 - PROJECT TRACKING SYSTEM' and the 'Currently Active Theme' is '42. Universal Theme'. The 'Switch to Theme' dropdown is set to '142. Master Theme'. A table below lists template mappings:

Template Type	From Template	To Template	Status
Breadcrumb	Breadcrumb	Breadcrumb	✓
Button	Text with Icon	Text with Icon	✓
	HTML button (legacy - APEX 5 migration)	HTML button (legacy - APEX 5 migration)	✓
	Icon	Icon	✓
	Text	Text	✓

At the bottom of the dialog is a note: 'When you switch to a new theme, Application Builder maps all currently used templates to a template in the new theme using the template class. This report displays these template mappings and allow for selecting alternate templates when no templates with a matching template class exist.' The Oracle logo is at the bottom left, and copyright information is at the bottom right.

After your custom theme is added to your application, you may want to make it the current theme. To do so, perform the following steps:

1. Click **Switch Theme**.
2. Select the Active Theme and the Switch to Theme values, and click **Next**.
3. A Theme Compatibility window appears. Make sure that there is an equivalent to Template Class specified and click **Next**.
4. Click **Switch Theme**.

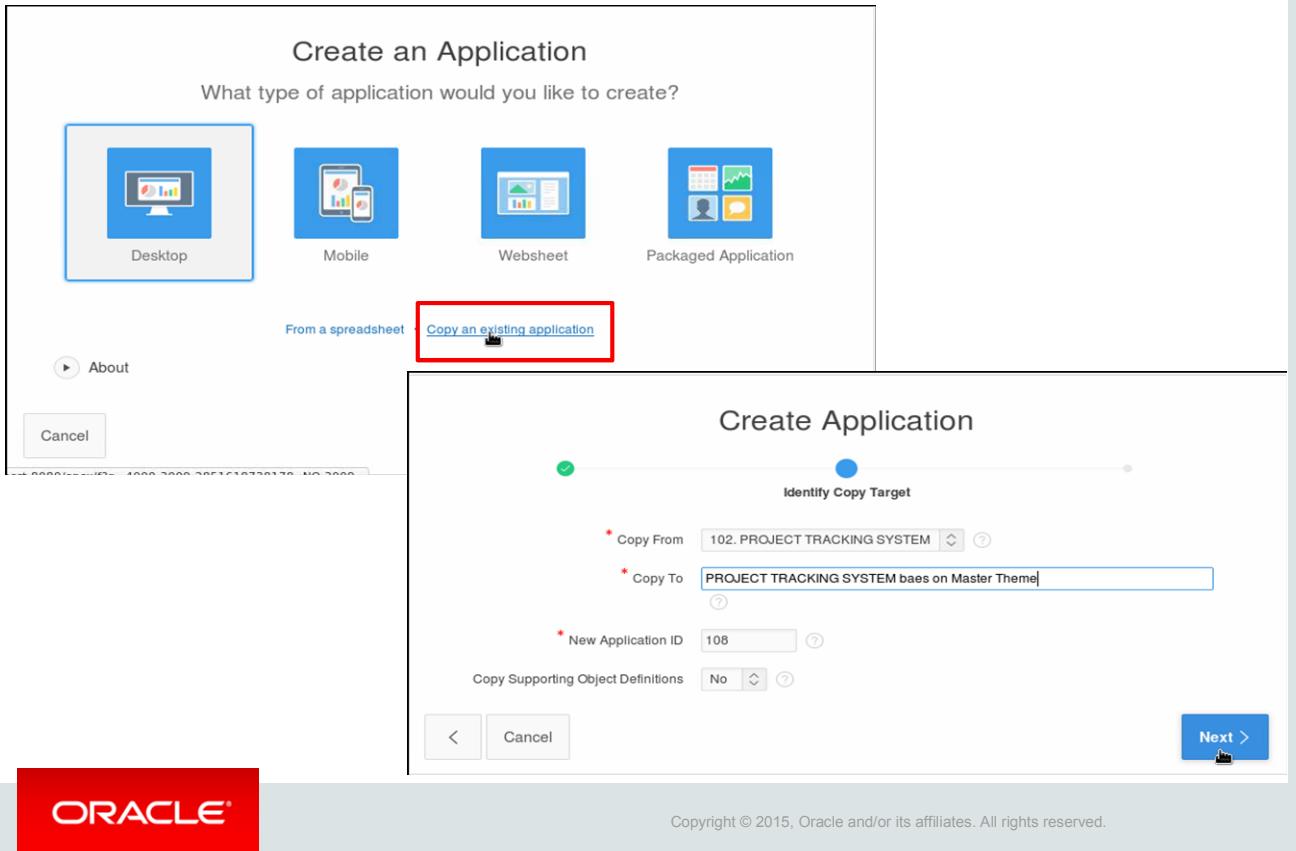
Deleting the Default Theme

The screenshot shows two panels. On the left is a sidebar titled 'Themes' with a descriptive text about themes. Below it is a 'Tasks' section with several options: 'Copy Theme', 'Delete Theme' (which is highlighted with a blue background), 'Export Theme', 'Import Theme', 'Change Identification Number', 'Restore Theme Subscription', and 'Manage Workspace Themes'. On the right is a 'Delete Theme' dialog box. It has a title 'Delete Theme' with a circular progress bar. Below it is a section for 'Application' set to '102 - PROJECT TRACKING SYSTEM'. A field labeled 'Delete Theme:' contains '101. My New Theme'. At the bottom are 'Cancel' and 'Next >' buttons.

When you created an application, you used one of the APEX-supplied themes. To delete the default theme, perform the following steps:

1. From the Tasks List, select **Delete Theme**.
2. Select the Theme that was created when the application was created and click **Next**.
3. Click **Delete Theme**.

Copying an Existing Application From the Master Application



When you have the Master Application, you can create all your other applications by copying from an existing application. To do so, perform the following steps:

1. From the Application Builder, click **Create**.
2. Select **Copy an existing application**.
3. Select your Master Application for the Copy From field, enter a new application name for Copy To, and click **Next**.
4. Click **Create Application**.

Quiz



A standard theme contains templates for every type of application component and region type.

- a. True
- b. False

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a

Practice 11 Overview: Using Templates and Themes

This practice covers the following topics:

- Copying and Customizing a Button Template
- Creating a Custom Theme
- Creating a Master Application



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Differentiate between the various types of applications (Desktop, Mobile, and Responsive Design)
- Use substitution strings within a template
- Create a new theme and use it in an application



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you should have learned the differences between the various types of applications, such as desktop, mobile, and responsive designs. You should have learned how substitution strings work in a template and you should also be able to create a new theme and use it in an application.

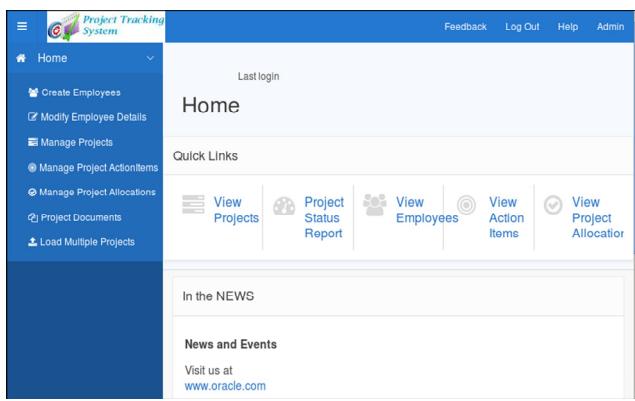
10

Developing a New Theme for Your Application Using Theme Roller

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Theme Roller in PTS



How do I change
the look and feel
of the
application?



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack wants to apply a new theme to the PTS application with different colors, fonts, and so on.

Objectives

After completing this lesson, you should be able to develop a new theme for your application using Theme Roller.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learn how to build a tab structure using a static list.

What is Theme Roller?

Theme Roller:

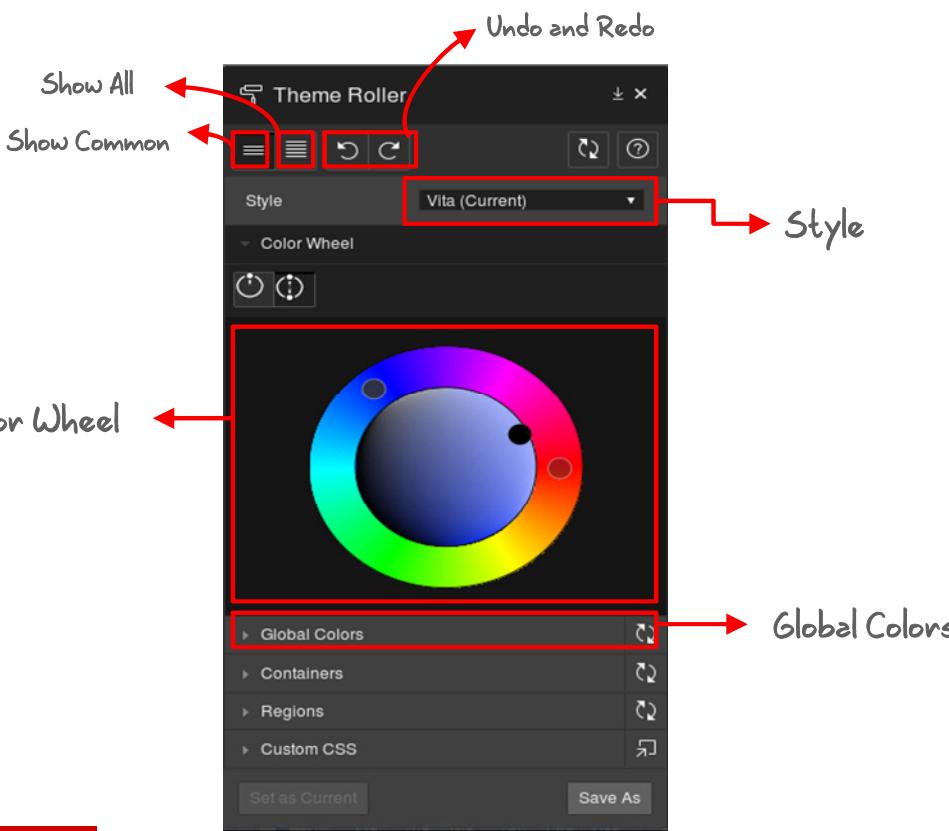
- Is a live CSS editor
- Allows developers to play around with theme colors, fonts, and theme layouts
- Allows easy customization of UI without getting into CSS, HTML, or JavaScript
- Can completely change the look and feel of UI just by playing around with Theme Roller
- Can save private themes using Theme Roller



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Theme Roller is a live CSS editor which allows developers to play around with theme colors, fonts, and theme layouts without getting into CSS, HTML, or JavaScript.

Components of Theme Roller



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can expand or collapse sections of the Theme Roller by clicking on the section names.
Primary control includes:

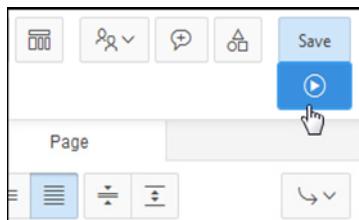
- **Show Common:** Displays all attributes.
- **Show All:** Displays all attributes.
- **Undo:** Reverts the currently edited theme to the previous action. This feature will not work if you have switched themes.
- **Redo:** Dismisses the current Undo revision and goes to the next one in the history. This feature will not work if you have switched themes.
- **Search:** Searches for properties, groups, and colors. As you search, Theme Roller automatically displays only properties that match your search string.
- **Reset:** Reverts the selected theme to the last version saved on the server. After resetting your theme and reloading your page, you can use Undo and Redo to restore your changes.
- **Help:** Displays a Help window.
- **Style:** There are two styles that come with APEX 5.0: Blue and Gray. To select a new style, choose an existing style from the Style list. It will set predefined colors for the different parts of the templates.

Color Wheel: Changes colors quickly using the Color Wheel. To change all the colors of the theme style at once, drag the circles in the color wheel. You can choose between making your theme style dependent on one color (monochromatic) or two (dual).

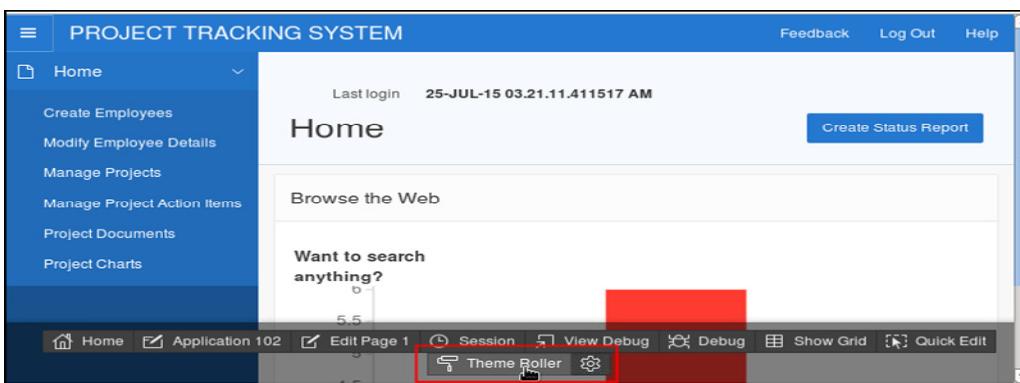
Global Colors: If the Color Wheel is not specific enough for what you need, you can start by customizing the Global Colors. Those are the main colors of the Universal Theme and are used to drive the specific components. You can still customize the different components, such as the header, by clicking further down the list.

Using Theme Roller in PTS

- 1 Run the application.



- 2 Click Theme Roller on the APEX Developer toolbar.



Using Theme Roller in PTS

- 3 To select a new style, choose an existing style from the Style list.

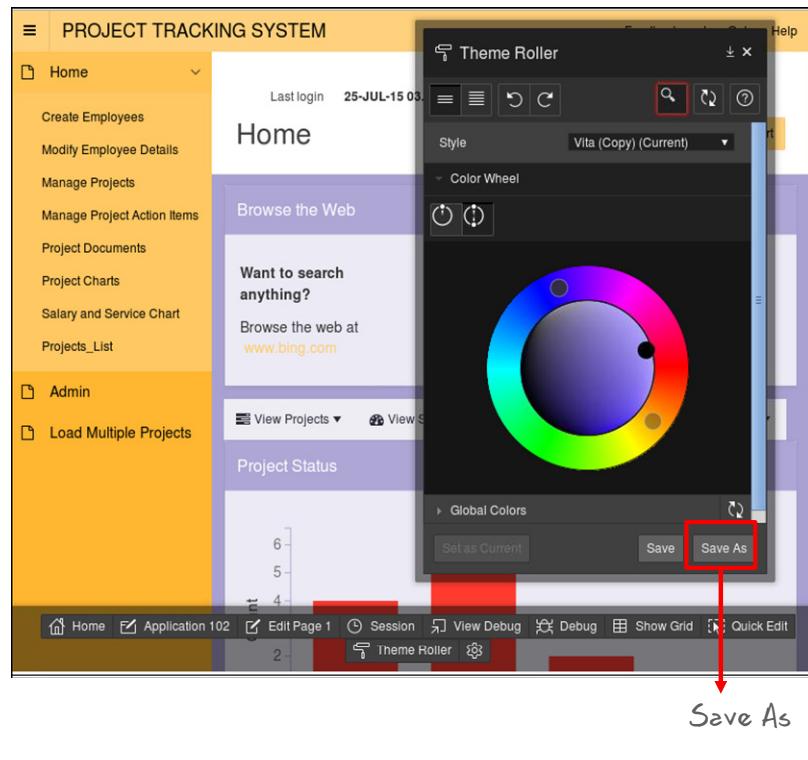


ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Using Theme Roller in PTS

- 4 To change all the colors of the theme style at once, drag the circles in the Color Wheel. You can choose between making your theme style dependent on one color (monochromatic) or two (dual).



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

1. To edit a specific component, expand a group and click on the color or styling of the component you wish to edit.
2. Click **Save** to commit your changes to the server. If the selected theme is read-only, then click **Save As** to save your changes as a new theme.

Practice 10 Overview: Developing a New Theme for Your Application Using Theme Roller

In this practice, use Theme Roller to change the look and feel of the GMT application.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to develop a new theme for your application using Theme Roller.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only



Making an Application Production-Ready

Unit III

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack Wants to Deploy the PTS Application



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack is almost done with upgrading the Project Tracking System with some great new features. As a final task, he is planning to add some additional security, optimization, and globalization features. He then needs to package the application and deploy it on the production server for the team to start using the upgraded application.

Course Road Map

Unit 1: Enhancing Application Pages

**Unit 2: Adding Advanced APEX
Functionality to an Application**

**Unit 3: Making an Application
Production-ready**

Lesson 11: Securing an Application

**Lesson 12: Deploying and Maintaining
an Application**

Lesson 13: Optimizing an Application

Lesson 14: Globalization and Translation



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this unit, you learn APEX features and functionalities that will help you make an application production-ready.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

11

Securing an Application

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Securing PTS



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The project tracking application already takes care of authentication and authorization for the various users. Currently, Jack is using APEX default authentication in the application. He has been requested by Jill to try and incorporate Lightweight Directory Access Protocol (LDAP) authentication.

In addition, Jack has come to know that he can secure his application against SQL Injection and other such security threats. He decides to implement these in the project tracking application.

Objectives

After completing this lesson, you should be able to:

- Prevent SQL Injection Attacks
- Prevent cross-site scripting
- Apply application-level security
- Authenticate users by using LDAP



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learn how applications can be secured by preventing SQL Injection Attacks and cross-site scripting. You also learn how to authenticate users by using LDAP.

Common Hacking Mechanisms

In this lesson, you learn to secure an APEX application against the following two common hacking mechanisms:

- SQL Injection
- Cross-Site Scripting



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Web applications connect to the Internet and provide opportunities to share information with a global audience. But it also provides opportunities for Internet users who are involved in malicious activities. Two of the most common hacking mechanisms used by these users are listed on the slide.

Most of the time, an APEX application is considered safe from such attacks. However, with more and more applications delivering personalized web experiences to users, there is an increase in the amount of dynamic content generated by web applications. Dynamic content is most vulnerable to security threats. Data security should not be an afterthought or practice done at the end of application development. Data security should be looked into at the time of development itself. Each time you write code, especially dynamic code, you should ensure the code is written to prevent any security threats.

In this lesson, you learn about these two security threats and what precautions you can take to prevent these attacks in an APEX application.

SQL Injection

SQL Injection is a technique for maliciously exploiting applications that use client-supplied data in SQL statements.

- Attackers trick the SQL engine into executing unintended commands.
- SQL Injection techniques may differ, but they all exploit a single vulnerability in the application.
- To immunize your code against SQL Injection Attacks, use bind arguments, or validate and sanitize all input concatenated to dynamic SQL.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

SQL Injection is a technique for maliciously exploiting applications that use client-supplied data in SQL statements. In other words, SQL Injection means adding additional SQL code to a SQL statement to change the way it runs from the way it was intended to run.

Attackers trick the SQL engine into executing unintended commands by supplying specially crafted string input, thereby gaining unauthorized access to a database to view or manipulate restricted data.

SQL Injection techniques may differ, but they all exploit a single vulnerability in the application. String literals that are incorrectly validated or not validated are concatenated into a dynamic SQL statement and interpreted as code by the SQL engine.

To immunize your code against SQL Injection Attacks, you must use bind arguments (either automatically with static SQL or explicitly with dynamic SQL), or validate and sanitize all input concatenated to dynamic SQL.

Although any program or application may be vulnerable to SQL injection, web applications are at a higher risk because an attacker can perpetrate SQL Injection Attacks without any database or application authentication.

SQL Injection: Example

This page contains a search item and a PL/SQL process region with the following code:

```
declare
  lname varchar2(1000);
begin
  lname:= 'select last_name from oechr_employees where employee_id=' || 
  :P<n>_SEARCH;
  return lname;
end;
```

Execute the example and enter 103. Click Go.

The screenshot shows a web application interface. At the top, a header reads "SQL Injection Example". Below it is a search form with a "Search" input field containing "103" and a "Go" button. Underneath is a table titled "Employee List" with two columns: "COL01" and "Last Name". The first row shows "103" in the "COL01" column and "Hunold" in the "Last Name" column. A yellow callout box with the text "So far so good!" has an arrow pointing to the "Last Name" cell. The Oracle logo is at the bottom left, and a copyright notice "Copyright © 2015, Oracle and/or its affiliates. All rights reserved." is at the bottom right.

The example shown in the slide demonstrates a substitution string in a `SELECT` statement. On the page, there is also a search item where an employee ID is entered and the employee's last name is shown in the resulting list. In this case, the employee ID 103 was entered into the Search field and the employee's last name Hunold was shown in the result.

SQL Injection: Example

Now try entering 103 union all select username from all_users.

All the names from the all_users table are shown.
SQL Injection vulnerability!

COL01
Hunold
ORA22
ORA21
ORA20
ORA19
ORA18
ORA17
ORA16
ORA15

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

If you enter 103 union all select username from all_users in the Search field and click Go, you can see that the list includes all the schema users and the employee's last name. This is a SQL Injection vulnerability.

Correcting SQL Injection: Using Bind Variables

Change the substitution string to use the bind variable:

```
declare
  lname varchar2(1000);
begin
  lname:= 'select last_name from oehr_employees where
employee_id=:P<n>_SEARCH';
  return lname;
end;
```

Execute the example and enter 103. Click Go.

The screenshot shows a user interface titled "SQL Injection Example". A search input field contains the value "103". Below it, a table titled "Employee List" displays one row with columns labeled "COL01" and "Hunold". A callout box with an arrow points to the "Hunold" cell, containing the text "Still receive correct value". The Oracle logo is at the bottom left, and a copyright notice is at the bottom right.

The standard approach to remedy this type of SQL Injection is to use bind variables. So instead of concatenating the substitution variable with another string in a step before the PL/SQL block is parsed, thus allowing the query to be arbitrarily extended beyond its original syntactic form, use :P<n>_SEARCH, which will use only the session state value during the parsing and execution of the block.

Correcting SQL Injection: Using Bind Variables

However, entering 103 union all select username from all_users causes an error.

SQL Injection Example

Search Go

Employee List

report error:
ORA-01722: invalid number

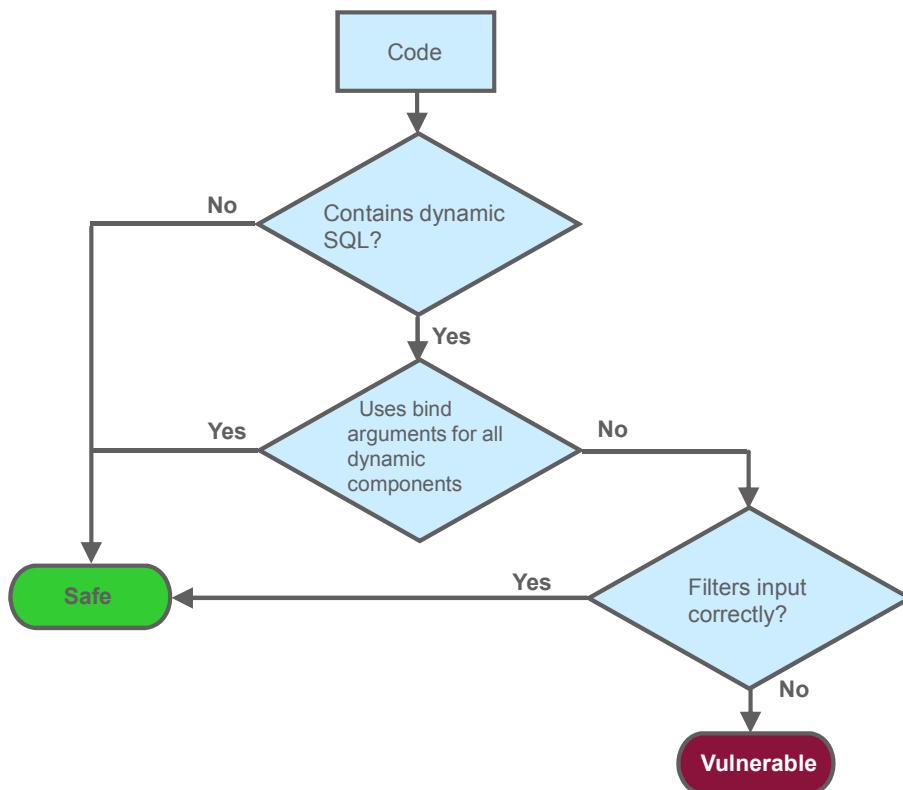
Bind variable can be used only to compare one value against another and cannot extend the query.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

With a bind variable, no matter what value is in session state for the page item P<n>_SEARCH, it can serve only as a value to compare against in the predicate (where employee_id = <something>). It can never be used to extend the query as in the original case.

Assessing Vulnerability



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can analyze your code to determine SQL Injection vulnerability. The flowchart shows you how to start assessing for vulnerability.

Best Practices Against SQL Injection

Strategy	Description
Reduce the attack surface.	Ensure that all excess database privileges are revoked and only those routines that are intended for end-user access are exposed.
Avoid dynamic SQL with concatenated input.	Dynamic SQL built with concatenated input values presents the easiest entry point for SQL injections. Avoid constructing dynamic SQL this way.
Use bind arguments.	Parameterize queries by using bind arguments. Not only do bind arguments eliminate the possibility of SQL injections, they also enhance performance.
Filter and sanitize input.	The Oracle-supplied DBMS_ASSERT package contains a number of functions that can be used to sanitize user input and to guard against SQL Injection in applications that use dynamic SQL built with concatenated input values. In case your filtering requirements cannot be satisfied by the DBMS_ASSERT package, you may need to create your own filter.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use several avoidance strategies to safeguard against, or mitigate the impact of, SQL Injection Attacks. Listed in the slide are high-level descriptions of each of the strategies that are examined in more detail on subsequent pages.

The available and best methods for eliminating SQL Injection vulnerability may depend on the vulnerability itself. Not all methods are available for addressing every vulnerability.

Methods:

- **Reduce the amount of likely attack points:** Make sure that all the excess database privileges are revoked and only those routines that are intended for end-user access are exposed. Though this does not entirely eliminate SQL Injection vulnerabilities, it does mitigate the impact of the attacks.
- **Use static SQL:** Use static SQL if all Oracle identifiers (for example, column, table, view, trigger, program unit, or schema names) are known at code-compilation time.
Note: Static SQL automatically binds arguments. Data Definition Language (DDL) statements cannot be executed with static SQL.
- **Use dynamic SQL with bind arguments:** Use this option if any WHERE clause values, VALUES clause values, or SET clause values are unknown, and any Oracle identifiers are unknown at code-compilation time.

Note: Use bind arguments for values (literals). Use string concatenation for validated and sanitized Oracle identifiers.

- **Validate and sanitize input:** Use this option if concatenating any strings and if bind arguments require additional filtering.

Quiz



Code that is most vulnerable to SQL Injection Attacks:

- a. Input parameters
- b. Dynamic SQL with bind arguments
- c. Dynamic SQL with concatenated input values
- d. Call to external functions

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: c

Cross-Site Scripting

Cross-site scripting (also referred to as XSS) is a security breach that takes advantage of dynamically generated web pages.

- XSS happens when a web application is sent a script that activates when it is read by a user's browser.
- When activated, these scripts can steal data, even session credentials, and return the information to the attacker.
- These scripts could be rendered into HTML regions and other places within the application during normal page rendering.
- To prevent XSS, the Application Express engine escapes characters in certain cases.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Cross-site scripting is a security breach that takes advantage of dynamically generated web pages. In an XSS attack, a web application is sent a script that activates when it is read by a user's browser. When activated, these scripts can steal data, even session credentials, and return the information to the attacker.

If malicious code were introduced into an Oracle Application Express application, it could be rendered into HTML regions and other places within the application during normal page rendering. To prevent the introduction of malicious code into session state, the Application Express engine escapes characters in certain cases.

Cross-Site Scripting: Example

1. Create a hidden item with default value “Welcome New User.”
2. Create a dynamic PL/SQL region with the `htp.p(:P<n>_H); source.`

The screenshot shows a web page titled "Cross-Site Scripting Demo". It has a button labeled "Print Value" and a dynamic region containing the text "Welcome New User". A yellow callout box on the right says "So far so good!". The Oracle logo is at the bottom left, and a copyright notice is at the bottom right.

When components in an Application Express page use `htp.p` to emit the values of page items or application items to the browser, special precautions are necessary to protect against cross-site scripting attacks.

In the example in the slide, you create a hidden item type. You create a computation that sets the item's value to Welcome New User and execute the computation only when the value for the item is null. Finally, create a dynamic PL/SQL region with the `htp.p(:P<n>_H); source.` When you run the page, you notice that the value is shown.

Cross-Site Scripting: Example

Replace the last “.” in the URL with the following script:

```
P<n>_H:<script>alert (document.cookie) ;</script>
```



Potentially harmful information can be displayed. XSS vulnerability!

ORACLE®

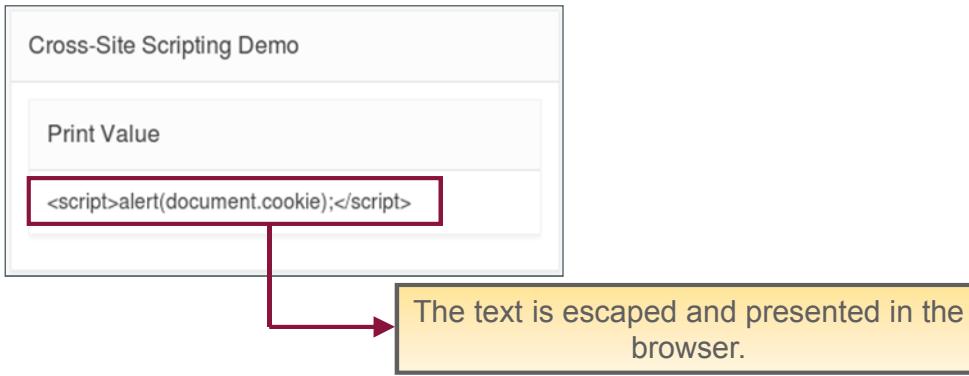
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This time, an alert box is displayed, listing all the cookies sent by the browser to the application in the current request. Although displaying cookie values to the current user may not be a security problem, the technique illustrates that a malicious user can get the application to send information to the browser, contrary to the developer's intention. As a result, the user can potentially mount harmful security attacks using similar methods.

Cross-Site Scripting: Example

To prevent XSS vulnerability, change the code in the dynamic PL/SQL region to escape the text sent to the browser:

```
htp.p(htf.escape_sc(:P<n>_H));
```



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When the text is escaped, the information entered in the URL is displayed in the browser window.

Specifying Browser Security

The screenshot shows the 'Edit Security Attributes' interface for an application named 'Application 20031'. The 'Security' tab is selected. Under the 'Browser Security' tab, there are four configuration options:

- Cache:** Set to 'Disabled'.
- Embed in Frames:** Set to 'Deny'.
- HTML Escaping Mode:** Set to 'Extended'.
- HTTP Response Headers:** An empty text area for specifying additional headers.

At the bottom right of the interface are 'Cancel' and 'Apply Changes' buttons.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can specify additional browser security settings by editing an application's properties. To do so, perform the following steps:

1. From the development home page of an application, click the Edit Application Properties button displayed at the top right corner.
2. Click the Security tab.
3. Click the Browser Security tab.

Here, you can specify the following:

- **Cache:** Select "Enabled" to save the contents of pages for this application in its cache, both in memory and on disk.
- **Embed in Frames:** You can specify the following options:
 - **Deny:** The page cannot be displayed in a frame.
 - **Allow from same origin:** The page can only be displayed in a frame on the same origin as the page itself.
 - **Allow:** The page can be displayed in any frame.
- **HTML Escaping Mode:** You can select the Basic (&, “,<,>) or the Extended (&, “,<,>,’,/ and non-ASCII characters) options.
- **HTTP Response Headers:** You can specify additional application-specific HTTP headers.

What Is a Directory?

- A directory is:
 - A special purpose, distributable database
 - Entry oriented
 - Used for storing and retrieving entries
- Applications that use directory services include:
 - Email address books
 - Corporate white pages
 - Centralized applications for managing credentials and privileges
 - Applications that configure and manage system resources



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

An online directory is a specialized database that stores and retrieves collections of information about objects. The information can represent any resources that require management; for example: employee names, titles, security credentials, information about partners and information about shared resources, such as conference rooms and printers.

The information in the directory is available to different clients, such as single sign-on solutions, email clients, and database applications. Clients communicate with a directory server by means of the LDAP. The data that is stored in the directory is mostly read or searched, and is seldom modified. The data is stored in the directory in the form of entries.

The applications that use online directories include the following:

- The email address book is perhaps the most familiar directory application. Most email systems provide a way for users to search for email addresses by giving a username.
- Directory services can be used to deploy an online white pages directory service. The idea is to replace paper-bound corporate directories with an online service that is easily searchable and always up-to-date.
- A centralized system for applications to manage credentials and privileges.
- Applications that can be configured to manage system resources.

Oracle Internet Directory

Oracle Internet Directory is an LDAP v3 directory with the following characteristics:

- Scalability
- High availability
- Information security
- Integrated management



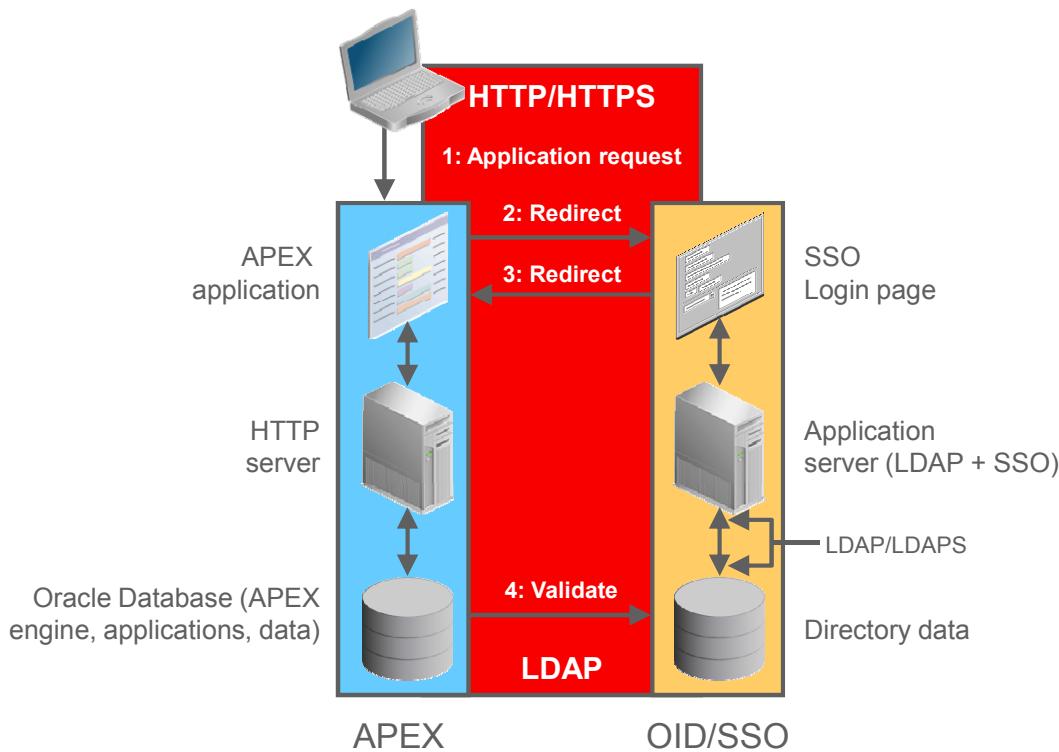
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Internet Directory is an LDAP directory that uses an Oracle Database for storage. Oracle Internet Directory (OID) is Oracle's LDAP v3-compliant directory. OID has the following features:

- **Scalability:** Scalability is measured by the number of entries that can be supported by a single server instance and the number of simultaneous client accesses supported by the server. OID excels in both respects by providing millisecond response times when servicing tens of thousands of concurrent connections, to a terabyte data store representing over half-a-billion real-world directory entries.
- **High availability:** OID supports both LDAP fan-out and Advanced Symmetric Replication Services (ASR)-based multimaster replication to create highly available and distributed deployment topologies.
- **Information security:** OID natively supports both password- and certificate-based authentication and provides integrated access control list support. OID also supports external authentication for environments where user passwords are stored in a third-party directory (for example, Microsoft Active Directory) and are not synchronized with OID.
- **Integrated management:** OID integrates with Oracle Enterprise Manager Grid Control to provide centralized management of OID with other enterprise applications and resources.

Technical Architecture: Overview



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ORACLE®

1. Users specify a browser request by logging in to the Application Express (APEX) page. This request is transmitted from the browser to the HTTP server by using HTTP or HTTPS protocol. HTTPS is a secure version of HTTP, often referred to as SSL. The HTTP server forwards requests to the Application Express engine. The APEX engine determines from the request which application and page are being requested. The APEX engine then fetches the application metadata from the Oracle Database and proceeds with the authentication step.
2. If the application is set up to use Single Sign-On (SSO), the application redirects to the SSO login page. Users view the SSO Login page in their browser, enter a username and password, and click the Login button. The SSO login server (part of an Application Server instance) then verifies the user's credentials by checking them in the LDAP directory, OID for the Oracle SSO facility. This communication takes place using LDAP or LDAPS protocol, the latter being a more secure version of LDAP protocol.
3. After authentication, the login server redirects to the application page that the user requested originally. This page is now displayed on the browser. As the user (now logged in) continues to interact with the application, HTTP/HTTPS requests flow from the browser to the HTTP server and then to the Application Express engine in the database. The APEX application (via the APEX engine) returns HTTP responses through the HTTP server back to the browser.

4. During the execution of various parts of an application, the application may access the LDAP (OID) directory to obtain information about the authenticated user's roles, privileges, or group membership. This information is used to effect authorization rules within the application, such as controlling which components the user may view or operate.

Authenticating Users by Using LDAP

1. Create users in Oracle Internet Directory.
2. Find the distinguished name (DN) of the users in Oracle Directory Services Manager.
3. Create an LDAP authentication scheme in Oracle APEX. By default, this scheme becomes your current authentication scheme.
4. Run the application and log in as an Oracle Internet Directory user.

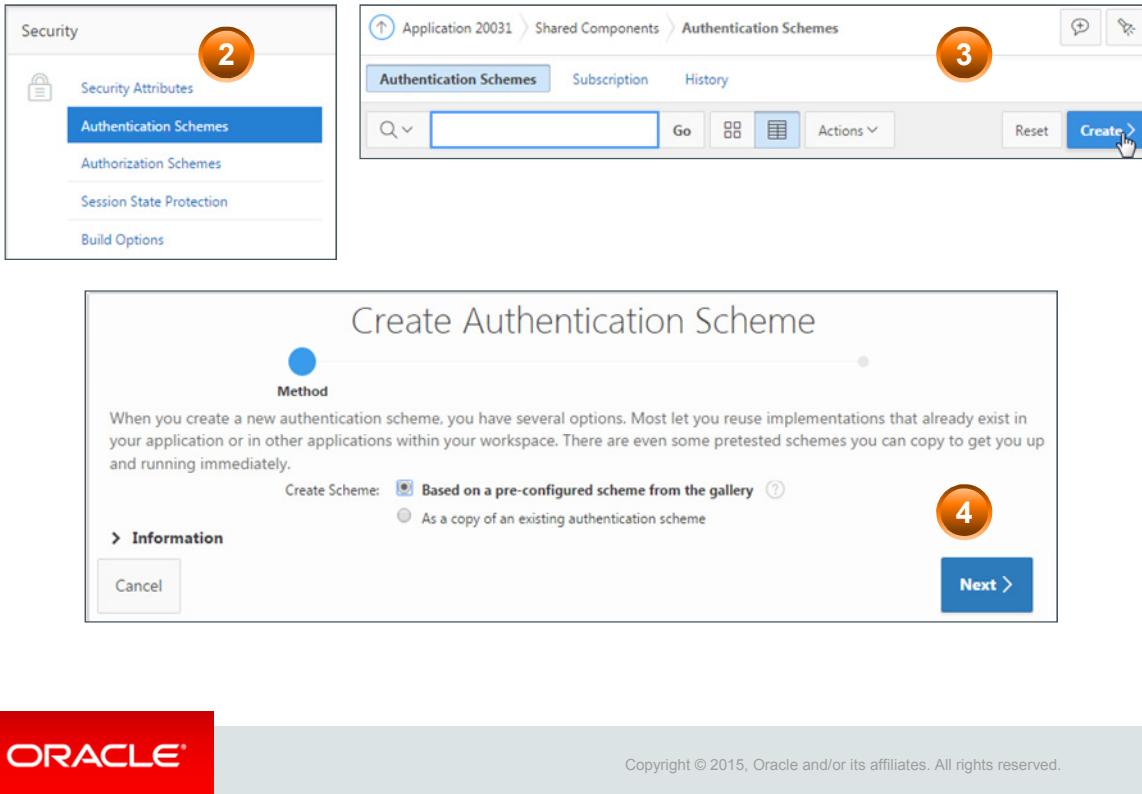


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Application Express offers some built-in credentials verification methods that use DBMS_LDAP to access the LDAP directory that you specify in the authentication scheme attributes. To authenticate a user using LDAP credentials:

1. Create the users you want to have access in your Oracle Internet Directory.
2. Open Oracle Directory Services Manager to find the distinguished name of the user that you will use when you create your authentication scheme.
3. Create an authentication scheme in Oracle Application Express and select Show Login Page and User LDAP Directory Credentials. By default, this authentication scheme becomes your current authentication scheme.
4. Run your application and log in as one of the Oracle Internet Directory users.

Creating an LDAP Authentication Scheme in Oracle APEX



To create your LDAP authentication scheme:

1. Navigate to your application and select Shared Components.
2. Under Security, select Authentication Schemes.
3. Click Create.
4. Select “Based on a pre-configured scheme from the gallery” and click Next.

Creating an LDAP Authentication Scheme in Oracle APEX

7

The screenshot shows the 'Authentication Scheme' creation screen in Oracle APEX. The 'Name' field is populated with 'LDAP-OID' (step 5). The 'Scheme Type' dropdown is set to 'LDAP Directory' (step 5). The 'Host' field contains 'localhost' (step 6). The 'Port' field is set to '3060' (step 6). The 'Use SSL' dropdown is set to 'No SSL'. The 'Distinguished Name (DN) String' field contains 'cn=%LDAP_USER%,cn=Users,dc=us,dc=' (step 6). The 'Username Escaping' dropdown is set to 'Standard'. The top right corner has a large orange circle with the number '7'.

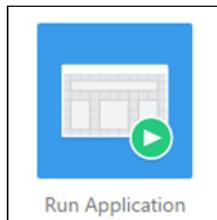
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

5. Enter a Name and select the Scheme Type as LDAP Directory.
6. Enter the LDAP host and port number, enter the LDAP Distinguished Name (DN) String field. Note that %LDAP_USER% is inserted into the DN string as a placeholder for the username when the authentication takes place.
7. Click Create Authentication Scheme.

By default, the newly created scheme becomes the current Authentication Scheme.

Running the Application and Logging In as a User in Oracle Internet Directory



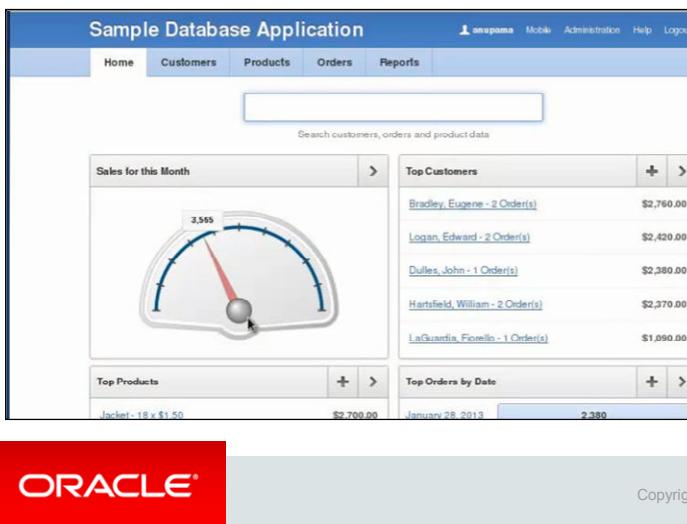
Sample Database Application

Username
Anupama

Password

Login

To log in to the Sample Database application, please use your Application Express Workspace username and password.
For further information, please refer to "Utilizing Packaged Applications" in the Oracle Application Express Application Builder User's Guide.



Enter the username you created in OID.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Run the application to test your LDAP authentication scheme:

1. Click Run Application.
2. Log in as an Oracle Internet Directory user and click Login.

Summary

In this lesson, you should have learned how to:

- Prevent SQL Injection Attacks
- Prevent cross-site scripting
- Apply application-level security
- Authenticate users by using LDAP



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you should have learned how to prevent SQL Injection Attacks and cross-site scripting, and authenticate users using LDAP.

Jack has gone through the code in the PTS application and ensured that there are no vulnerabilities in the application for SQL Injection and cross-site scripting attacks.

Practice 11-1 Overview: Securing an Application

This practice covers the following topics:

- Preventing SQL Injection Attacks



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

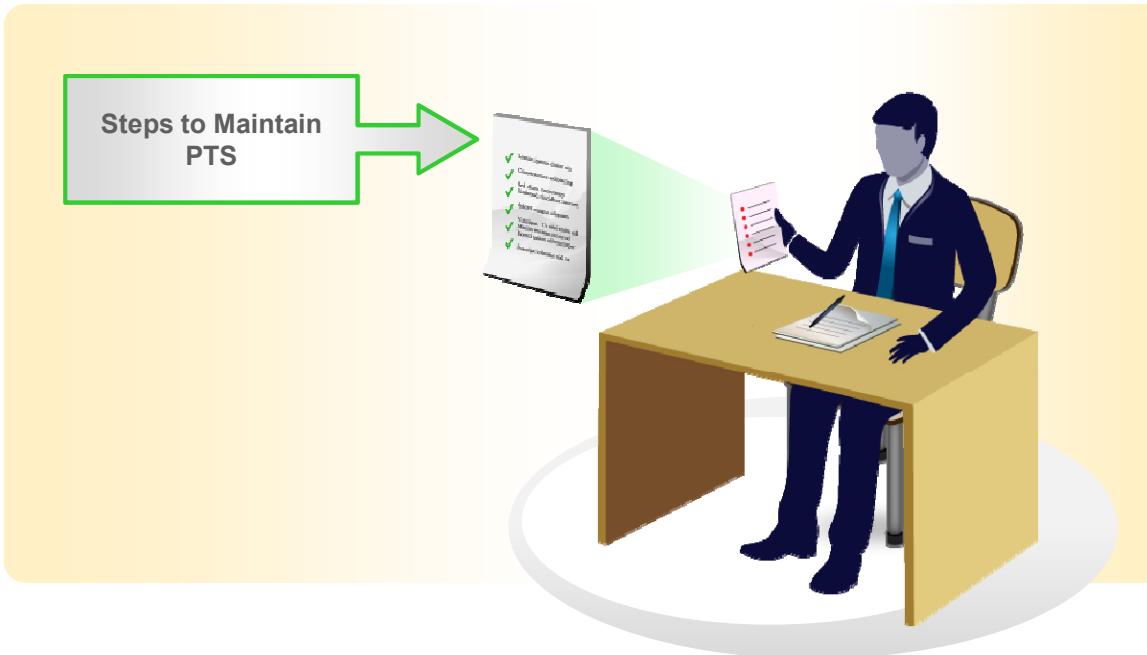
12

Deploying an Application

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

PTS Scenario



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack is all set to move the Project Tracking System (PTS) to the production server. He is been requested by Jill to document the steps required to back up and export the application. This will ensure that any new developer who is assigned to work on the PTS in Jack's absence will be able to maintain the application with ease.

Objectives

After completing this lesson, you should be able to:

- Identify the supporting objects for an application
- Export an application and its supporting objects
- Import an application
- Install the supporting objects



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This lesson shows you how to manage supporting objects for an application by defining prerequisites and uploading scripts. You then export the application, and import and install it into another Oracle Application Express instance.

Steps to Deploy an Application

- 1.** Create a packaged application.
 - Identify the application's supporting objects.
 - Manage the supporting objects' definition.
 - Export the application.
- 2.** Import the packaged application.
- 3.** Install the packaged application.
- 4.** Publish the URL.

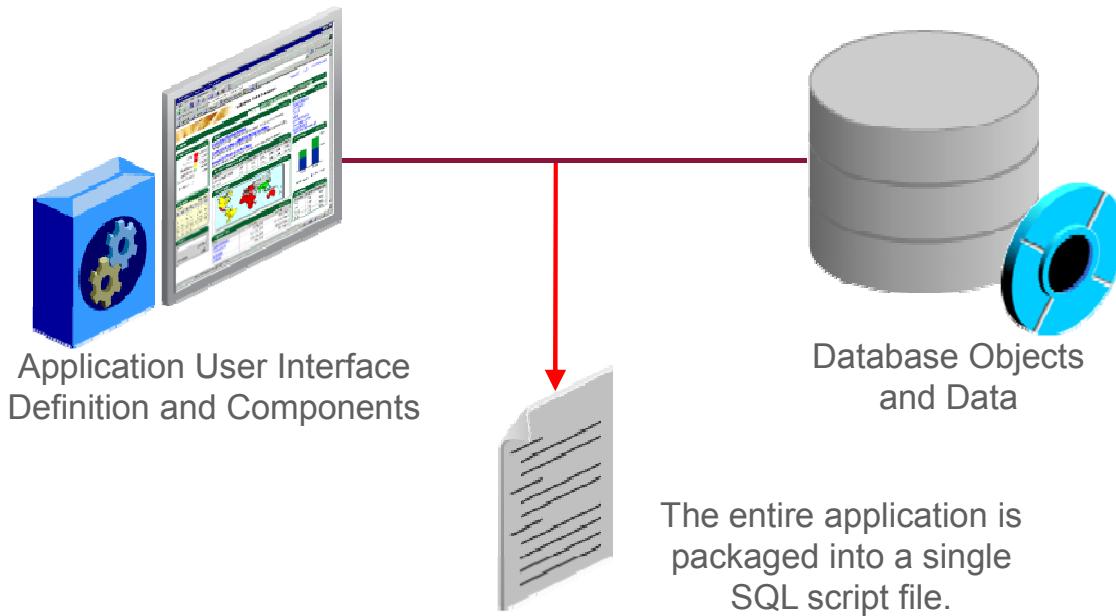


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When you develop an application in Application Builder, you create the application within a specific workspace. Each workspace has a unique ID and name. A common scenario is that you create an application in a development instance and deploy it to a production instance. The steps to deploy an application are listed in the slide.

What Is a Packaged Application?

A packaged application simplifies the process of deploying an application.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A packaged application is a single SQL file that contains information about an application and its components, as well as information about the database objects and the data in them. This single SQL file can then be imported and installed into another Oracle Application Express instance.

What Are Supporting Objects?

Supporting objects are the database objects and data needed to run an application successfully.

Application home page



Supporting Objects

Application 23159 > Supporting Objects

Supporting Objects

Use this utility to define the database object definitions, images, and seed data to be included in your application export.

Installation	Upgrade	Deinstallation
Prerequisites Application Substitution Strings 0 Build Options 0 Pre-installation Validations 0 Installation Scripts 4 Messages	Upgrade Scripts 5 Upgrade Message	Deinstallation Script 1 Deinstallation Message

Application: 23159: Sample Database Application ⓘ Check for Objects: No ⓘ
Verify System Privileges: Yes ⓘ Required Free KB: 100 ⓘ
Prompt for License: No ⓘ Include in Export: Yes ⓘ

ORACLE®

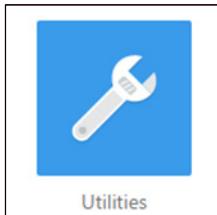
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Supporting objects are the database objects needed by an application to run successfully. These include database object definitions, images, and seed data. The instance to which you are trying to deploy your application may already have all the supporting objects. In this case, you do not have to export and install them.

To manage supporting objects for an application, navigate to the application's home page and click the Supporting Objects icon. From the Supporting Objects page, you can create and manage the scripts required to install, upgrade, or uninstall an application.

Identifying the Supporting Objects for an Application

1



2



Database Object Dependencies
Review the database objects referenced by this application.

3

Compute Dependencies					
Application	20922	Owner	Referenced Name	Referenced Type	Reference Count
APEXWSI	20922	APEXWSI	EMPLOYEES	Table	1
			PROJECT_MEMBERS	Table	1
APEX_050000	20922	APEX_050000	WWV_FLOW_GLOBAL	Package	2
SALOME	20922	SALOME	APEX_ACCESS_CONTROL	Table	4
			APEX_ACCESS_SETUP	Table	4
			DOCUMENT_TYPES	Table	1
			EMPLOYEES	Table	4
			PROJECT_DOCUMENTS	Table	1
			PROJECT_MEMBERS	Table	1
			PROJECT_TYPES	Table	1
			STATUS	Table	1
			APEX_ACCESS_CONTROL	%	2
			APEX_ACCESS_SETUP	%	2
			DOCUMENT_TYPES	%	2
			EMPLOYEES	%	5
			PROJECTS	%	4
			PROJECT_ACTIONITEMS	%	2
			PROJECT_DOCUMENTS	%	2
			PROJECT_MEMBERS	%	2

The Oracle logo, consisting of the word "ORACLE" in a white sans-serif font inside a red rectangle.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To identify which database objects an application uses, you can run the Data Object Dependencies report:

1. Navigate to the application's home page and click the Utilities icon.
2. Click the Database Object Dependencies icon.
3. Click the Compute Dependencies button.

A report is displayed, listing the database objects used by the application. These are the application's supporting objects.

Creating Installation Scripts

The screenshot shows the 'Installation' pane of the Supporting Objects page. It includes sections for Prerequisites, Application Substitution Strings (0), Build Options (0), Pre-installation Validations (0), Installation Scripts (4), and Messages.

Installation	
	Prerequisites
Application Substitution Strings	0
Build Options	0
Pre-installation Validations	0
Installation Scripts	4
Messages	



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

From the Installation pane in the Supporting Objects page, you can:

- Specify any prerequisites for the installation
- Specify substitution strings used in the application
- Specify build option for the application pages
- Verify the validations to be performed before installing the application
- Create the installation scripts
- Enter messages to be displayed during installation. The supported HTML tags include ``, `<i>`, `<u>`, `<p>`, `
`, `<hr>`, ``, ``, ``, and `<pre>`.

Specifying Prerequisites and Other Options

The screenshot shows the 'Prerequisites' tab of the Oracle Application Express configuration interface. At the top, there are tabs for 'Messages', 'Prerequisites' (which is selected), 'Substitutions', 'Build Options', 'Validations', 'Install', 'Upgrade', 'Deinstall', and 'Export'. Below the tabs, the title 'Prerequisites' is displayed, along with 'Cancel' and 'Apply Changes' buttons. A 'Required Free Space in KB:' input field contains the value '100'. Under 'Required System Privileges:', several checkboxes are listed: 'CREATE DATABASE LINK', 'CREATE SEQUENCE', 'CREATE TRIGGER' (which is checked), 'CREATE MATERIALIZED VIEW', 'CREATE SYNONYM', 'CREATE TYPE', 'CREATE PROCEDURE' (which is checked), 'CREATE TABLE' (which is checked), and 'CREATE VIEW' (which is checked). A section titled 'Objects that will be Installed' contains a note about avoiding errors due to pre-existing objects and a list of object names with an 'Add' button.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can check for disk space, sufficient privileges, or existing table names before running the installation scripts. From the Supporting Objects page, click the Prerequisites link. From the **Prerequisites** tab, you can define built-in checks that are performed before installation. If prerequisites are defined, before installation, the installer checks whether there is enough space to create all the objects that are needed. It checks that the system privileges are set accordingly and that the list of objects does not exist.

You can use the **Messages** tab to enter the messages that you want to display during installation and deinstallation.

The **Substitutions** tab lists static substitution strings defined for the application. You can specify the substitution strings that a user can define while installing.

The **Validations** tab lists validations defined for the packaged application. These validations prevent a user from installing database objects if the defined conditions are not satisfied. On the **Export** tab, you can specify whether the deployment attributes should be exported with your application by default.

Specifying Build Options

The screenshot shows the 'Build Options' tab selected in the navigation bar of the Oracle Application Express interface. The main content area displays a message: 'Identify the build options that a user may change when installing this application.' Below this message, another message states: 'This application has no build options.' A 'Cancel' button is located in the top right corner of the content area.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

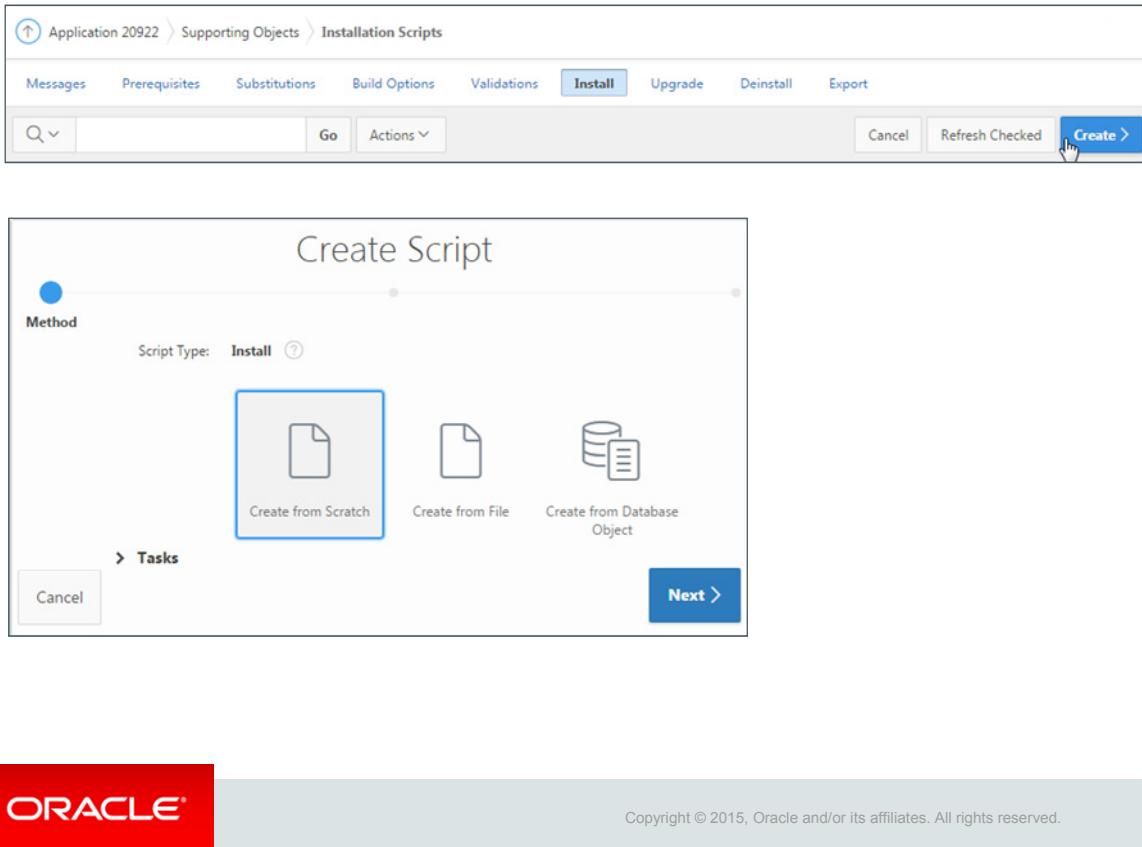
The **Build Options** tab lists build options defined for this application. Build options allow you to hide or show specific functionality within an application. You can apply build options to an entire page or specific components of the page.

To create a build option, navigate to the Shared Components page and click the Build Options link under Security. Click the Create button. Specify a name and select a status for the build option. The available build options are `INCLUDE` and `EXCLUDE`. As the name specifies, you can select an option depending on whether you want to include or exclude a functionality.

After creating the build option, you can apply it to a page or component. Edit the page or component and specify the build option (located on the configuration tab).

From the Build Options tab under Supporting Objects, you can specify the build options that should be included in the export file. These options can be changed by the user while installing the application.

Creating an Installation Script



The Install tab enables a developer to define multiple installation scripts that install the supporting application objects. The DDL for these scripts can be created by using the DDL Generator under Utilities. To create the DML script, you can use Oracle SQL Developer (a free downloadable tool from <http://otn.oracle.com>) or you can create it from the beginning. To add a script to the list, click the Create button.

- If you select the “Create from Scratch” option, you must enter the SQL code that you want to use for installation. You have an option to use the Script Editor.
- If you select the “Create from File” option, you can browse and select a script file that you want to use for installation.
- If you select the “Create from Database Object” option, you can declaratively select the database object that you want to include in the script.

You also have links to create installation scripts for Access Control tables your application uses. These are displayed under the Tasks link.

- If you click the “Create Scripts for Access Control Tables” link, the scripts to install the tables, populate the tables with data, and uninstall the tables can be automatically created. You can review the tables that will be created and click the Create Script button.

Creating Upgrade Scripts

The screenshot shows the Oracle Application Express Upgrade interface. At the top left is a sidebar titled 'Upgrade' with options for 'Upgrade Scripts' (selected, showing 5 results) and 'Upgrade Message'. The main area has a breadcrumb path: Application 20922 > Supporting Objects > Upgrade. The tabs at the top include Messages, Prerequisites, Substitutions, Build Options, Validations, Install, **Upgrade**, Deinstall, and Export. The 'Upgrade' tab is active. Below the tabs is a search bar with a magnifying glass icon and a 'Go' button. A message states 'No upgrade scripts found.' A section titled 'Detect Existing Supporting Objects' contains a note about identifying supporting objects and includes a 'Query to Detect Existing Supporting Objects' editor with a toolbar and a single row of data (ID 1). Buttons for 'Cancel', 'Create >', and 'Apply Changes' are visible.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use the Upgrade page to define scripts to upgrade database objects, images, and seed data when upgrading an existing application. You specify a query to run to detect whether an existing supporting object exists. If a row is returned, the script is executed. To create the upgrade scripts, click the Create button on the Upgrade tab.

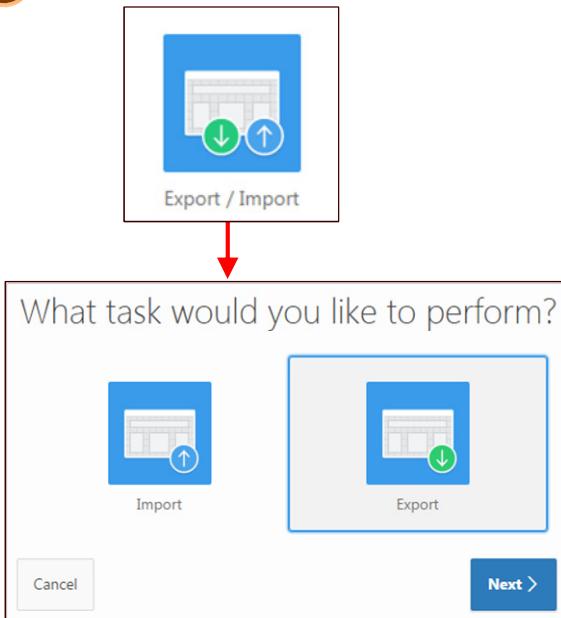
Creating Deinstallation Scripts

The screenshot shows the Oracle Application Express interface for creating a deinstallation script. At the top, there's a navigation bar with tabs: Messages, Prerequisites, Substitutions, Build Options, Validations, Install, Upgrade, Deinstall (which is highlighted in blue), and Export. Below the navigation bar, the title 'Deinstall Script' is displayed, along with 'Cancel' and 'Create >' buttons. A main text area contains the instruction: 'Define a script to drop database objects and static files that are created by the installation scripts.' At the bottom of the interface, there's an 'ORACLE' logo and a copyright notice: 'Copyright © 2015, Oracle and/or its affiliates. All rights reserved.'

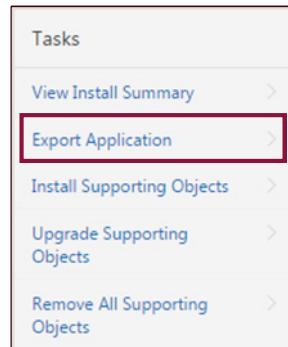
The Deinstall tab enables you to define a deinstallation script that runs when a user clicks the Deinstall option. In this script, you specify the DROP commands to drop objects and operations performed in the installation scripts. To create a deinstallation script, click the Create button on the Deinstall tab.

Accessing the Export Page

- a From the Application home page



- b From the Supporting Object page



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To move your application from a development instance to a production instance of Oracle Application Express, you can export the application definition and all the supporting objects to a file. You can export your application by performing either of the following actions:

- From the Application home page, click the Export/Import icon. Select Export and click Next.
- From the Manage Supporting Object Definitions page, select Export Application from the Tasks section.

Exporting an Application

The screenshot shows the 'Export Application' page in Oracle Application Express. The application selected is 'Project Tracking System_V1'. The export preferences are set to include supporting object definitions ('Yes'). Other options like 'Export Public Interactive Reports' and 'Export Private Interactive Reports' are also configured.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When you export your application, all the application attributes, templates, pages, regions, items, buttons, and supporting objects are exported to a single file.

On the Export Application page, perform the following steps and then click Export Application:

- Make sure that your application is selected in the Application list.
- For File Format, select how the rows in the export file will be formatted. Select UNIX for a file containing rows delimited by line-feeds. Select DOS for a file containing rows delimited by carriage returns and line-feeds. Select Database to save the file as a normal SQL script with the `.sql` extension.
- For Owner Override, select an optional overriding owner for this application.
- For Build Status Override, select Run Application Only if you want to run the application in the target instance but want to make it inaccessible to developers. Selecting "Run and Build Application" makes it available to developers.
- For Debugging, select Yes to enable debugging.
- Use "As of" to export your application as it was previously defined.
- Set the export preferences.

In addition to exporting the actual application file, you can also use Export to export other related files, such as cascading style sheets, images, plug-ins, and script files.

Quiz



Before the application installation can proceed, the installer should check whether there is enough disk space to create all the objects that are needed. Where would you specify the required free space?

- a. Supporting Objects > Build Options
- b. Supporting Objects > Validations
- c. Supporting Objects > Install
- d. Supporting Objects > Prerequisites



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: d

Importing an Application

1

2

3

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

After you export your application and supporting files from the development environment, you can import the file into the target Oracle Application Express workspace, and then install the application in the new environment. To import the application file:

1. Navigate to the Application Builder page and click the Import button.
2. Browse and locate the file that you have previously exported from Application Express. Click Next.
3. You get a confirmation message that the file was imported successfully. Click Next to install the file. You may also choose to install the application at a later time by accessing the Export Repository.

Installing the Application

Install Database Application

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

Current Workspace: SALOME ⓘ
 Export File Workspace ID: 222973639628611653 ⓘ
 Export File Application ID: 14594 ⓘ
 Export File Version: 2013.01.01 ⓘ
 Export File Parsing Schema: APEXWSI ⓘ
 Application Origin: This application was exported from another workspace. ⓘ
 Parsing Schema: SALOME ⓘ
 Build Status: Run and Build Application ⓘ
 Install As Application: Auto Assign New Application ID (selected) ⓘ
 Reuse Application ID 14594 From Export File ⓘ
 Change Application ID ⓘ

> Tasks Cancel Install Application

Install Application

Supporting Objects
 This application installer will guide you through the process of creating your database objects and seed data.

Application: 20031 - Project Tracking System_V1 ⓘ
 Parsing Schema: SALOME ⓘ
 Free Space Required in KB: 100 ⓘ
 Install Supporting Objects: Yes ⓘ

> Tasks Cancel Next >

Application Installed

Your imported file is located in the [Export Repository](#). Unless you plan to install it again, you should remove it.

Edit Application Run Application

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

After you import the packaged application file into the target Oracle Application Express instance, you can install it. To install the imported file:

1. Select the Schema where you want to install the application and its supporting objects. Click **Install**.
2. Specify whether you want to install the supporting objects for the application and click **Next**.
2. The application is installed successfully.

These steps continue from the wizard steps discussed in the previous slide. You can also install an imported file by navigating to the Export Repository and clicking the **Install** link next to the file.

Publishing the Application URL

`http://app.oracle.com:8080/apex/f?p=<app_id>`

Provide this URL
to end users of the application
if your setup uses the
embedded PL/SQL gateway or
the APEX Listener.

`http://app.oracle.com/pls/apex/f?p=<app_id>`

Provide this URL
to end users of the application
if your setup uses Oracle
HTTP Server with `mod_plsql`.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

After you have deployed your Application Express application, you can provide its URL to the end users.

A typical Application Express URL looks like the following:

`http://app.oracle.com/pls/apex/f?p=<app_id>`

- `http://app.oracle.com` is the URL of the server.
- `pls` is the indicator to use `mod_plsql`.
- `apex` is the DAD name.
- `f?p=` is the prefix used by Application Express.
- `<app_id>` is the application number of the application being called.

The application will automatically redirect the user to the appropriate home page (as defined in the application attributes). Also, if the application requires authentication, the user will be redirected to the Login page.

If you want to protect against changing an application ID in the future, you can define an alias for your application and use that in the published URL (for example, `f?p=SALES`). The alias is set on the Application Definition page.

Quiz

Q

Which of the following can you export by using the Export/Import utility? (Select all that apply.)

- a. Application
- b. Uploaded cascading style sheets
- c. Uploaded images

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c

Summary

In this lesson, you should have learned how to:

- Identify the supporting objects for your application
 - Export your application
 - Import your application
 - Install the supporting objects



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learned how to manage supporting objects for your application by defining prerequisites and uploading scripts. You also learned how to export the application, and import and install it into another Oracle Application Express instance.

Jack has successfully exported the PTS application and made it ready to be moved to the production server.

Practice 12 Overview: Deploying and Maintaining Your Application

This practice covers the following topics:

- Identifying which supporting database objects you want to export into a single file
- Exporting an application



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this practice, you package the GMT application.

13

Optimizing Your APEX Application

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Optimizing the PTS Application



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack has completed developing the PTS application. He will now learn how to optimize the performance of the PTS application and improve it.

Objectives

After completing this lesson, you should be able to:

- Manage your services
- Monitor the activity of your application
- Monitor and improve the performance of your application



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learn about application performance considerations. You learn about Managing Services and monitoring application activity, as well as monitoring and improving the performance of your application.

Application Performance Considerations

- Enable Caching
- Place image files on file system rather than upload into APEX
- Minimize number of JavaScript files
- Minify JavaScript and CSS files
- Create a sprite image combined into one image file
- Refer CSS files in the header tags and JavaScript in the HTML file
- Make XHR response small



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

For applications having a large number of concurrent users, maintaining optimal performance is critical. Some application tuning performance considerations are:

- Enable region caching to improve the performance of static regions, such as regions containing lists that do not use conditions, or regions containing static HTML. When you enable region caching, the Application Express engine renders a region from a cached (or stored) repository instead of rendering it dynamically.
- Place image files on the file system rather than upload into APEX for faster image access and to save database storage space.
- Minimize the number of JavaScript files for faster page loading.
- Minify JavaScript and CSS files to reduce the size of the files, which will, in turn, load the files faster. APEX provides a #MIN# substitution string to load the minified version of the files.
- Create a sprite image that can be combined into one image file and then access the appropriate image using CSS sprites. APEX does use CSS sprites in its own themes and widgets.
- CSS file reference should be within the header tags and the JavaScript should be at the bottom of the HTML file.
- XHR (XMLHttpRequest) usually used in Ajax should be small.

Managing Services

The screenshot shows the 'Manage Service' page. It includes the following sections:

- Make a Service Request:** Request access to a database schema, additional storage, or to terminate the workspace.
- Set Workspace Preferences:** Configure functionality available to developers within this workspace.
- Edit Message:** Edit the Workspace Message.
- Workspace Utilization:** View the workspace summary report, including schema and activity details.

A dropdown menu titled 'Manage Meta Data' is open, listing the following items:

- Developer Activity and Click Count Logs
- Session State
- Application Cache
- Websheet Database Objects
- Application Build Status
- File Utilization
- Interactive Report Settings

At the bottom left is the Oracle logo, and at the bottom right is the copyright notice: Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use the Manage Service page to manage service requests, configure workspace preferences, edit workspace announcements, and view workspace utilization reports. Additionally, use the links on the Manage Meta Data list to manage the following:

- Developer activity and click count logs
- Session state
- Application cache
- Schema requests
- Websheet database objects
- Application build status
- Application models
- File utilization
- Interactive report settings

To access the Manage Service page:

1. Navigate to the **Administration** page.
2. Click the **Manage Service** icon.

Monitoring Activity

The screenshot shows the Oracle Monitor Activity page. At the top, there are tabs for 'Real Time Activity' (which is selected) and 'Archived Activity'. Below the tabs is a 'View Dashboard' button. The main area is divided into several sections:

- Page Views:** Contains links for 'By View', 'By Application', 'By Application and Page', 'By Day', 'By Hour', and 'By Interactive Report'.
- Developer Activity:** Contains links for 'By Developer', 'By Developer Bar Chart', 'By Day', 'By Application', 'Application Changes, detailed', and 'By Day, Monthly View'.
- Sessions:** Contains links for 'Active Sessions' and 'Bar Chart of Active Sessions by Hour'.
- Page View Analysis:** Contains a link for 'Most Viewed Pages over All Applications'.
- Environment:** Contains links for 'By User Agent' and 'By Browser'.
- Login Attempts:** Contains links for 'Login Attempts' and 'Login Attempts by Authentication Details'.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can monitor developer activity and changes within your workspace by accessing the Monitor Activity page. The Monitor Activity page contains a variety of reports that track changes to page views and applications.

To access the Monitor Activity page:

1. Navigate to the Workspace home page.
2. From the Administration list, click **Monitor Activity**. The Monitor Activity page is displayed.

The Monitor Activity page features activity reports that are divided into the following sections:

- **Page Views:** Contains reports of page views by user, application, browser, or operating system
- **Developer Activity:** Contains reports of developer login attempts and so on
- **Page View Analysis:** Contains reports of the analysis of page views, such as top page views by application
- **Sessions:** Lists active sessions with the current workspace
- **Login Attempts:** Contains reports listing login attempts

- **Environment:** Contains reports of environments organized by user agent, browser, external clicks, or operating system
- **Application Errors:** Contains reports of application errors
- **Workspace Schema Reports:** Contains reports of database privileges by schema, schema tablespace utilization, and so on

Monitoring Activity: Page Views Reports

The screenshot shows the Oracle Application Express Monitoring Activity interface. On the left, a sidebar menu titled 'Page Views' lists several options: 'By View', 'By Application', 'By Application and Page' (which is highlighted with a blue background and a cursor icon), 'By Day', 'By Hour', and 'By Interactive Report'. To the right, the main content area is titled 'Monitor Activity > Page Views by Application and Page'. It features a search bar with 'Since 1 day' and a 'Set' button. Below the search bar is a table with the following columns: Application, Page, Average Elapsed, Distinct Users, Distinct Sessions, Page Events, Partial Page Views, Cached Regions, and Page Name. The table contains ten rows of data, with the first row being the header. The data includes various application IDs (e.g., 4000, 10499) and page numbers (e.g., 4500, 9, 4003, 4761, 1, 6, 1000, 571, 385, 1500, 4150), along with their respective statistics and page names.

Application	Page	Average Elapsed	Distinct Users	Distinct Sessions	Page Events	Partial Page Views	Cached Regions	Page Name
4000	4500	0.447	1	1	26	0	0	Page Designer
4000	9	0.390	1	1	11	0	0	Shared Components
4000	4003	0.969	1	1	9	2	0	Templates
4000	4761	0.505	1	1	9	0	0	Translate Application
4000	1	0.608	1	2	8	0	0	Application Builder
10499	6	0.147	2	1	8	0	0	Projects1
4000	1000	0.083	1	1	6	0	0	Run Page
4000	571	0.525	1	1	5	0	0	Copy Template - New Template
4000	385	5.398	1	1	5	0	0	Seed and Publish Translatable Text
4000	1500	0.336	1	2	5	0	0	Application Builder
4000	4150	0.372	1	1	5	0	0	Page Definition

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Page Views reports provide useful statistics and timing information about all the pages in your application. This page answers the question: What is the most popular page in my application?

Monitoring Activity: Page View Analysis Reports

The screenshot shows the 'Page View Analysis' section of the Oracle Application Express interface. A red arrow points from the 'By Weighted Page Performance' link in the sidebar to the corresponding report page below. The report page is titled 'Monitor Activity > Page Views by Weighted Page Performance'. It includes filters for 'Since' (1 day), 'Application' (- All Applications -), and a 'Set' button. The main area displays a table of page performance metrics:

Application	Page	Page Name	Page Events	Average Elapsed	Weighted Average	Median Elapsed	Weighted Median	Median Content
10499	6	Projects1	8	0.1467	1.1735	0.1397	1.1175	
10499	101	Login Page	2	0.2362	0.4725	0.2362	0.4725	
10499	7	Form on PTS_PROJECTS	2	0.1896	0.3793	0.1896	0.3793	

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The weighted page performance displays the pages having the highest average elapsed times multiplied by the number of page views. As a workspace administrator, this report can help you find the slowest pages that are accessed the most. The developer can then evaluate the pages to understand why performance is so slow and figure out ways to improve performance for those pages identified in this report.

To view this report:

1. From the Oracle Application Express home page, under the Administration task menu, click **Monitor Activity**.
2. Under Page View Analysis, click **By Weighted Page Performance**.

Database Configuration Considerations

- Oracle APEX requires that the shared pool size of the target database be at least 150 MB.
- To check and change SHARED_POOL_SIZE, execute the following commands:

```
sqlplus / as sysdba
STARTUP
SHOW PARAMETER PFILE;
SHOW PARAMETER SHARED_POOL_SIZE;
ALTER SYSTEM SET SHARED_POOL_SIZE='150M' SCOPE=spfile;
SHUTDOWN
STARTUP
```

- If the system uses an initialization parameter file, check the init<sid>.ora file to make sure that the value of SHARED_POOL_SIZE is at least 150 MB.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The Database System Global Area (SGA) is essentially memory that is allocated to an Oracle instance, and is shared among various Oracle processes. These typically comprise several memory components, which are pooled to satisfy a particular class of memory allocation requests. Examples of memory components include the shared pool (used to allocate memory for SQL and PL/SQL execution), the Java pool (used for Java objects and other Java execution memory), and the buffer cache (used for caching disk blocks).

Oracle Application Express requires the shared pool size of the target database to be at least 150 MB. Inadequate memory for SGA automatically impacts the performance of Application Express operations. For instance, insufficient SGA size:

- Degrades the performance of PL/SQL operations
- Impacts the retrieval and processing of Oracle Application Express metadata. The metadata is generated via dynamic PL/SQL.
- Affects performance. Larger cache sizes generally reduce the number of disk reads and writes. However, a large cache may take up too much memory and induce memory paging or swapping.
- Reduces the memory available for buffer cache
- Impacts the available memory for the DB shared pool

Monitoring Application Performance

Reference session state values by using bind variables within your application:

```
select * from (
  select * from employees)
 where (
  instr(upper("LAST_NAME") , upper(nvl (:P<n>_REPORT_SEARCH, "LAST_NAME" ))) > 0 )
```

The screenshot shows a report search interface with a search bar containing 'ba'. Below it is a table with columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, and EMAIL. The data is as follows:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL
204	Hermann	Baer	515.123.8888	HBAER
116	Shelli	Baida	515.127.4563	SBAIDA
167	Amit	Banda	011.44.1346.729268	ABANDA
172	Elizabeth	Bates	011.44.1343.529268	EBATES
106	Valli	Pataballa	590.423.4560	VPATABAL



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can reference session state values by using bind variable syntax in SQL queries and application logic such as PL/SQL executed from processes and validations. Accessing session state by using bind variables is the most efficient way to reference the session state.

In this example, the search string is a page item. If the region type is defined as SQL Query, you can reference the value by using standard SQL bind variable syntax. Using bind variables ensures that parsed representations of SQL queries are reused by the database, optimizing memory usage by the server.

Monitoring Application Performance: #TIMING# Substitution String

Include the #TIMING# substitution string in the region footer.

The screenshot shows a page structure with a 'Header and Footer' section on the left and a 'Project Tracking System' report on the right. The 'Region Footer' section contains the text 'This page took #TIMING# CPU seconds.' which is highlighted with a red box. The report table has columns: Emp Id, Emp First Name, Emp Last Name, Emp Email Id, Emp Phone, and Emp Mobile. The data rows are:

Emp Id	Emp First Name	Emp Last Name	Emp Email Id	Emp Phone	Emp Mobile
EMP017	Martin	Johnson	martin.johnson@pts.com	3157862402	2221115555
EMP019	Adams	Henry	adams.henry@pts.com	3157862404	3452326789
EMP021	Miller	Emanuel	miller.emmanuel@pts.com	3157862406	2321213333

At the bottom of the report, another instance of the 'This page took #TIMING# CPU seconds.' message is highlighted with a red box, showing the actual execution time of 0.09 seconds.

You can include the #TIMING# substitution string in the region footer so that you can view the timing of each region. This is the wall clock time and not the time taken to render the region.

Monitoring Application Performance: Object Reports

The Utilities menu is displayed. The Object Reports option is highlighted with a blue background and a white mouse cursor icon pointing to it. Other options include Data Workshop, Generate DDL, Methods on Tables, Schema Comparison, and Object Reports.

The Object Reports interface is shown. It includes sections for Table Reports (Table Columns, Table Comments, Table Constraints, Table Statistics, Table Storage Sizes) and Security Reports (Object Grants, Column Privileges, Role Privileges, System Privileges). Below these are PL/SQL Reports (Program Unit Arguments, Unit Line Counts, Search PL/SQL Source Code) and All Object Reports. At the bottom is an Exception Reports section (Tables without Primary Keys, Tables without Indexes, Unindexed Foreign Keys, Tables without Triggers).

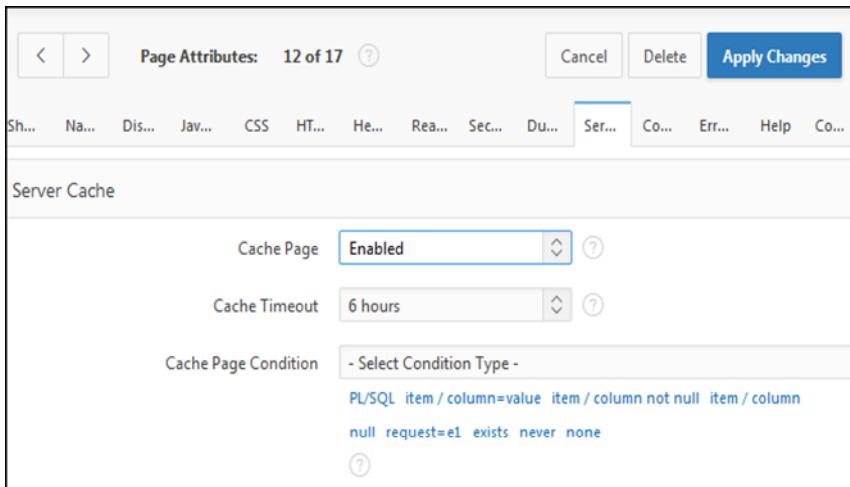
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Under **SQL Workshop > Utilities**, there are several object reports that help identify bottlenecks in performance. For example, you can view tables without primary keys, indexes, or triggers and evaluate whether creating these objects enhances application performance. In addition, you can view a table's statistics and storage size to make sure they are optimized.

Monitoring Application Performance: Caching

Caching reduces the time taken to render page or region. If cached, the page or region is rendered from a cached repository instead of being rendered dynamically.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can improve the performance of your application by using page and region caching. Page caching works well for static pages. Region caching is a good choice for regions such as lists that do not have any conditions or regions containing HTML text. When you enable caching, the Application Express engine renders the page or region from a cached (or stored) repository instead of rendering it dynamically. Note that when you make a change to a page that has been cached, you must purge the cache before the page is rendered with the change.

To enable caching:

1. Navigate to the appropriate page definition or region definition.
2. Click the **Server Cache** tab.
3. For Cache Page, select Enabled. For region caching, select Cached for the Caching field. You can also specify the following:
 - a) **Timeout Cache After:** Identify how long the cached region remains valid.
 - b) **Cache Condition Type:** Select a condition type from the list. If the condition returns false, the region is rendered dynamically and is not cached. If the condition returns true, the region is cached.
 - c) **Expression 1 and Expression 2:** Enter values based on the specific condition type.

Actual session identifiers are not cached. Instead, the Application Express engine caches a &SESSION. substitution string, and the current session rendering the cached region is substituted on display. For example, if a region contains a link and the link includes a session, the exact session is not cached to ensure that the link works for all sessions.

Monitoring Application Performance: Tracing Your Session

1. In APEX, specify &p_trace=YES in URL.

```
http://<hostname>:<port>/apex/f?p=100:101&p_trace=YES
```

2. Navigate to the trace directory and execute the TKPROF utility.

```
tkprof <tracefilename.trc> <tracefilename>.prf SORT=PRSDSK,  
EXEDSK, FCHDSK PRINT=10
```

3. View the TKPROF analysis output.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Tracing your session can be a very effective way to debug an application. From a database perspective, each page request is a single database session. If you enable SQL tracing, Oracle Application Express creates a temporary file that you can then analyze by using the TKPROF utility.

Users can enable SQL tracing in Oracle Application Express by using f?p syntax to set the p_trace=YES argument. For example, to trace the display of page 1 in application 100, use the syntax in the slide. You cannot trace unless it is enabled at the instance level and if the Debug flag for the application is set to Yes.

Note: The Trace feature is not enabled by default in Oracle Application Express.

To use the TKPROF utility:

1. Log in to SQL*Plus as a privileged user (that is, SYSTEM).
2. Execute the following statement:

```
show parameter USER_DUMP_DEST
```
3. Navigate to the directory in which the trace file is created.
4. Run the TKPROF utility from the operating system prompt. The statement in the slide is an example.

For more information about using TKPROF in Oracle Application Express, see “Debugging an Application” in the *Oracle Application Express User’s Guide*.

Monitoring Application Performance: Database Monitor

The screenshot shows the Oracle Application Express interface. At the top, there are tabs for Application Builder, SQL Workshop (which is selected), Team Development, and Packaged Apps. Below the tabs, under the Utilities section, there are several icons and descriptions. One icon, 'Database Monitor', is highlighted with a red box and a cursor arrow pointing to it. The 'Database Monitor' description reads: 'Run database activity reports. Note: A Database account granted the DBA role is required.' To the right of this main window, a smaller secondary window titled 'Database Monitor' shows a list of activity categories: Sessions, System Statistics, Top SQL, and Long Operations.

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The reports available on the Database Monitor page provide a database-wide view of the database sessions, system statistics, SQL statements, and longer operations. You can use these reports to identify poorly performing SQL and better understand the workload of the database.

Note: To access any of the icons on the Database Monitor page, you must have an account that has been granted a database administrator role.

To access the Database Monitor Activity page:

1. Navigate to the workspace home page and click **SQL Workshop**.
2. Click **Utilities**.
3. Click **Database Monitor**.

Sessions: A session is the connection of a user to an Oracle Database instance. A session lasts from the time the user connects until the time the user disconnects or exits the database application.

System Statistics: The System Statistics page displays statistics for Physical I/O, Logical I/O, Memory Statistics, Time Statistics (Seconds), SQL Cursor Statistics, and Transaction Statistics.

Top SQL: The “top” SQL statements represent the SQL statements that are executed most often, that use more system resources than other SQL statements, or that use system resources more frequently than other SQL statements.

Long Operations: The Long Operations page displays the status of various operations that run for longer than six seconds (in absolute time).

Monitoring Application Performance: Locks

Locks prevent destructive interaction between transactions accessing the same resource.

Status	SID	Instance ID	Username	Osuser	Client Info	Module	Object Owner	Object Name
	142	1	ORA01	oracle	-	SQL*Plus	ORA01	?p=4500:1001:3439112722195:FOCUS::OB_OBJECT_ID,quot;>OEHR_EMPLOYEES
	30	1	APEX_PUBLIC_USER	oracle	ORA01_ADMIN:2715024486760901	APEX - SQL Workshop - ORA01 No Autocommit		?p=4500:1001:3439112722195:FOCUS::OB_OBJECT_ID,quot;>OEHR_EMPLOYEES



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Locks are mechanisms that prevent destructive interaction between transactions accessing the same resource.

To view the locks report:

1. Navigate to the workspace home page and click **SQL Workshop**.
2. Click **Utilities**.
3. Click **Database Monitor**.
4. Click **Sessions**.
5. Click the **Locks** tab to view the Locks report.

The Locks report displays a report of sessions that have locks that are blocking other session(s). To control the number of rows that appear, make a selection from the Rows list and click Go.

Monitoring Application Performance: Tuning SQL

- Use bind variables whenever possible.
- Ensure that an optimal query plan is being used for the query.
- Enable SQL tracing for an entire page view to analyze with the Oracle utility, TKPROF.
- Use the hierarchical PL/SQL profiler to determine which functions contain inefficient PL/SQL code.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The Oracle Application Express engine parses, binds, executes the statement, and fetches the results of the SQL for a reporting region. Use bind variables whenever possible to avoid unnecessary parsing and promote reuse of shared SQL by Oracle Database.

Additionally, ensure that an optimal query plan is being used for the query. Use the Explain Plan from the SQL Commands menu to easily generate plans for a particular query.

For a thorough examination of every element of SQL and PL/SQL on a particular page, you can enable SQL tracing for the entire page view. SQL tracing generates a trace file on the server that you can analyze with the Oracle utility, TKPROF.

In Oracle Database 11g, the hierarchical PL/SQL profiler is available to help you identify the hotspots and performance tuning opportunities in your PL/SQL applications. By using the hierarchical PL/SQL profiler, you can find both function-level execution summary information and detailed parent/children information about each function, which will help find inefficiencies in your PL/SQL code. To find more information about the hierarchical PL/SQL profiler, see “Using the PL/SQL Hierarchical Profiler” in the *Oracle Database Advanced Application Developer’s Guide 11g Release 1 (11.1)*.

DBAs can also effectively diagnose performance problems. Active Workload Repository reports and Active Session History (ASH) reports will give an overall view of system activity and usage.

Monitoring Application Performance: Tuning Page Elements

- Set page processes to run per page (rather than per session).
- Optimize global page components.
- Use declarative conditions.
- Use “Rows X to Y” pagination for reports rather than “Rows X to Y of Z.”



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- **Set page processes to run on a per-page (rather than per-session) basis:** If you are employing application-level processes, you can set the process to run once per-session or per-page view. Keep in mind that using the per-session option for an application process can affect all the other page views in the application. A poorly performing application-level process can affect every page view in an application.
- **Optimize global page components:** Global Page components are rendered on every page, so take special care to optimize all global page logic. Consolidate numerous PL/SQL blocks into packages. If you are writing large PL/SQL blocks on a page, consolidate them into PL/SQL packages, which are then invoked from your application.
- **Use declarative conditions:** Declarative conditions are faster than dynamic SQL and PL/SQL conditions. For example, using an `Item=Value` condition is faster than using the PL/SQL expression `:ITEM=value`.
- **Use “Rows X to Y” pagination for reports that return numerous rows:** “Rows X to Y of Z” takes longer to compute than the simple “Rows X to Y” pagination scheme. With “Rows X to Y of Z,” if your report returns 900 rows, Oracle Application Express needs to fetch all rows to obtain the total row count. However, with “Rows X to Y,” the reporting engine needs to fetch only $Y + 1$ rows.

Practice 13 Overview: Managing and Monitoring Your Application

This practice covers answering questions about:

- Managing your services
- Monitoring the activity of your application
- Monitoring and improving the performance of your application



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This practice addresses many topics related to managing and monitoring your application. You review the service and monitoring reports that you can execute within Oracle Application Express. In addition, you examine the activity report supplied with this course. You also evaluate some of the performance-monitoring capabilities.

Summary

In this lesson, you should have learned how to:

- Manage your services
- Monitor the activity of your application
- Monitor and improve the performance of your application



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you should have learned how to manage your services, monitor activities in your application and also how to improve the performance of your application.

Jack learnt several ways to monitor the PTS application and improve performance. He used bind variables within the application whenever possible. He included a #TIMING# substitution string in the region footer so that you can view the timing of each region. He also enabled caching for the PTS application pages and regions.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

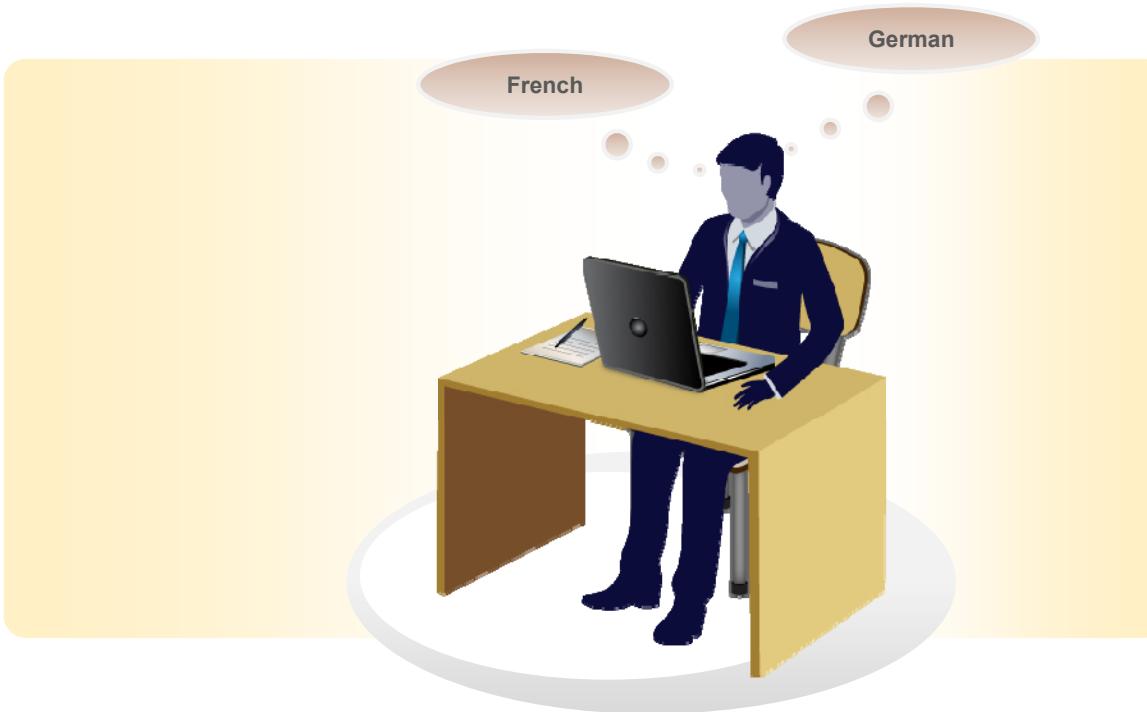
14

Globalization And Translation

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Translating the PTS Application



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Jack wants to have the PTS application language based on the user's browser language preference, application preference, or item preference. He explores APEX and tries to find out what are the languages available in APEX.

Objectives

After completing this lesson, you should be able to:

- Configure globalization attributes in an application
- Translate an application
- Specify a primary language for an application
- Translate messages used in PL/SQL procedures

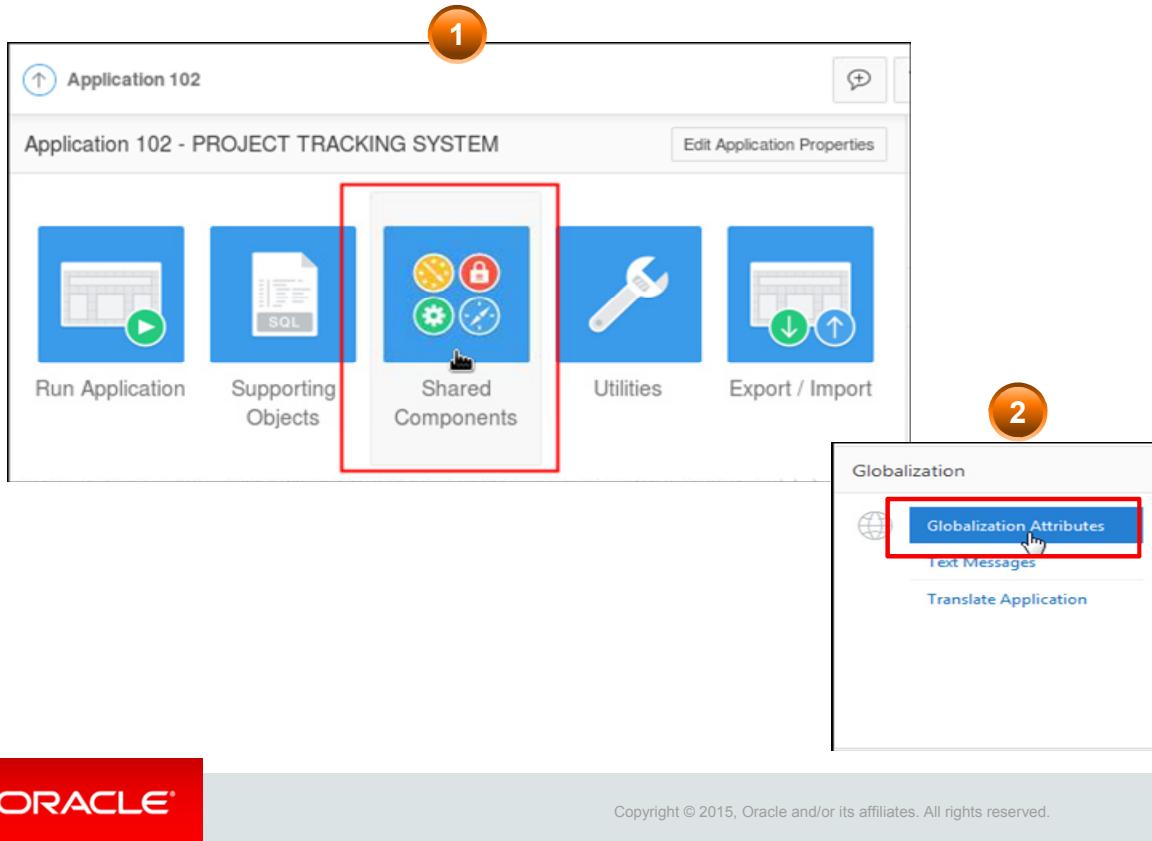


ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you understand how to set the globalization attributes for an application and also understand the various steps involved in translating an application. You learn how to specify a primary language for an application and translate messages used in PL/SQL procedures.

Accessing the Globalization Attributes Page



You can develop applications in Application Builder that can run concurrently in different languages. A single Oracle Database instance and Oracle Application Express supports multiple database sessions that can be customized to support different languages.

You can use the attributes on the Edit Globalization Attributes page to specify globalization options such as the primary application language. To access the Edit Globalization Attributes page:

1. Navigate to the application's home page and click **Shared Components**.
2. Under Globalization, click **Globalization Attributes**.

Editing the Globalization Attributes Page

The screenshot shows the 'Globalization' tab selected in the top navigation bar. The application name 'Application 102' is displayed above a form. The form contains the following fields:

- Application Primary Language: English (en)
- Application Language Derived From: Application Primary Language
- Application Date Format: DD-MON-YYYY
- Application Date Time Format: (empty)
- Application Timestamp Format: (empty)
- Application Timestamp Time Zone Format: (empty)
- Character Value Comparison: (empty)
- Character Value Comparison Behavior: Database session NLS setting (default)
- Automatic Time Zone: No
- Automatic CSV Encoding: Yes

At the bottom left is the Oracle logo, and at the bottom right is the copyright notice: Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can specify the following attributes on the Edit Globalization Attributes page:

Application Primary Language: This attribute identifies the language in which an application is developed. This language is the base language from which all translations are made. For example, suppose application 100 was authored in English, translated into French, and published as application 101, English would be the Application Primary Language.

Application Language Derived From: This attribute determines how Application Builder determines or derives the application language. The application primary language can be static, derived from the web browser language, or determined from a user preference or item. The database language setting also determines how the date is displayed and how certain information is sorted. This option enables you to disable browser-derived language support. You also have the option of having the application language derived from an application preference.

Application Date Format: This attribute determines the date format to be used in the application. Use this date format to alter the `NLS_DATE_FORMAT` database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle date format mask or an item reference using substitution syntax. If no value is specified, the default date format is derived from the database session at run time.

Application Date Time Format: This attribute is used to specify the date time format to be used in the application. This date time format can be referenced in an application using the substitution reference &APP_DATE_TIME_FORMAT., or in PL/SQL using the functionv('APP_DATE_TIME_FORMAT'). This attribute does not alter any NLS settings. This value can be a literal string containing a valid Oracle date format mask or an item reference using substitution syntax. If this attribute value is not specified, a reference to APP_DATE_TIME_FORMAT returns the NLS database session date format and the NLS time format.

Application Time Stamp Format: This attribute determines the time stamp format to be used in the application. Select a time stamp format from the list of values. Use this time stamp format to alter the NLS_TIMESTAMP_FORMAT database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle time stamp format mask or an item reference using substitution syntax. If no value is specified, the default time stamp format is derived from the database session at run time.

Application Time Stamp Time Zone Format: This attribute determines the time stamp with the time zone format to be used in the application. Use this date format to alter the NLS_TIMESTAMP_TZ_FORMAT database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle time stamp with time zone format mask or an item reference using substitution syntax. If no value is specified, the default time stamp with time zone format is derived from the database session at run time.

Character Value Comparison: This attribute determines the collating sequence for character value comparison in various SQL operations and clauses; for example, ORDER BY, LIKE, MIN/MAX. Use this value to alter NLS_SORT database session parameter for the execution of SQL queries in classic report and interactive report regions. If no value is specified, the default value is derived from the database session at run time.

Character Value Comparison Behavior: This attribute determines the collation behavior of SQL operations; for example, LIKE, MIN/MAX. Use this value to alter the NLS_COMP database session parameter for the execution of SQL queries in classic report and interactive report regions. The options include:

- **Database session NLS setting (default):** The NLS_COMP value is derived from the database session at run time.
- **Binary:** Comparisons in WHERE clauses and other SQL operations are binary.
- **Linguistic:** Comparisons in WHERE clauses and other SQL operations use the linguistic sort specified in the Character Value Comparison attribute (NLS_SORT).

Automatic Time Zone: This attribute controls the setting of the database session time zone. When set to Yes, the client time zone is derived from the client's web browser and set for the duration of the Application Express session. Subsequent page views have the database session time zone set properly per page view. When set, this setting can be overridden using APEX_UTIL.SET_SESSION_TIME_ZONE, or reset using APEX_UTIL.RESET_SESSION_TIME_ZONE.

Automatic CSV Encoding: This attribute controls the encoding of all comma-delimited (CSV) report output in an application. The default value for Automatic CSV Encoding is No. If Automatic CSV Encoding is set to Yes, the CSV report output is converted to a character set compatible with localized desktop applications. The character set for the CSV encoding is determined by the Application Language Derived From setting.

By default, the CSV output from report regions is encoded in the same character set as the database access descriptor. However, some desktop spreadsheet applications require that the data is encoded in the client desktop operating system character set. For multibyte data, the CSV output from report regions often appears corrupted when opened by a desktop spreadsheet application. This is because the CSV output from report regions is encoded differently than what is required by the desktop application. Enabling Automatic CSV Encoding resolves this issue.

For example, if the user's language preference for an application is de, the CSV data is encoded in Western European Windows 1252, regardless of the Database Access Descriptor character set setting. If the user's language preference is zh-cn, the CSV data is encoded in Chinese GBK.

Translating an Application and Globalization Support

To translate an application built in APEX:

- Map the primary and target application IDs
- Seed and export the text to a file for translation
- Translate the text in the file
- Apply the translated file
- Publish the translated file



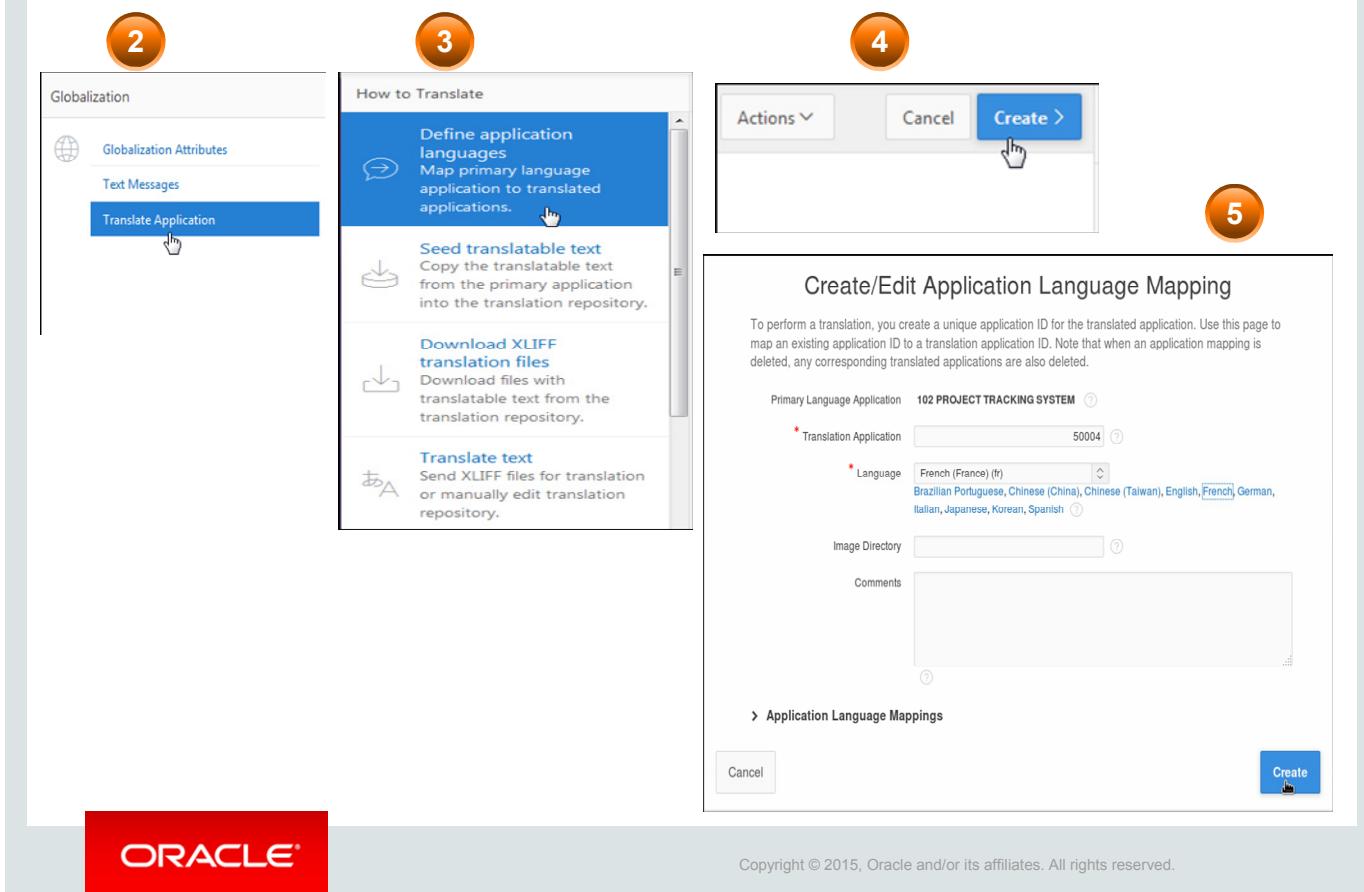
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In general, translating an application built in Application Builder involves the steps shown in the slide.

How Translated Applications are Rendered?

After Oracle Application Express determines the language for an application, the Application Express engine alters the database language for a specific page request. It then looks for a translated application in the appropriate language. If the Application Express engine finds that language, it renders the application using that definition. Otherwise, it renders the application in the base (or primary) application language.

Step 1: Mapping the Target Language



To translate an application, you need to map the primary and target application languages. The primary application is the application to be translated. The target application is the resulting translated application. To map the primary and target application languages:

1. Navigate to the application's home page and click **Shared Components**.
2. Under Globalization, click **Translate Application**.
3. Click Map primary language application to a translated application.
4. Click **Create**.
5. Select the attributes on the Translation Application Mapping page and click **Create**.

Jack wants to translate the PTS application language to French. He implements the steps one by one.

Step 2: Seeding and Downloading to a Translation File

How to Translate

- 3 Define application languages
- 4 Seed translatable text
- 5 Download XLIFF translation files

XLIFF Export

Translated Application	Language	Published	Strings	Distinct Strings	Requires Synchronization
<input checked="" type="checkbox"/> 50004	French (France) (fr)				Yes

Primary Language Application: 102

Seed

Export XLIFF File for Application

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

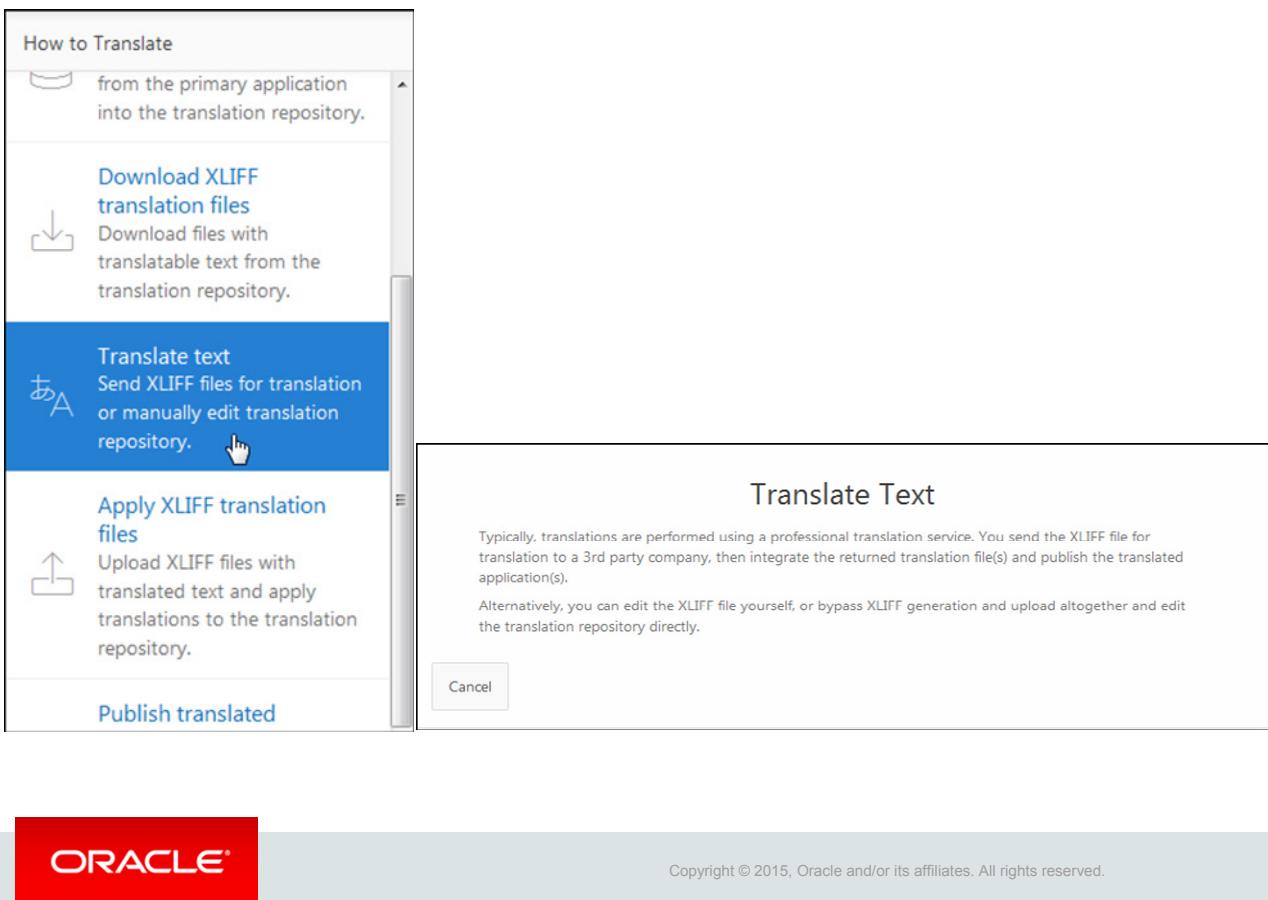
The second step is to seed the translation table and then export the translation text to a translation file.

Seeding Translatable Text: Seeding the translation copies all translatable text into the Translation Text repository. After you specify the language and seed the Translation Text, you can then generate and export an XLIFF (XML Localization Interchange File Format) file for translation. The seeding process keeps your primary language application synchronized with the Translation Text repository. You should run the seed process any time your primary language application changes. To seed translatable text:

1. Navigate to the application's home page and click **Shared Components**.
2. Under Globalization, click the **Translate Application**.
3. Select **Seed translatable text** to translation repository.
4. Select the appropriate target application and click **Seed**.

Downloading an XLIFF File: When the translation mappings are established, the translatable text within the application is seeded into a translation repository. This repository can then be exported to an XML Localization Interchange File Format (XLIFF) file for translation. The XLIFF Export page is divided into two sections. Use the upper section to export translatable text for an entire application (that is, all pages, lists of values, messages, and so on). Use the lower section to export translatable text for a specific page.

Step 3: Translating the XLIFF File



After exporting a translatable file to XLIFF format, you can translate it into the appropriate languages. Most vendors support XLIFF because it is an open standard XML file for exchanging translations. Oracle Application Express only supports XLIFF files encoded in UTF-8 character sets.

Translation is a time-consuming task. Oracle Application Express performs incremental translation so that application development can be done simultaneously with the translation. An XLIFF file can be translated and uploaded to Oracle Application Express even when only part of the XLIFF file is translated. For strings that have no translation in the corresponding translated application, Oracle Application Express uses the corresponding ones in the primary language.

Step 4: Applying XLIFF Translation Files

The screenshot shows three panels illustrating the process:

- Panel 3:** A sidebar menu with the following items:
 - Copy the translatable text from the primary application into the translation repository.
 - Download XLIFF translation files
 - Translate text
 - Apply XLIFF translation files** (highlighted with a blue background)
 - Publish translated applications
- Panel 4:** A modal dialog titled "Apply XLIFF translation files" with buttons for "Cancel", "Publish >", and "Upload Files >". The "Upload Files >" button is highlighted with a hand cursor icon.
- Panel 5:** The "XLIFF Upload" page. It displays a list of 8 file upload slots, each with a "Browse..." button. The first slot has a file selected: "f102_50004_en_fr.xlf". Below the slots is a table with columns: "Filename", "Document Size", "Created", "Created By", and "Apply to Translation". The first row in the table corresponds to the selected file. A red box highlights the "Apply Checked" button in the table header. At the bottom right of the page is a large blue "Upload" button.

After your XLIFF document has been translated, the next step is to upload and then apply it.

To upload a translated XLIFF document:

1. Navigate to the application's home page and click **Shared Components**.
2. Under Globalization, click **Translate Application**.
3. Click **Apply XLIFF translation files**.
4. Click **Upload Files**.
5. On the XLIFF Upload page, enter a description, locate the file to be uploaded, and click **Upload**. The uploaded document appears on the XLIFF Translation Files page.

After uploading an XLIFF document, the next step is to apply the XLIFF document and then publish the translated application. When you apply an XLIFF document, the Application Express engine parses the file and then updates the translation tables with the new translatable text.

To apply the XLIFF files, select the XLIFF files you want to apply and choose a translation mapping. Click **Apply Checked**.

Step 5: Publishing the Application

The screenshot shows two panels. The left panel, labeled '3', is a 'How to Translate' guide with five steps: 'Seed translatable text', 'Download XLIFF translation files', 'Translate text', 'Apply XLIFF translation files', and 'Publish translated applications'. The 'Publish translated applications' step is highlighted with a blue background and a cursor icon. The right panel, labeled '4', is a 'Primary Language Application' table with one row. The table columns are: Translated Application (checkbox), Language (dropdown), Published (checkbox), Strings, Distinct Strings, and Requires Synchronization. The row shows: 50004, French (France) (fr), Published checked, 13,177, 1,038, Yes. A 'Publish' button is visible at the top of this panel.

Translated Application	Language	Published	Strings	Distinct Strings	Requires Synchronization
<input checked="" type="checkbox"/>	50004 French (France) (fr)	<input checked="" type="checkbox"/>	13,177	1,038	Yes

Publishing your application creates a copy of the base language application, substituting the translated text strings from your translations table. This published application can then be used to render your application in alternative languages. To run an application in an alternative language, you must run it with globalization settings that cause an alternative language version to display. For example, if the language is derived from the browser language, you must set the browser language to the same language as the translated application.

To publish an application:

1. Navigate to the application's home page and click **Shared Components**.
2. Under Globalization, click the **Translate Application** page.
3. Click **Publish translated applications**.
4. Select the items you want to publish and click **Publish**.

Specifying the Primary Language for an Application

The screenshot shows the 'Globalization' configuration page for 'Application 102'. On the left, a sidebar lists 'Globalization' options: 'Globalization Attributes' (selected), 'Text Messages', and 'Translate Application'. The main panel displays various globalization settings:

- Application Primary Language:** English (en) (dropdown)
- Application Language Derived From:** Application Primary Language (dropdown)
- Application Date Format:** DD-MON-YYYY (input field)
- Application Date Time Format:** (input field)
- Application Timestamp Format:** (input field)
- Application Timestamp Time Zone Format:** (input field)
- Character Value Comparison:** (input field)
- Character Value Comparison Behavior:** Database session NLS setting (default) (dropdown)
- Automatic Time Zone:** No (dropdown)

At the bottom right of the main panel, there are 'Cancel' and 'Apply Changes' buttons.

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can use Globalization attributes to specify how the Application Express engine determines the primary language of an application. To edit globalization attributes:

1. Navigate to the application's home page and click **Shared Components**.
2. Under Globalization, select **Globalization Attributes**.
3. From Application Primary Language, select the language in which the application is being developed.
4. From Application Language Derived From, specify how the Application Express engine determines (or derives) the application language.
5. Configure other options as appropriate.
6. Click **Apply Changes**.

Translating Messages Used in PL/SQL Procedures

Syntax for APEX_LANG.MESSAGE

```

BEGIN
  --
  -- Print the greeting
  --
  APEX_LANG.MESSAGE ('GOOD_MORNING',
    V('APP_USER'));
END;

```

ORACLE

If your application includes PL/SQL regions, PL/SQL processes, or calls PL/SQL package, procedures, or functions, you may need to translate generated HTML. First, you define each message on the Translatable Messages page. Second, you use the APEX_LANG.MESSAGE API to translate the messages from PL/SQL stored procedures, triggers, or packaged procedures and functions. You create translatable messages on the Translate Messages page. To define a new translation message:

1. Navigate to the application's home page and click **Shared Components**.
2. Under Globalization, click **Text Messages**.
3. On the Translate Messages page, click **Create Text Message**.
4. On Identify Text Message, enter the attributes and click **Create Text Message**.

Use the APEX_LANG.MESSAGE API to translate text strings (or messages) generated from PL/SQL stored procedures, triggers, packaged procedures, and functions.

Practice 14 Overview: Applying Globalization and Translation

This practice covers the following:

- Configuring globalization attributes in an application
- Translating an application
- Specifying a primary language for an application
- Translating messages used in PL/SQL procedures



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Translate an application
- Specify the primary language of an application
- Translate messages used in PL/SQL procedures



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you should have learned the various steps involved in translating an application. You also learned how to specify the primary language of an application and translate messages used in PL/SQL procedures.

Jack translates and specifies the primary language of an application, as well as translates messages used in PL/SQL procedures.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only



Quick Reference: Additional How-To Guide

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

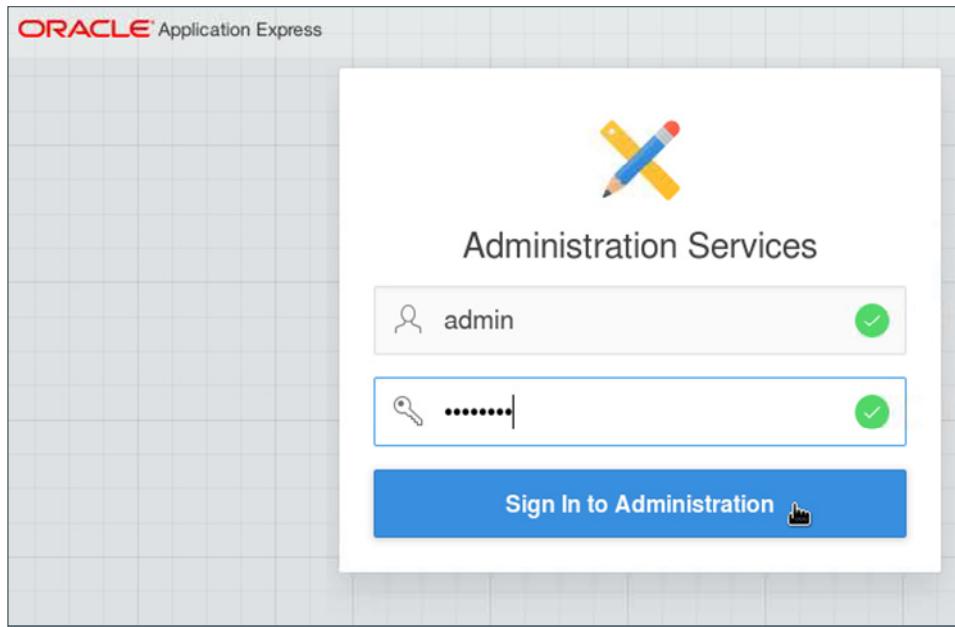
After completing this lesson, you should be able to:

- Create a workspace and a workspace administrator
- Create database objects
- Run SQL commands and SQL scripts
- Create a database application
- Create a report
- Create a form on a table with report
- Create a region
- Create and edit page items and buttons
- Create a branch
- Create a navigational menu entry
- Create lists and list entries



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Logging In to Oracle Application Express Administration



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Before any development can occur, a workspace and workspace administrator need to be created. This is done using Oracle Application Express Administration. You can log in to Oracle Application Express Administration by using the following URL:

http://<hostname>:<port>/apex/apex_admin

The login window appears. Enter the default username `admin` and the password, and then click Sign In to Administration.

Creating a Workspace and Workspace Administrator

The screenshot shows a dialog box titled 'Create Workspace'. It contains fields for workspace information (Name: demo, Security Group ID: System Assigned, Description: ...), administrator information (User Name: ADMIN, E-mail: demo@oracle.com), and schema information (Reuse Existing Schema: No, Schema Name: DEMO, Tablespace will be created: APEX_XXX, Datafile for tablespace: /u01/app/oracle/oradata/orcl/APEX_XXX.DBF). There are 'Cancel' and 'Create Workspace' buttons at the top right.

You have requested to provision a new Workspace.

Workspace Information:

Name **demo**

Security Group ID **System Assigned**

Description **...**

Administrator Information:

User Name **ADMIN**

E-mail **demo@oracle.com**

Schema Information:

Reuse Existing Schema **No**

Schema Name **DEMO**

Tablespace will be created **APEX_XXX**

Datafile for tablespace **/u01/app/oracle/oradata/orcl/APEX_XXX.DBF**

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Application Express administrators create workspaces. When you create a workspace, you assign a name to the workspace, workspace administrator, and database schema. You can create a new schema or use an existing schema.

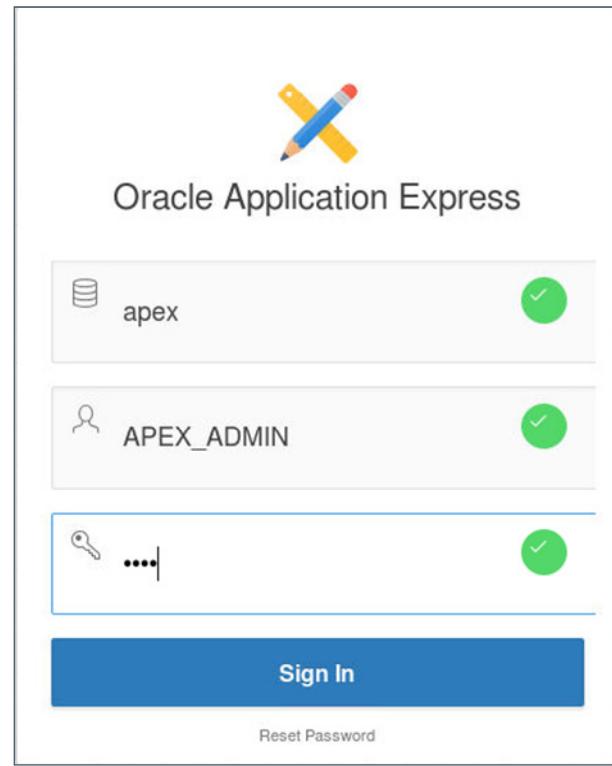
To create a workspace:

1. Click the Manage Workspaces icon.
2. Click the Create Workspace link.
3. Enter a workspace name and click Next >.
4. You can use an existing schema or enter a new schema that can be created. Enter the schema password, select the size of the schema, and click Next >.
5. Enter an administrator username and password that are created for this workspace. Enter your email address and click Next >.
6. Confirm your entries and click Create Workspace.

Logging In to a Workspace

To log in to an Oracle APEX workspace:

- Enter the correct URL in your browser address bar.
- Enter the workspace name.
- Enter the username and password.
- Click Sign In.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To log in to Oracle Application Express, you need a workspace name and the username and password created for that workspace. You can log in to Oracle Application Express as a workspace administrator or as a developer. You can access the Oracle Application Express application with the following URL:

`http://<hostname>:<port>/apex`

The login page appears. Enter the workspace name, username, and password. Click Sign In. You may be prompted to change your workspace password the first time you log in. This option is set when your username and password are created by the Oracle Application Express administrator. You can set your new password to be the same as your old password.

Note:

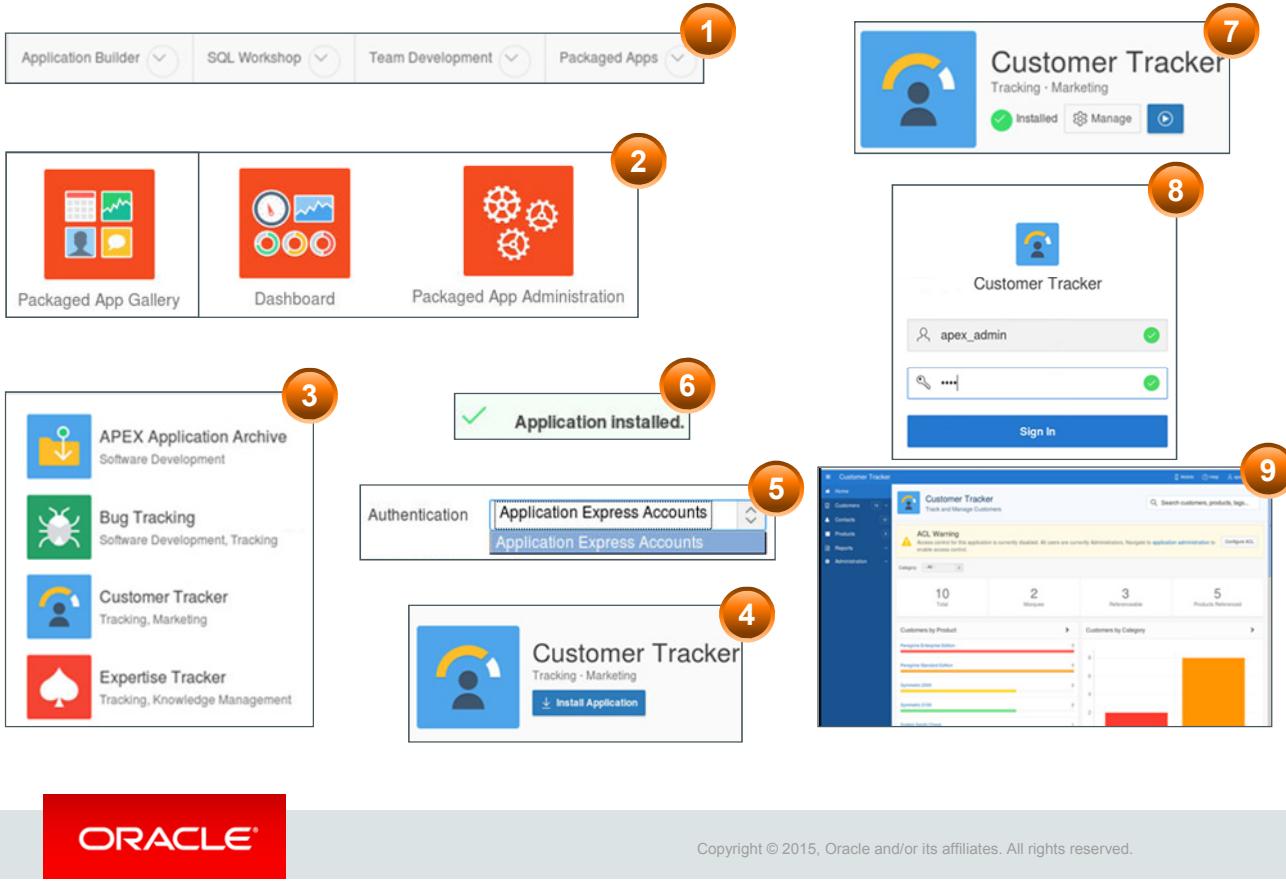
If your setup uses Oracle HTTP Server with `mod_plsql`, use:

`http://<hostname>:<port>/pls/apex`

If your setup uses embedded PL/SQL gateway or APEX Listener, use:

`http://<hostname>:<port>/apex`

Installing a Packaged Application



ORACLE®

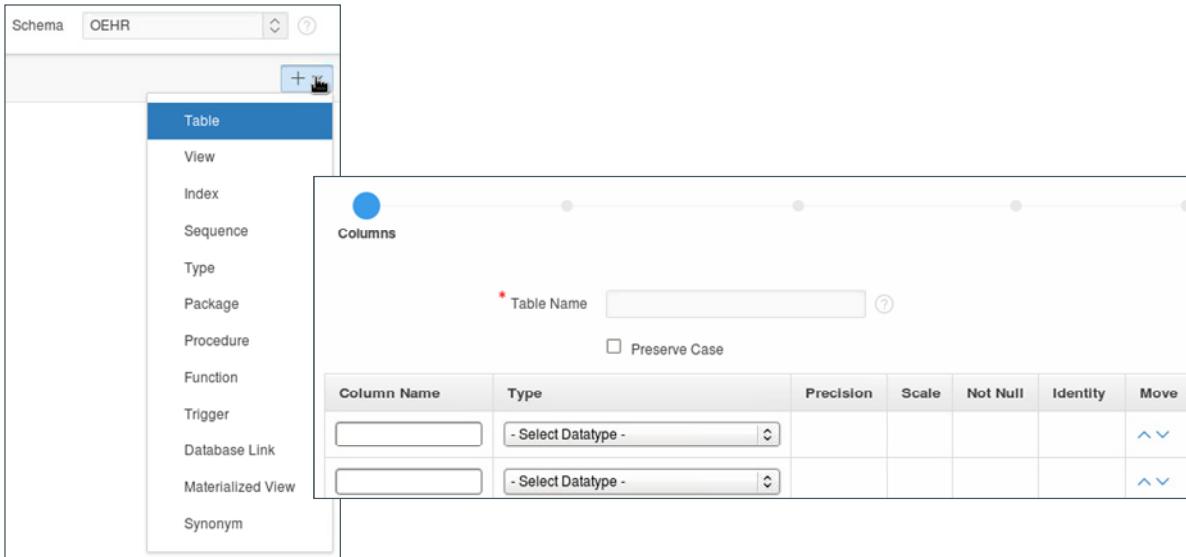
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To install a packaged application:

1. On the Application Builder home page, click the Packaged Apps tab. The Packaged Apps page appears.
2. Select Packaged App Gallery.
3. Locate the application to be installed. In the example in the slide, the Customer Tracker application is selected.
4. Click the application image. A summary page appears. Click Install Application.
5. Select an Authentication scheme and click Next.
6. Click Install Application again. A success message appears.
7. To run the application, click the Run Application icon.
8. Enter the appropriate login credentials and click Login.
9. The application is installed.

Creating Database Objects

SQL Workshop > Object Browser > Create

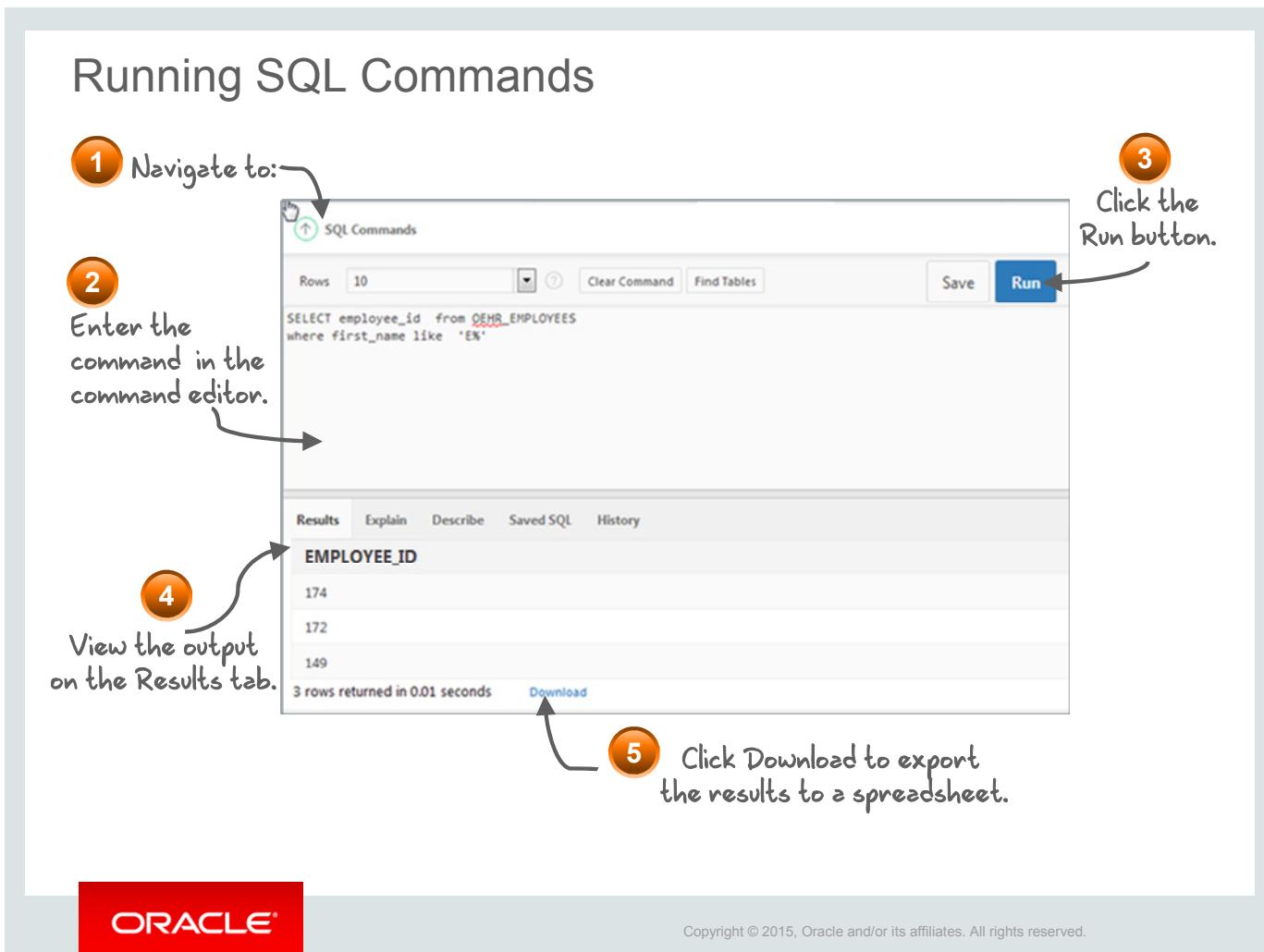


ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To access Object Browser, click the Object Browser icon on the SQL Workshop page. You can also select Object Browser from the drop-down menu on the SQL Workshop tab.

To create a database object, click the Create button on the Object Browser page. A list of objects you can create is displayed. The Create Database Objects wizard has three main steps: selecting the object you want to create, defining the object, and confirming the details. The steps for defining an object differ based on the object you are creating. For example, if you want to create a database table, select Table from the Create Database Object list (shown in the slide). The Create Table wizard opens. Defining the table object involves adding the columns, primary key, foreign key, and constraints. Finally, you confirm the details you entered and create the table.



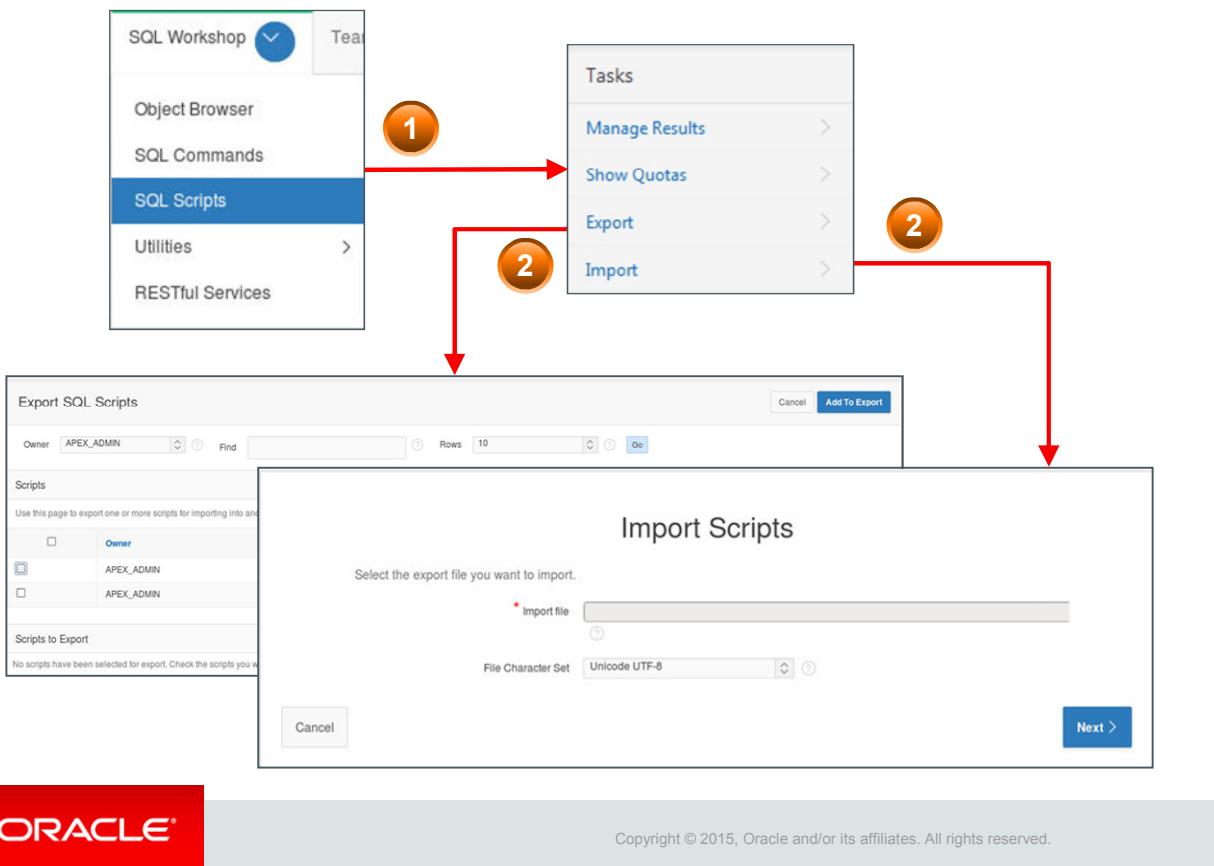
To execute SQL code with SQL Commands:

1. Navigate to the SQL Commands page by selecting **SQL Commands** from the drop-down menu on the SQL Workshop tab.
2. Enter the SQL or PL/SQL statement in the command editor.
3. Click the **Run** button.
4. View the output on the Results tab of the display pane.
5. (Optional) Click the **Download** link to export the results of the query to a spreadsheet in Microsoft Excel.

Note:

- If you have multiple commands in the command editor, you can run only one command at a time. Select the command and click Run. Only the command that was selected is executed.
- SQL commands that are created and saved by using Query Builder can be executed from the SQL Commands page.

Importing and Running a SQL Script



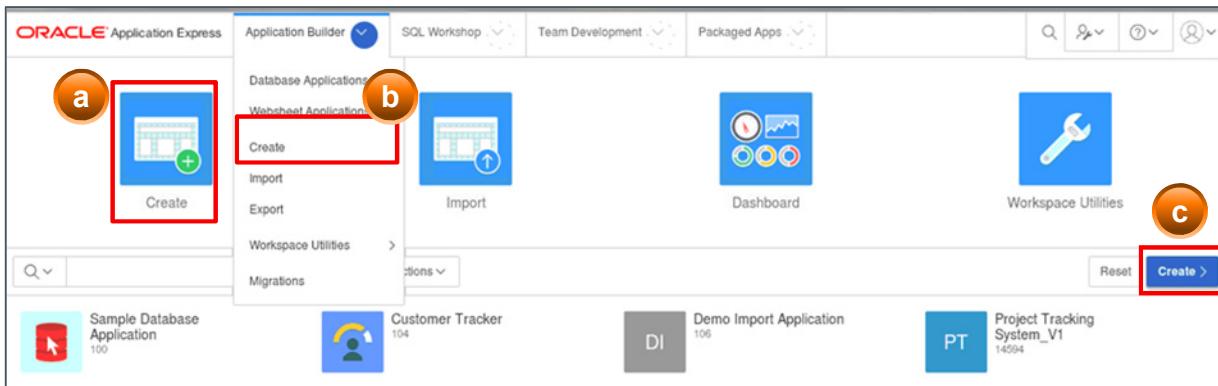
Using the Export and Import tasks, you can transfer scripts between workspaces.

Using Export, you can export multiple scripts from your current workspace to another workspace. All the scripts that you select to export are encoded into a single script file. You can save this file to your local file system and import it to another workspace.

To export scripts, click the Export link. The scripts available in the script repository are listed in the Scripts pane. Select the scripts that you want to export and click the Add To Export button. The selected scripts are listed in the “Scripts to Export” pane. You can finalize the scripts that you want to export by removing or adding scripts. To export all the scripts, click the Export All button. The scripts are exported as a single export file, which you can save to your local file system.

Using Import, you can import a script file exported from a different workspace into your current workspace. To import a script file, click the Import link. Click the Browse button and locate the file to import from your local file system. Click Next and Import Scripts to confirm. Only script files exported from the scripts repository can be imported. If you try to import any other script, you get a “script not compatible” error.

Accessing the Create Application Wizard

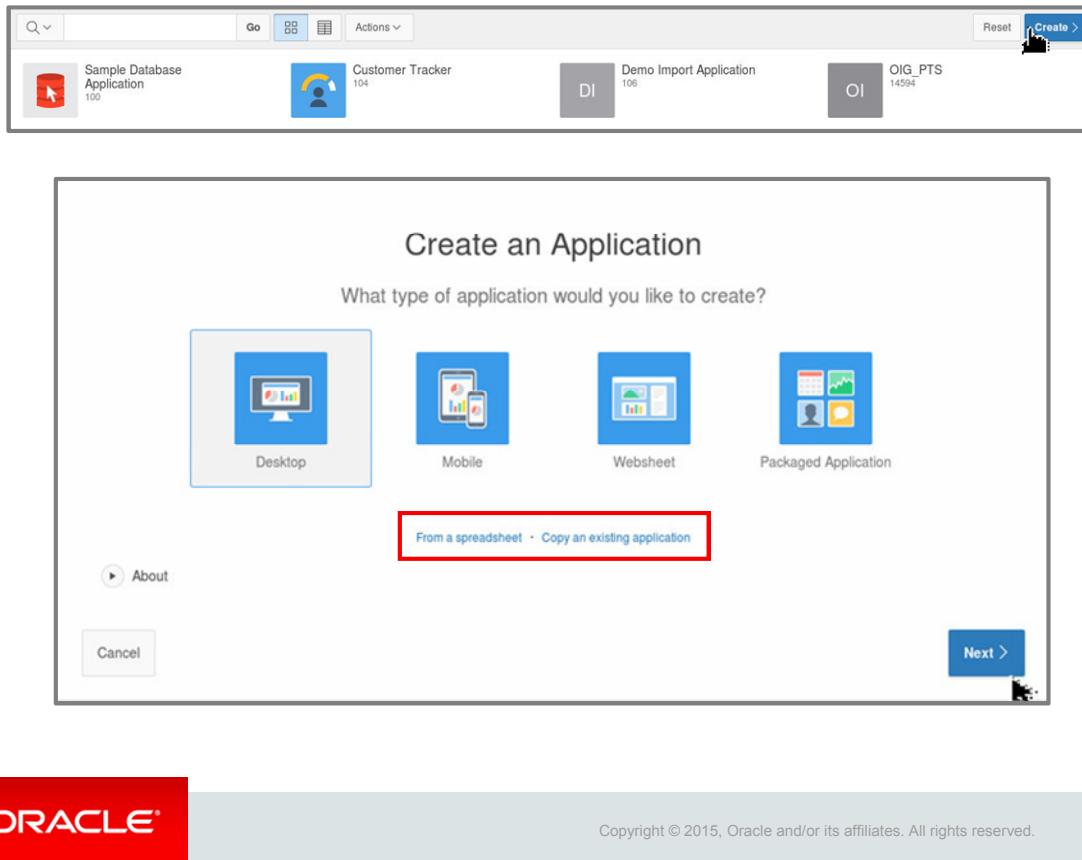


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To access the Create Application wizard, perform either of the following steps:

- Navigate to the Application Builder home page and click the Create icon.
- Select Create from the Application Builder menu.
- Navigate to the Application Builder home page and click the Create button.

Creating a Database Application



To create a database application, select Database for the application type and click Next. You have four options to create a database application:

- **From spreadsheet:** You can create an application based on spreadsheet data. You first upload or paste the spreadsheet data to create a table. Then, you select a default appearance. The resulting application enables end users to query, insert or update records, or analyze the data.
- **Copy of existing application:** You can create a copy of an existing application by running the Create Application wizard, or by selecting the application and then copying the application on the Application home page.
- **Install sample applications:** Oracle Application Express includes several sample applications. You can install, run, and use sample applications as they are, or analyze them to better understand how to use Application Builder to build specific types of functionality.

You can also create a database application based on a table, query, or drilldown query.

Creating a Desktop Database Application

In the Create Application wizard, after selecting Desktop, perform the following steps:

1. Specify an application name.
2. Select the type of page you want to add.
3. Specify whether you want to copy shared components from another application.
4. Specify the authentication scheme and date format.
5. Select a theme.
6. Confirm your selections to create the application.



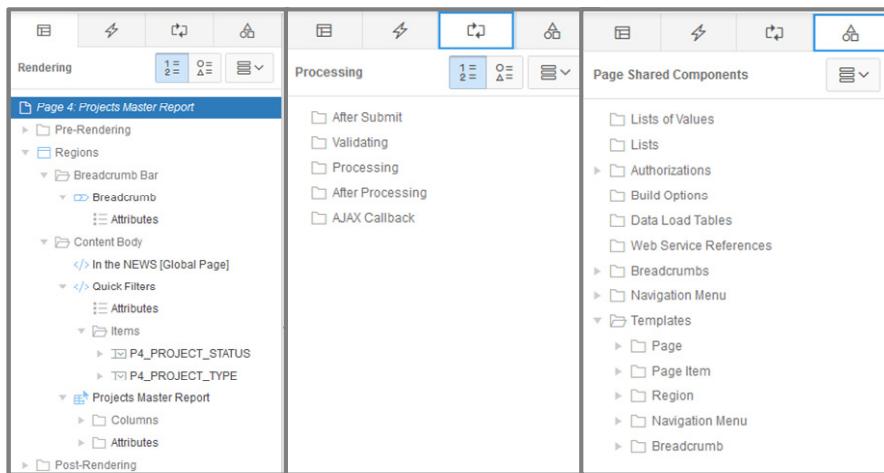
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When running the Create Application wizard, you must select a target user interface based on which the wizard is optimized to display the appropriate page types, attributes, and themes. The slide provides an overview of the steps to create a database application based on a table, query, or drilldown query.

Page Definition: Overview

- A page is the basic building block of an application.
- Page Definition is divided into three sections:
 - Page Rendering
 - Page Processing
 - Shared Components



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You build an application by using pages. Page Definition is divided into:

- **Page Rendering:** The process of generating a page from the database. You can use the Page Rendering section to modify controls that impact the rendering of a page, including the page definition, regions, buttons, items, page-rendering computations, and page processes.
- **Page Processing:** The process of submitting a page. A page is typically submitted when a user clicks a button. You can use the Page Processing section to specify application logic, such as computations, validations, processes, and branches. In general, the Application Express engine runs the logic of specific applications in the order in which they appear on Page Definition.
- **Shared Components:** List of the common components that can be displayed or applied on every page within an application. Some of the shared components include tabs, lists of values, breadcrumbs, lists, and navigation bars.

Different Views of a Page

Page Designer

Component View

ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are two ways to view a page: Page Designer and Component view.

- **Page Designer** organizes page components under four major categories:
 - **Tree View:** The components can be seen in tree view organized under Page Rendering, Dynamic Actions, Page Processing, and Shared Components.
 - **Grid Layout:** Gives a snapshot of how the page is going to look as per the design. You can drag an item, region, or button to the required position on the Grid Layout. There are also tabs to show error messages (if any), Page Search, and Help on an attribute selected in the property editor for any item.
 - **Gallery:** Contains all possible types of Regions, Items, and Buttons which can be designed into this page using Grid Layout.
 - **Property Editor:** Is located on the left side of the Page Designer and shows a complete list of all the properties of a component selected in the Page Components on the right side of the page.
- The **Component view** groups user interface elements and application logic by component type.

Switching Between Pages and View Types

Using the navigation bar, you can:

- Specify a specific page
- Undo or Redo a recent change
- Change view types
- Access links to create another Page, Region, Items and so on
- Access debugging tools
- Lock or unlock the page
- Save or ‘Save and Run’ the page



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are three ways to switch from one page to another:

1. Enter a page number in the Page field and click Go.
2. Click the up arrow next to the Page field and select a page from the list.
3. Click the Previous and Next buttons to the right of the Go button.

To switch from Designer view (which is the default) to Component view, click the Component View icon on the navigation bar. To switch to Designer view, click the Designer View icon on the navigation bar.

Creating a Report

To access the Create Report wizard by creating a new page:

1. Navigate to the Application home page and click Create Page.
2. From the “Select a page type” options, select Report.

To create a report by creating a new region on an existing page:

1. Select a report type region from the Regions Gallery.
2. Drag the selected report region into Content Body on the Grid Layout.
3. Update the report region properties in the Property Editor.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are two ways to create a report:

- By accessing the Create Report wizard while creating a new page in the application.
- By creating a report region on an existing page using Page Designer.

Editing Report Attributes

Report Region

Collapsible tabs under Region

Report Region Attributes in Property Editor

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ORACLE®

When you open a page in Page Designer mode and select the report region in 'Page Rendering', all the report attributes will be listed in the Property Editor on the right column of the page. The attributes will be organized into functional groups.

To edit any report attribute, locate it in the Property Editor and update its value as per the change required. The Messages tab will highlight errors in red color if any of the values updated in the properties pane are invalid.

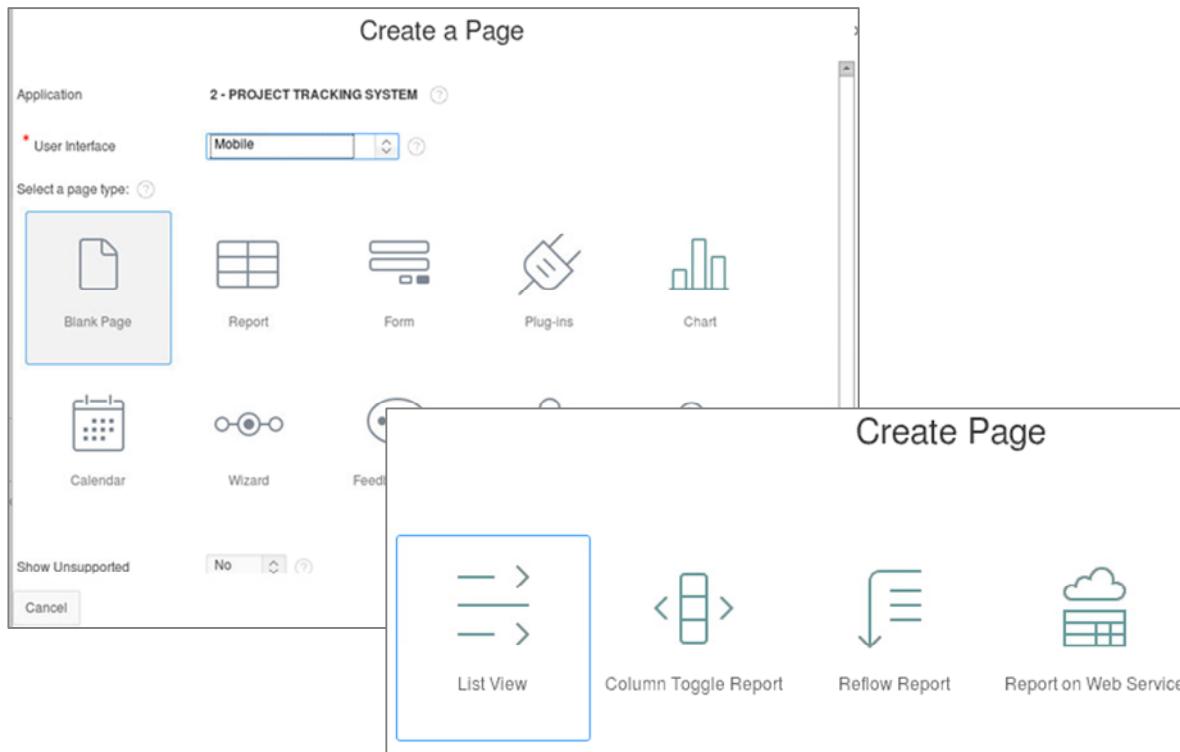
Update the report attributes as required and click the 'Save and Run Page' icon on the page to see the updated report.

All the report attributes are not associated only with the Report Region directly. Some of the attributes are organized under collapsible tabs shown under 'Report Region'.

All the columns selected from the database to show on the report are listed under the 'Columns' tab. By clicking each column, its properties appear in the Properties Pane on the right column. This allows you to update the properties at an individual column level. For example, by changing the 'Type' to 'Hidden', you can make any column not be listed even in the 'Do Not Display' box of the report.

The Attributes tab allows developers to deal with the properties related to Column Groups, Saved Reports, and Printing.

Types of Reports Supported for Mobile Interface



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When you create a report for a mobile application by using the Create Page wizard, you can select different report types. These report types are specific to the mobile interface.

When you create a new type of page report in a desktop application which has mobile interface added to it, you will be prompted to select the interface for which the new page is being created. If you select Mobile as the interface, the wizard shows four types of reports as seen in the slide.

- **List View:** Creates a report which displays the result of a SQL SELECT statement in the form of a list.
- **Column Toggle Report:** Creates a report based on a custom SQL SELECT statement with the priority set at column level. It allows end users to select or deselect columns so that you can control which columns are to be displayed.
- **Reflow Report:** Creates a responsive report which will automatically reflow itself into vertical view when the screen size is not sufficient to display the report in horizontal view.
- **Report on Web Service Result:** Creates a report based on a web service result.

Types of Forms



Form on a
Table or View



Tabular Form



Form on a
Table with Report



Master Detail
Form



Form on a
Procedure



Form on a
SQL Query



Form on
Web Service



Form and Report
on Web Service



Summary Page

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle Application Express provides wizards to create the types of forms listed in the slide.

- **Form on a Table or View wizard:** Creates a form to enable users to insert rows into a table.
- **Tabular Form wizard:** Creates a form to enable users to edit or delete multiple rows in a table simultaneously. Users will also be able to insert rows into the table.
- **Form on a Table with Report wizard:** Displays a report and enables users to edit or delete rows one at a time. Users will also be able to insert rows into the table.
- **Master Detail Form wizard:** Enables users to update data from two tables. You should have a foreign key relation between the two tables.

In this lesson, you learn how to create forms by using these four wizards.

Types of Forms

The following are the other wizards:

- **Form on a Procedure wizard:** Creates a form based on stored procedure arguments. Use this approach when you have implemented logic or data manipulation language (DML) in a stored procedure or package.
- **Form on a SQL Query wizard:** Creates a form based on the columns returned by a SQL SELECT Query.
- **Form on Web Service wizard:** Creates a page with items based on a web service definition. This wizard creates a user input form, a process to call the web service, and a Submit button.
- **Form and Report on Web Service wizard:** Creates a page with items based on a web service definition. This wizard creates a user input form, a process to call the web service, and a Submit button, and displays the results returned in a report.
- **Summary Page wizard:** Creates a read-only version of a form. A typical use case is to provide a confirmation page at the end of a wizard.

Creating a Form on a Table with Report



- Type of report
- Page number and name
- Region template and name
- Tabs
- Columns to display
- Image for edit link
- Page number and name
- Region template and name
- Primary key and trigger source
- Columns to edit
- Actions to enable (insert, update, and delete)

ORACLE®

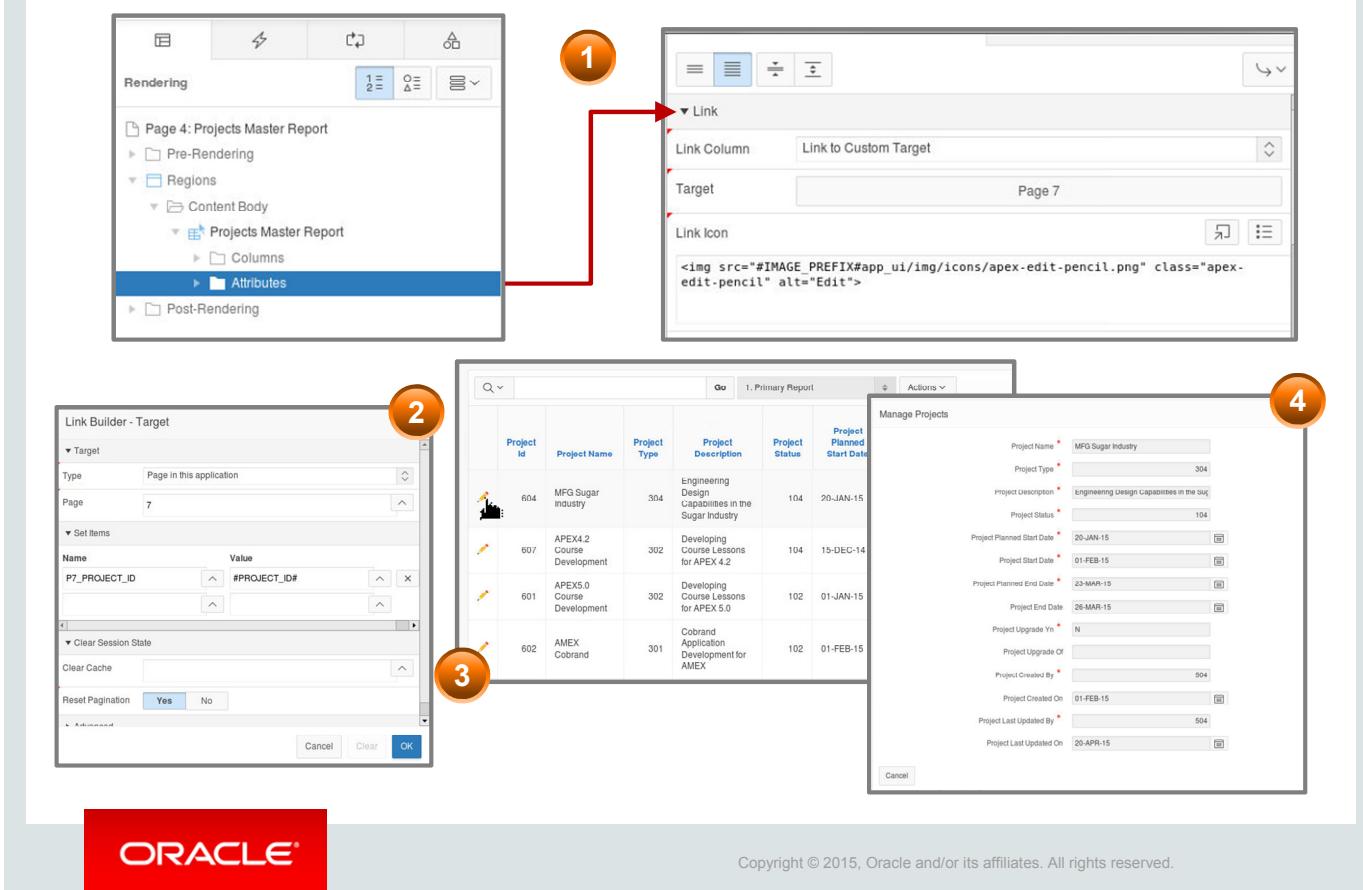
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To access the Create Form wizards while creating a new page in your application:

1. Click Create Page on the home page of the application where you want to create the form.
2. A Create Page wizard opens. Select Form from the available options and click Next.
3. The form wizards are displayed. You can select a wizard based on the type of form you want to create.

The Form on a Table with Report wizard combines the steps to create a report and the steps to create a form, and creates two pages. The first page is a report with an edit link for each row. The report page also includes a Create button to allow users to insert rows into the table. The second page is a form to edit/delete the row selected from the first page (reports page). The slide lists the steps to define the report and the form pages.

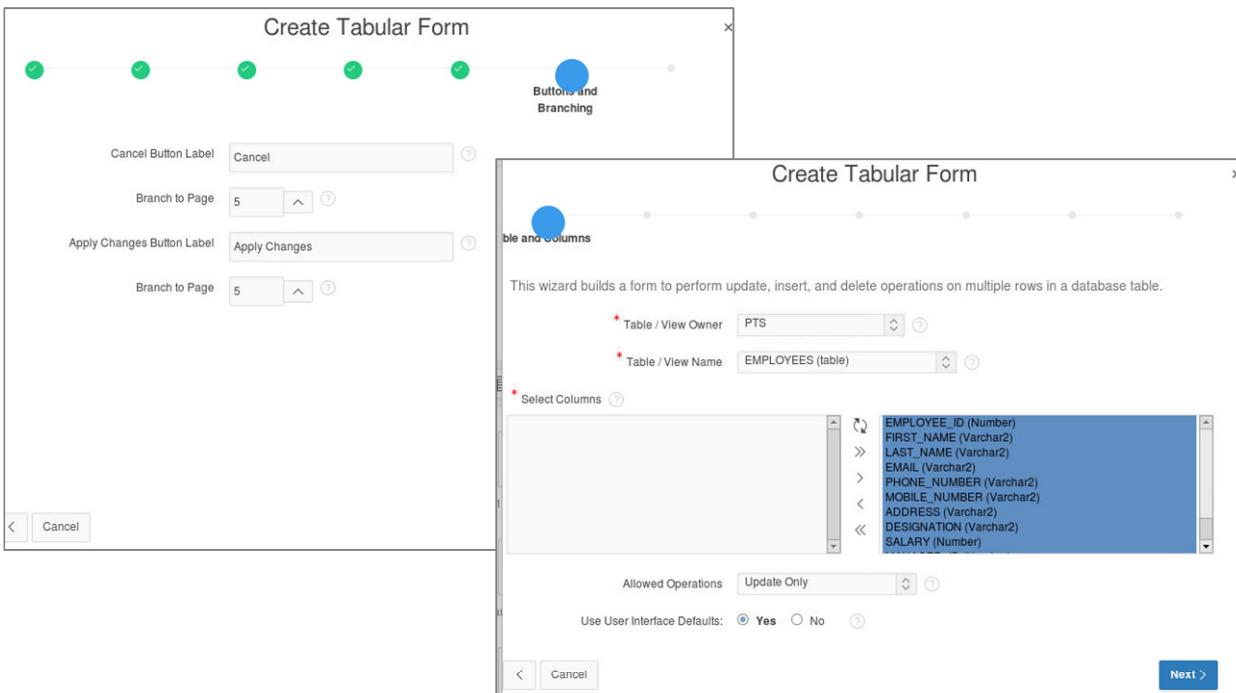
Linking a Report to a Form



When you create a Form on a Table with Report, the wizard automatically creates the required report, form, and links between them. Stand-alone reports (interactive or classic) can also be linked to existing forms. This slide shows how you can link a classic report to a form.

- From the Page Definition of the page where you have created the report, click **Attributes** under Rendering in the left pane.
- Locate **Link** under Properties in the right pane and select **Link to Custom Target** for Link Column.
- Click **Target** and a pop-up window opens. Click the up arrow beside Page and select the target page from All Pages. In this example, the Manage Projects form is selected as the target page.

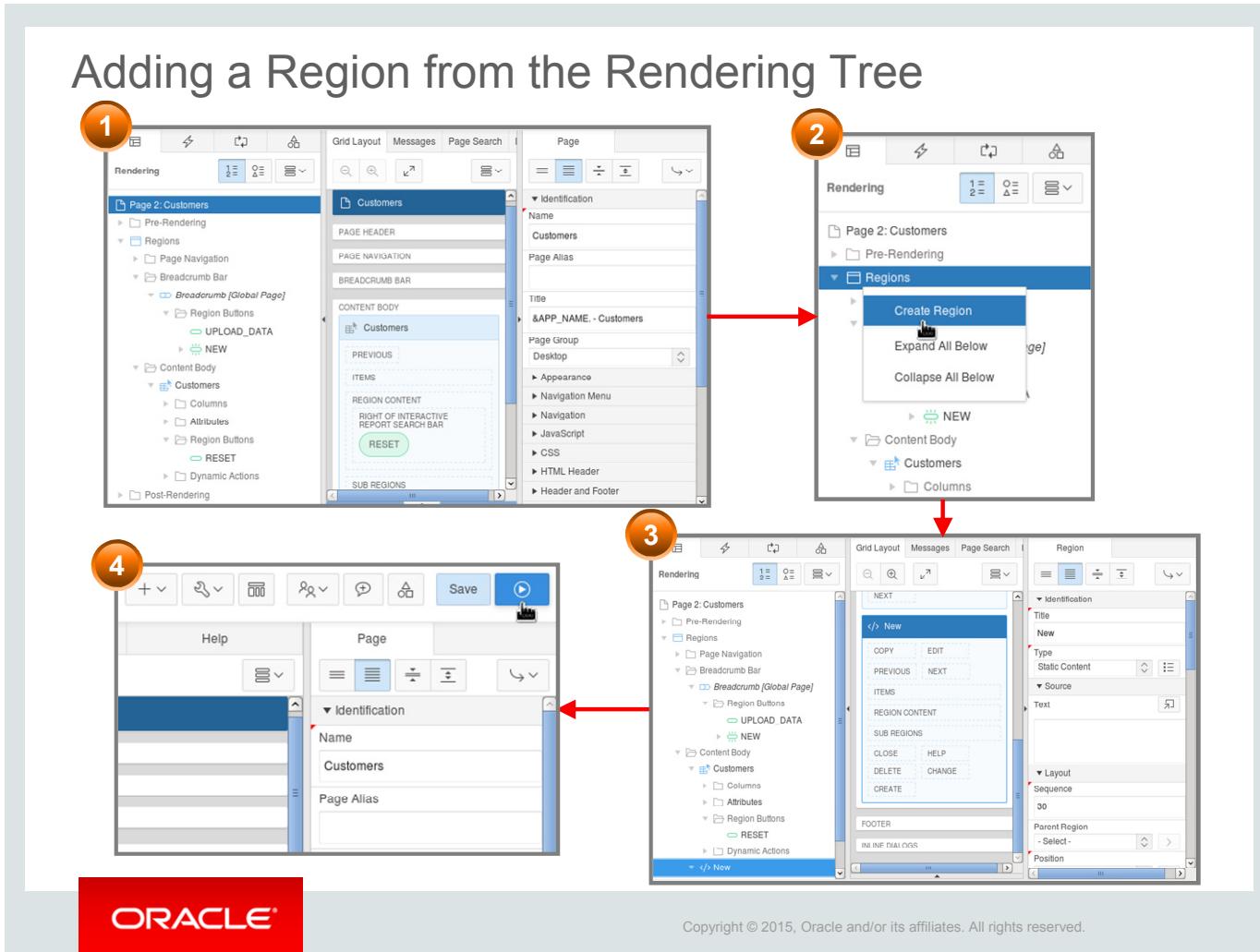
Creating a Tabular Form



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

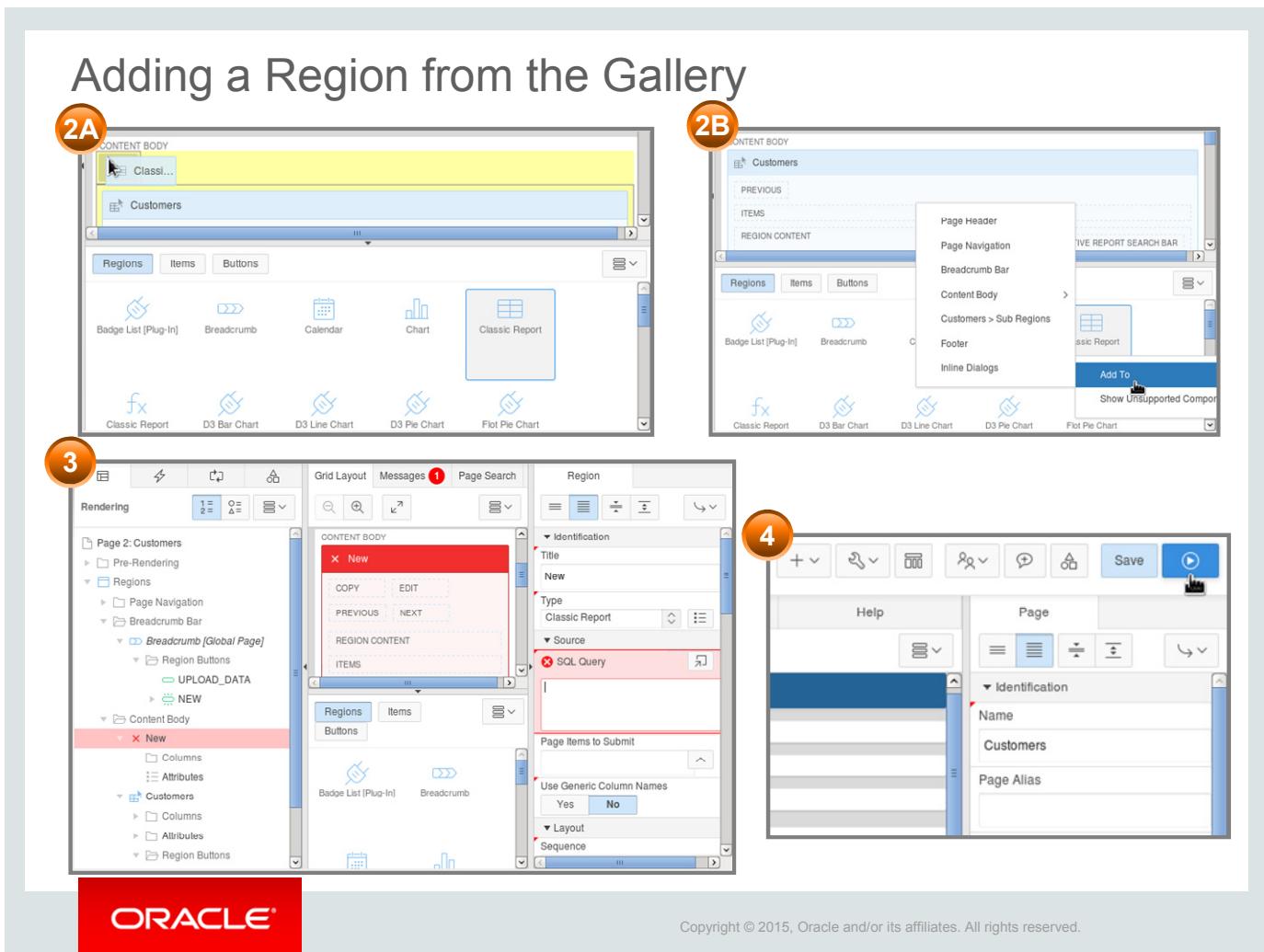
The Tabular Form wizard has steps that are similar to the Form on a Table or View wizard. While selecting the schema, the wizard prompts you to set the operations that you want to allow users to perform on the form. By default, the “Update, Insert, and Delete” option is selected. Using this wizard, you can select the columns to be displayed in the form and the columns that can be updated. Depending on the operations that you allowed the users to perform, the buttons are included on the form page. You can change the label name for the buttons.



Each page can have buttons and fields (called items) which are grouped into containers called regions. You can add regions to a page from the Rendering Tree or from the Gallery.

To add a region from the Rendering Tree:

1. View the page in Page Designer.
2. In the left pane, right-click **Regions** in the Rendering tab and select **Create Region**.
3. The Region is created. Edit the appropriate attributes in the Property Editor. Required attributes display a red triangle in the upper left corner above the attribute label.
4. Click **Save** or **Save and Run Page**.



To add a region from the Gallery:

1. View the page in Page Designer.
2. In the Gallery, locate the region you want to add. You can then perform either of the following options:
 - a. Click and hold the mouse on the component to be created, and drag it to the desired location in Grid Layout.
 - b. Right-click a component to view a context menu. For **Add To**, select the location where you want the region to be added.
3. The Region is created. Edit the appropriate attributes in the Property Editor. Required attributes display a red triangle in the upper left corner above the attribute label.
4. Click **Save** or **Save and Run Page**.

Page Items: Examples

The screenshot shows two examples of Oracle Application Express page items:

- Product Details:** A form for adding a product. It includes fields for Product Name (Text Field), Product Description (Text Area), Category (Select List), Product Available (Radio Group), List Price (Number Field), Product Image (File Browse), Tags (Text Field), and an Add Product button.
- Create Order:** A form for creating an order. It includes fields for Create Order for (radio group with Existing customer selected) and Customer (Pop-up LOV).

Annotations with arrows point to specific elements:

- Select List:** Points to the Category dropdown.
- Radio Group:** Points to the Product Available radio buttons.
- Text Field:** Points to the Product Name field.
- Text Area:** Points to the Product Description field.
- File Browse:** Points to the Product Image browse button.
- Pop-up LOV:** Points to the Customer dropdown.

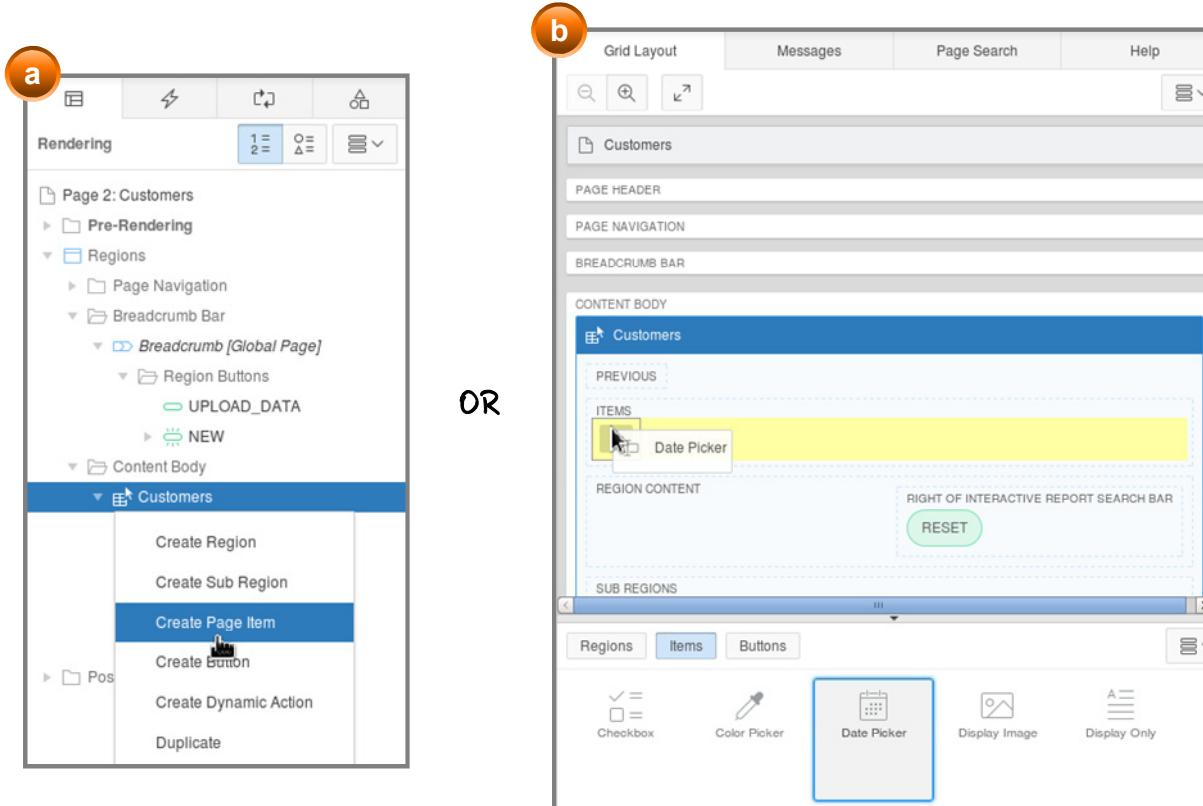
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The slide displays some page item examples. An item is part of an HTML form and can be used to store a value in session state so that it can be retrieved at a later time. The examples shown in the slide are page items. Page items are placed on a page and have associated user interface properties, such as Display As, Label, and Label Templates. Another type of item, application item, is discussed in the next slide.

When you create a form by using a wizard, an item is created for each column of the table. The default item type is a text field, text area, number field, date picker, or File Browse, depending on whether the database table column type is varchar, varchar2 (with size greater than 255 characters), numeric, date, or BLOB, respectively. You can edit the item properties to change the display type. For example, you can change a text field to a text area or select list.

Creating a Page Item



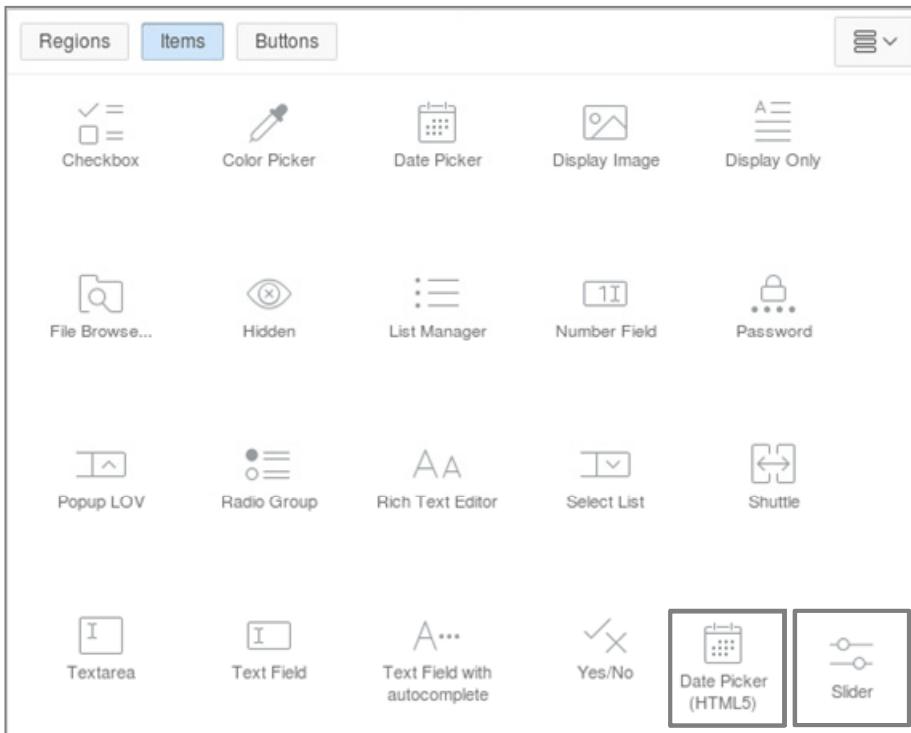
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can create a Page Item in either of the following ways:

- In the Rendering pane of the Page Designer, right-click the region node where you want to create the item and select Create Page Item.
- Select the item that you want to create from the Items gallery. Drag it from the gallery to the Grid Layout of the page under the region where you want to create the item.

After either of these steps, set/edit the property of the item by using the Property Editor (right pane of the Page Designer).

Types of Page Items



ORACLE®

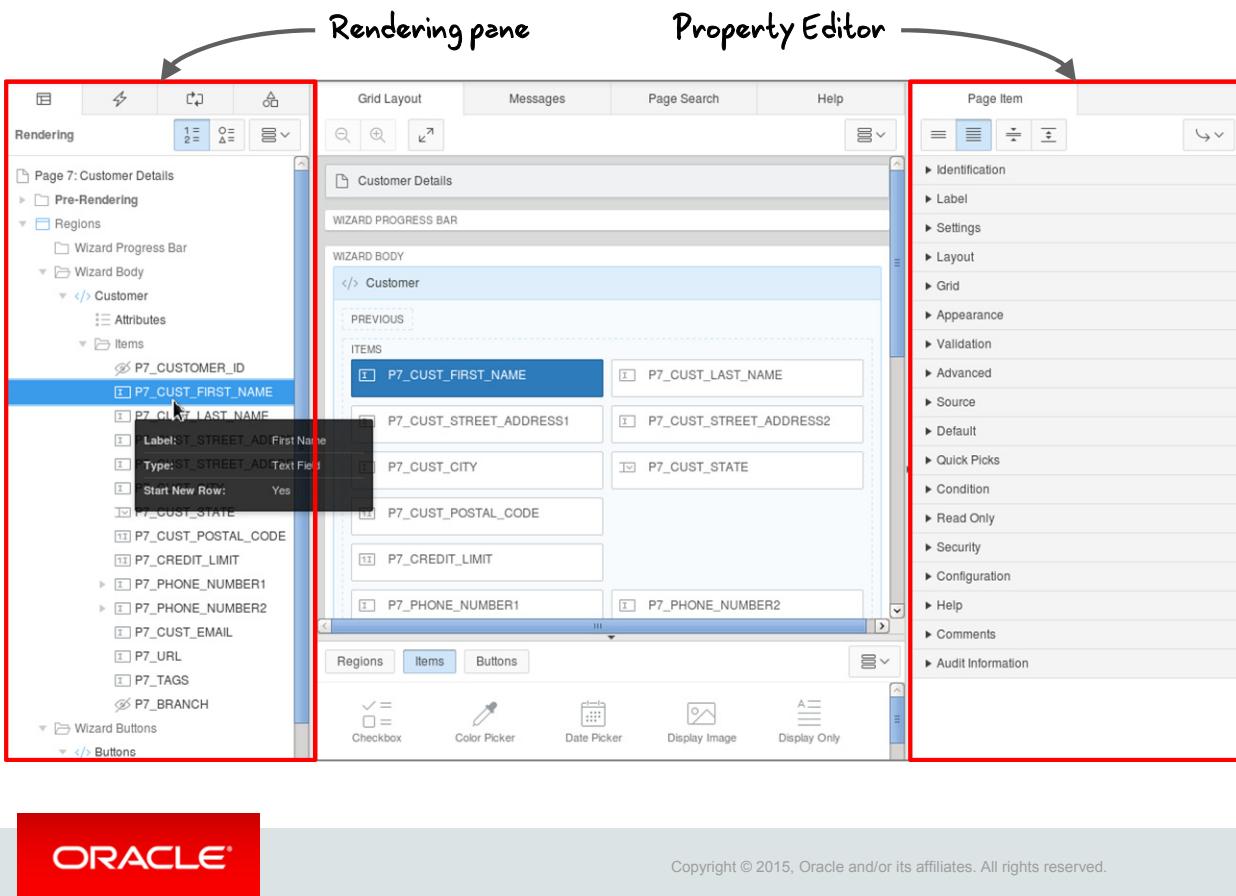
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

- **Checkbox:** Is based on a list of values. The value corresponding to a check box is returned in a string delimited by a single colon (:).
- **Color Picker:** Renders as a text field with an icon. When the user clicks the icon, a pop-up window appears. When the user makes a selection from the palette, the HTML value for the color selected (for example, #0000000 for black) is returned.
- **Date Picker:** Displays a text field with a calendar icon next to the text field. You can specify a format mask, maximum and minimum date, year range, and so on while creating the item.
- **Display Only, Display Image:** The Display Only item displays a read-only version of a display value. The Display Image item displays a specified image.
- **File Browse:** Displays a text field with a Browse button. Using this, you can locate a file in a local file system and upload it. The files that you upload are stored in a table called `wwv_flow_file_objects$`. Every workspace has access to this table through a view called `APEX_APPLICATION_FILES`.
- **Hidden:** Creates an HTML hidden form element. You can use this item to store session state values.
- **List Manager:** Is based on a list of values. Using this, you can manage a list of items by selecting from and adding to a list.

- **Number Field:** Validates the user input and accepts only numerical data.
- **Password:** Creates a text field that displays an asterisk for each character entered.
- **Popup LOV:** Renders a text field with an icon next to it. A user can click it and select a value from the pop-up window. The list in the pop-up window is driven by a list of values.
- **Radio Group:** Displays an HTML radio group form element based on a list of values.
- **Rich Text Editor, Text Area, Text Field:** Allow users to enter textual data. The Text Area field is resizable. Rich Text Editor provides various formatting options. You can specify up to 32,767 bytes for a Text Area or Rich Text Editor item.
- **Select List:** Displays a list of values. The values in the select list are determined by using a shared list of values or a list of values defined at the item level.
- **Shuttle:** Is used to move one or more list elements from left to right.
- **Text Field with autocomplete:** Shows data from a table as you enter text in the field.
- **Yes/No:** Displays a Flip Toggle Switch in mobile user interfaces and as a select list in nonmobile environments.
- **Plug-ins:** Enables developers to declaratively extend, share, and reuse the built-in types available with Oracle Application Express.
- **Slider:** Renders slider item type for mobile applications. This item type enables users to use slide handler to set a value.

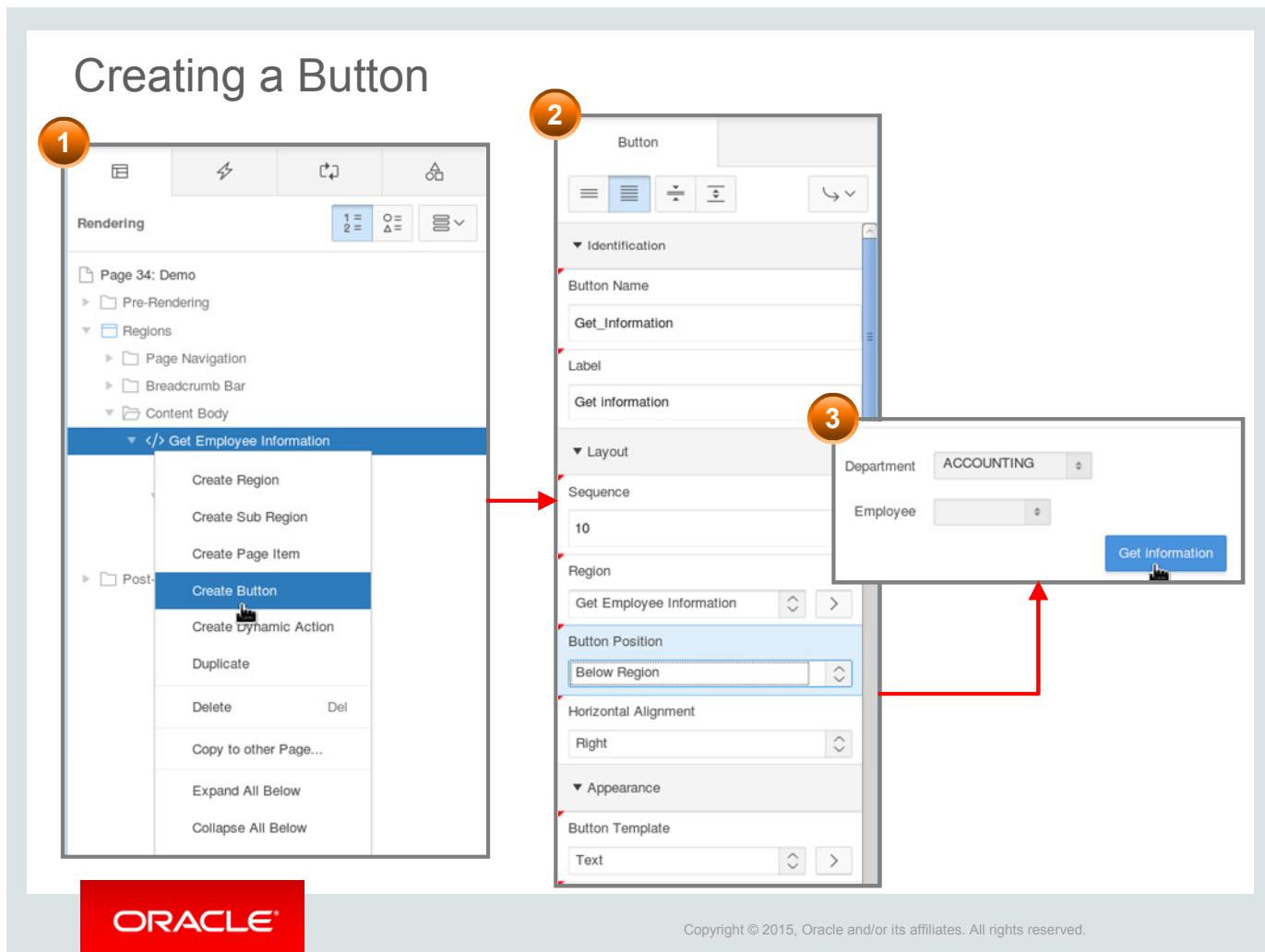
Note: You can create a maximum of 100 items on a page.

Editing an Item



To edit an item, navigate to Page Definition and select the item you want to edit in the Rendering pane. When you select the item, notice that the Property Editor on the right pane displays the list of attributes that can be edited for that particular item. These attributes are categorized as follows:

- Identification, Label, Settings, Layout, Grid, Appearance, Validation, Advanced, Source, Default, Quick Picks, Condition, Read Only, Security, Configuration, Help, Comments, and Audit Information.

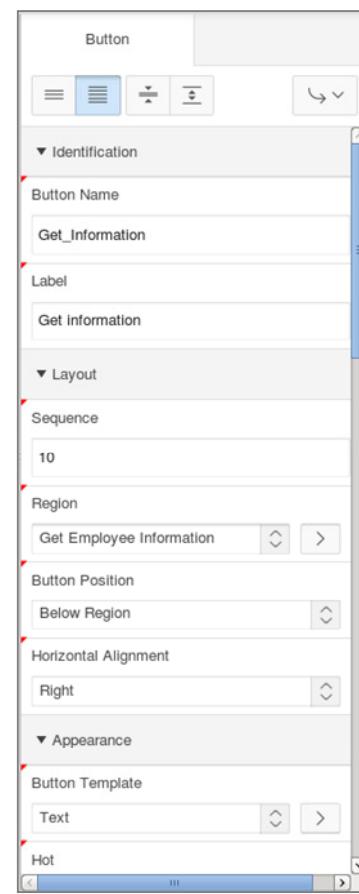
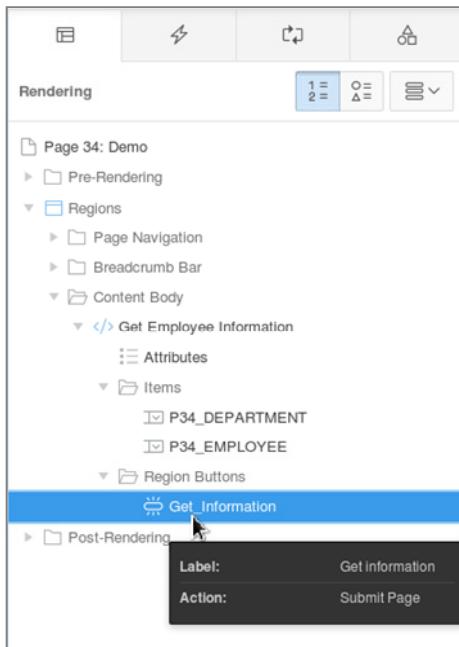


To create a new button, navigate to Page Definition and perform the following steps:

1. Identify the region to contain the button. In the Rendering pane, right-click the region node and select Create Button. Alternatively, you can drag a button from the Buttons Gallery to the Grid Layout.
2. Fill in the details in the Property Editor, and click the Save and Run Page button.
 - Enter a name and label for the button.
 - Specify the Button Position. You can also drag the button in the Grid Layout to the desired position.
 - Select a style for the button.

The button is created. If you run the page and click the button, notice that the page gets submitted. You can now define the actions that are required when the page is submitted.

Editing Button Attributes

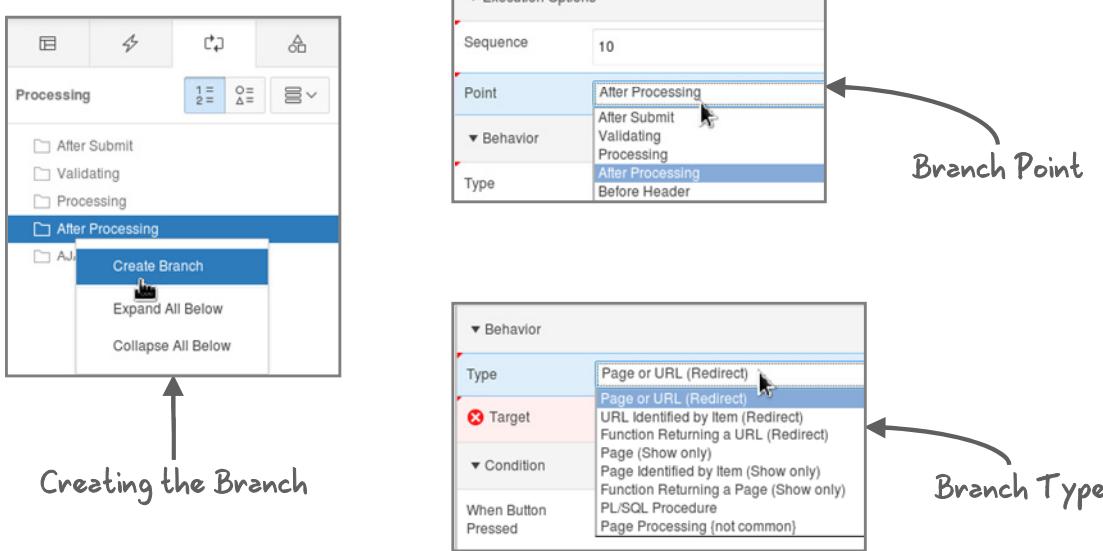


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

After you create a button, you can edit its attributes. To edit the attributes of a button, select the button in the Rendering pane of Page Designer. As soon as you select the button, the Property Editor (the right pane) loads the attributes of the button. You can then edit the attributes from this pane.

Save and run the page to verify that the changes to the attributes have been committed.

Creating a Branch

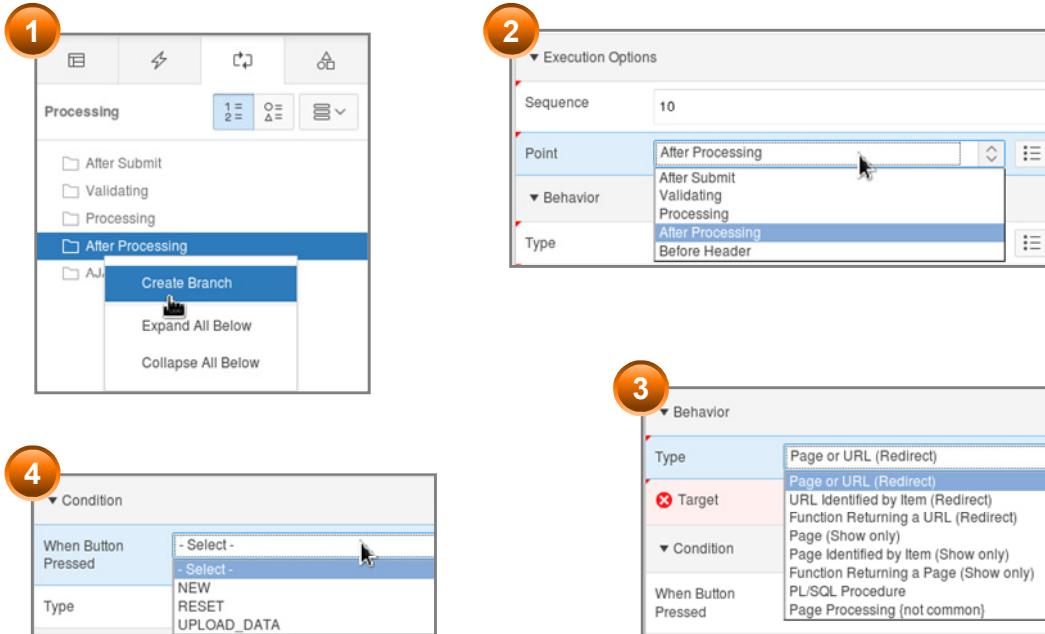


ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can create a new branch by navigating to the Processing tab and specifying the point where you want the branch created. You can specify various branch points and branch types as shown in the slide.

Creating a Branch: Example

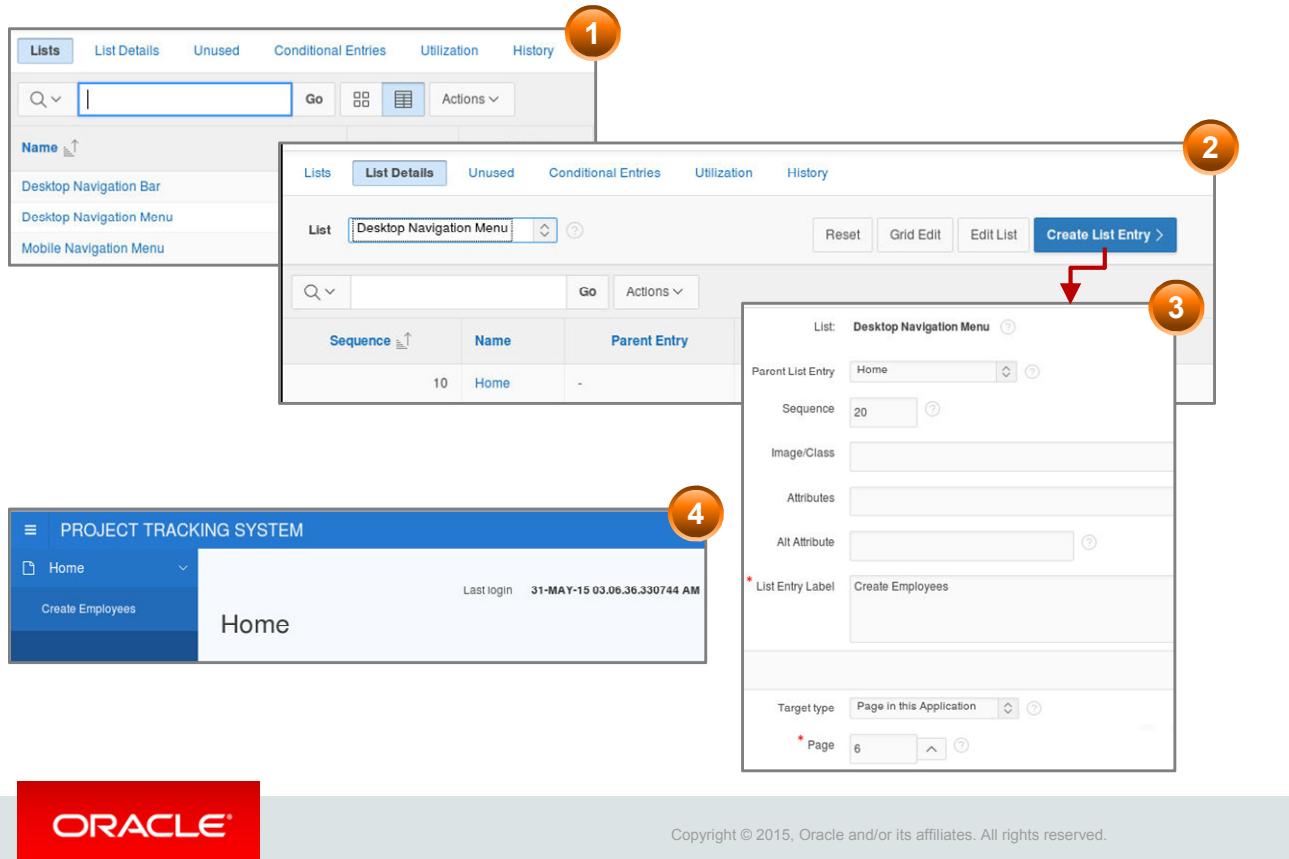


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this example, a branch to another page is created on submit, after processing. Perform the following steps:

1. Navigate to the Processing tab, select the branch point, right-click and select Create Branch.
2. In the Property Editor, specify the branch point.
3. Specify where the branch needs to redirect to by specifying the Behavior > Type and Target.
4. Specify conditions, if any. Click Save.

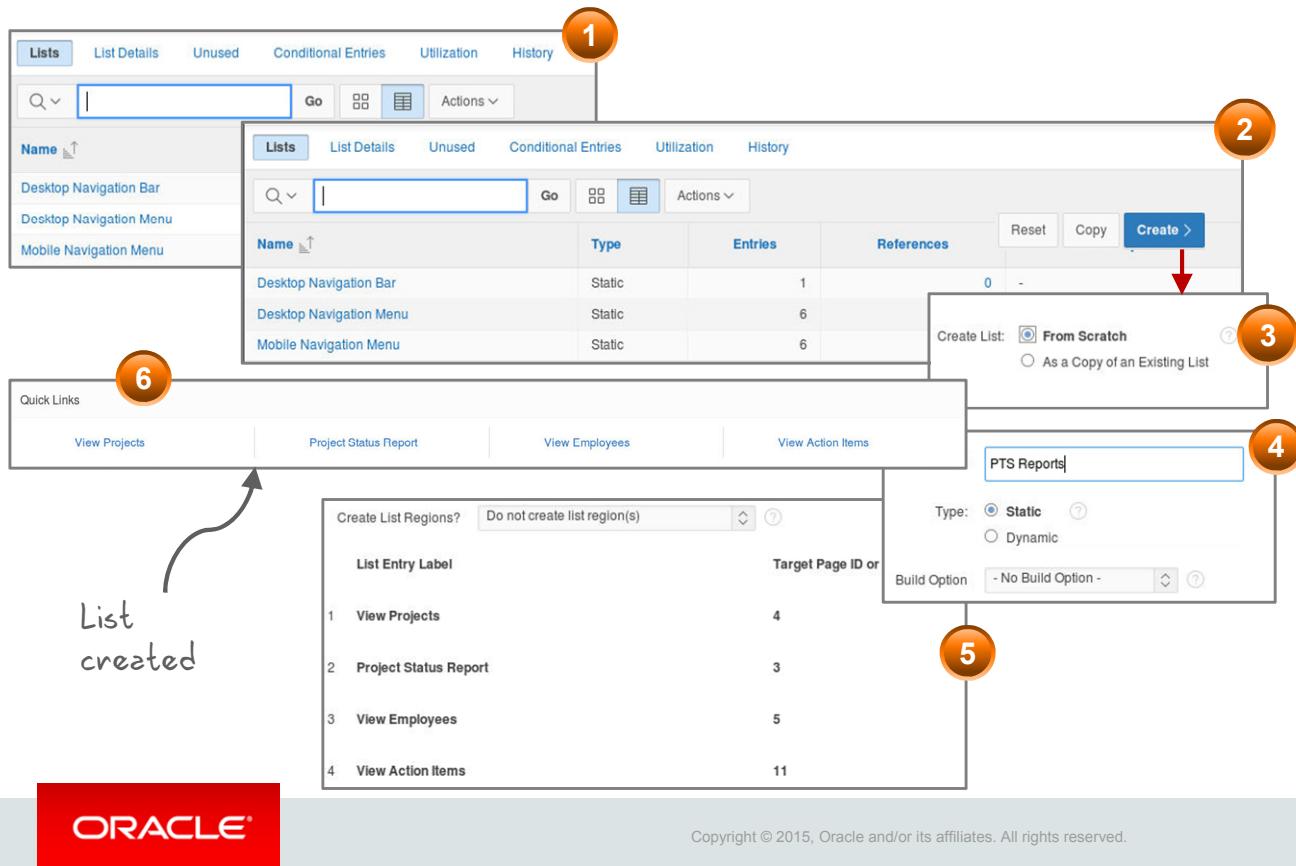
Creating Navigation Menu Entries



To create new Navigation Menu entries:

- Navigate to the application home page
- Click the **Shared Components** icon on the application's home page
- Locate **Navigation** group and click **Navigation Menu**
- Click **Desktop Navigation Menu** and then **Create List Entry**
- Enter details like **Parent List Entry**, **List Entry Label**, **Target Type**, and **Target Page**

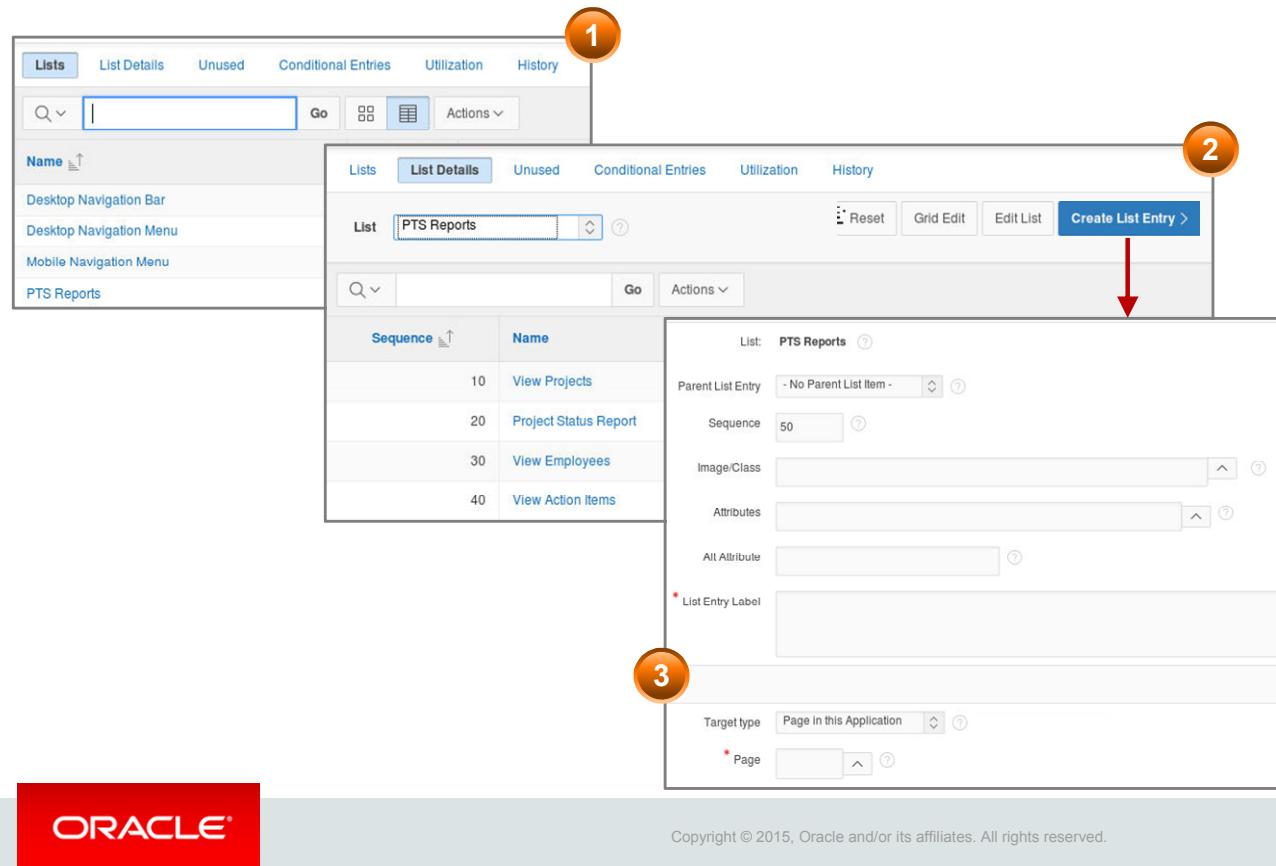
Creating a Static List



To create a static list, click the Create button on the Lists page and perform the following steps:

1. Ensure that From Scratch is selected and click Next.
2. Enter a name for the list. Accept the other defaults and click Next.
3. Enter the text for the list entries and specify the page number that you want to link with each entry. Click Next.
4. The static list is created. You can edit the list to add additional list entries.
5. A list region must be created on a page to display the list. Usually, it is done on the home page.
6. You can create a list region on the current page while creating the List itself.
7. Alternatively, a list region can be created separately on the page where you want to display the list.

Creating List Entries



After you create a list, you can populate the list. You can also create new list entries in lists that are already populated. To create a list entry:

1. Click Create List Entry on the Lists page.
2. Enter the text for the link in the List Entry Label field. On the Target tab, enter the page that you want to associate this list entry with. Click Create.

The list entry is created.

Note: On the Entry tab, if you select a list item for the Parent List Entry field, you can create a hierarchical list.

Creating a Dynamic List

1

2

3

4

5

6

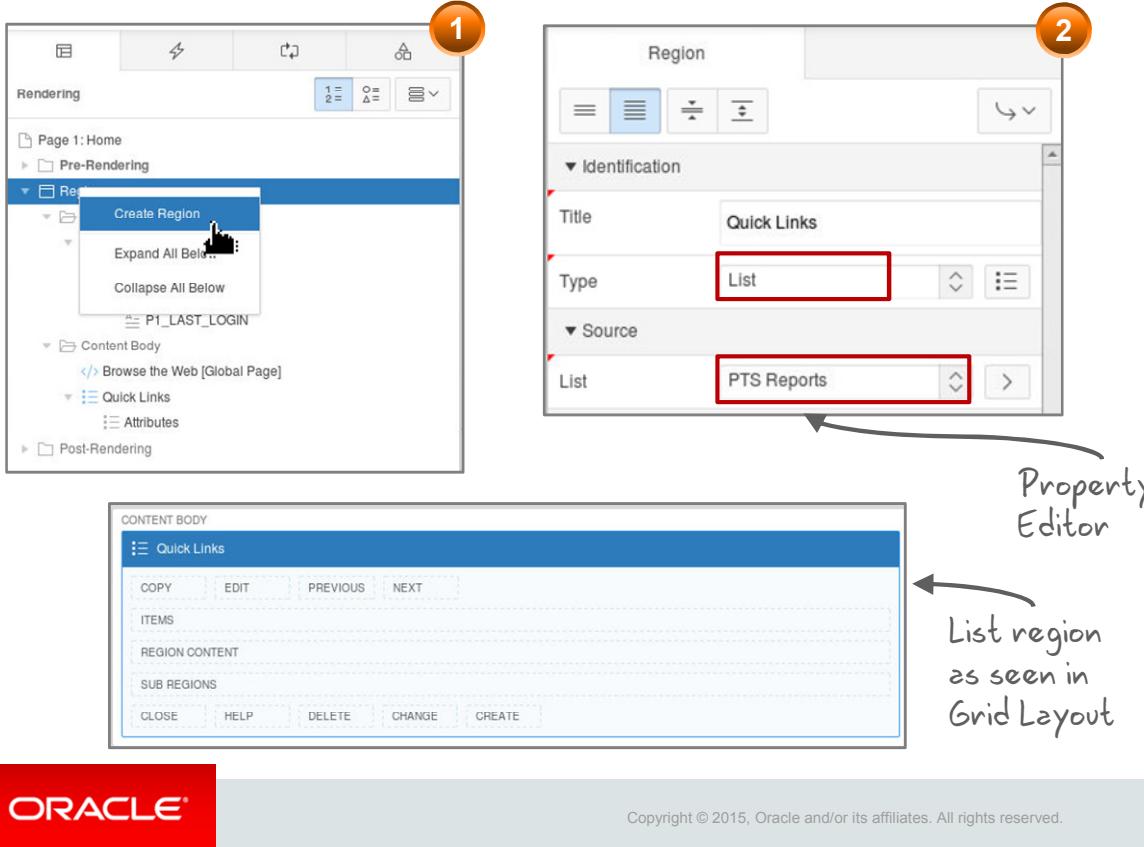
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To create a dynamic list, click the Create button on the Lists page and perform the following steps:

1. Ensure that From Scratch is selected and click Next.
2. Enter a name for the list, select Dynamic for Type, and click Next.
3. Enter the SQL Query to create the list and click Next. You can view examples of SQL queries by clicking the Examples link at the bottom.
4. You can create a list region on the current page. In this example, you accept the defaults and click Create.

The dynamic list is created. You can edit the query to modify the list entries.

Creating a List Region



After you create a list and populate it with values, you can add the list to a page. To add the list to a page, navigate to Page Definition and perform the following steps:

1. Under Rendering, right-click Regions and click Create Region.
2. In the Property Editor of this new region, select the Type as List option and select the list from the List drop-down.
- Note:** You see the list option in the List region property editor only if the application already has a list.
3. (Optional) Specify any conditions for the display of the region.
4. The list region is created on the page.

Summary

In this lesson, you should have learned how to:

- Create a workspace and a workspace administrator
- Create database objects
- Run SQL commands and SQL scripts
- Create a database application
- Create a report
- Create a form on a table with report
- Create a region
- Create and edit page items and buttons
- Create a branch
- Create a navigational menu entry
- Create lists and list entries



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.



Oracle Application Express: Other Features

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Debug PL/SQL remotely in SQL Developer
- Synchronize beta and development feedback



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Agenda

- Debug PL/SQL Remotely in SQL Developer
- Synchronize Beta and Development Feedback



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Remote PL/SQL Debugging with SQL Developer

1. Import packaged application.
2. Set DEBUG DBA privileges.
3. Compile PL/SQL package and package body for Debug.
4. Set remote Debug.
5. Add the DEBUG code to Application.
6. Debug Application with the DEBUG code.

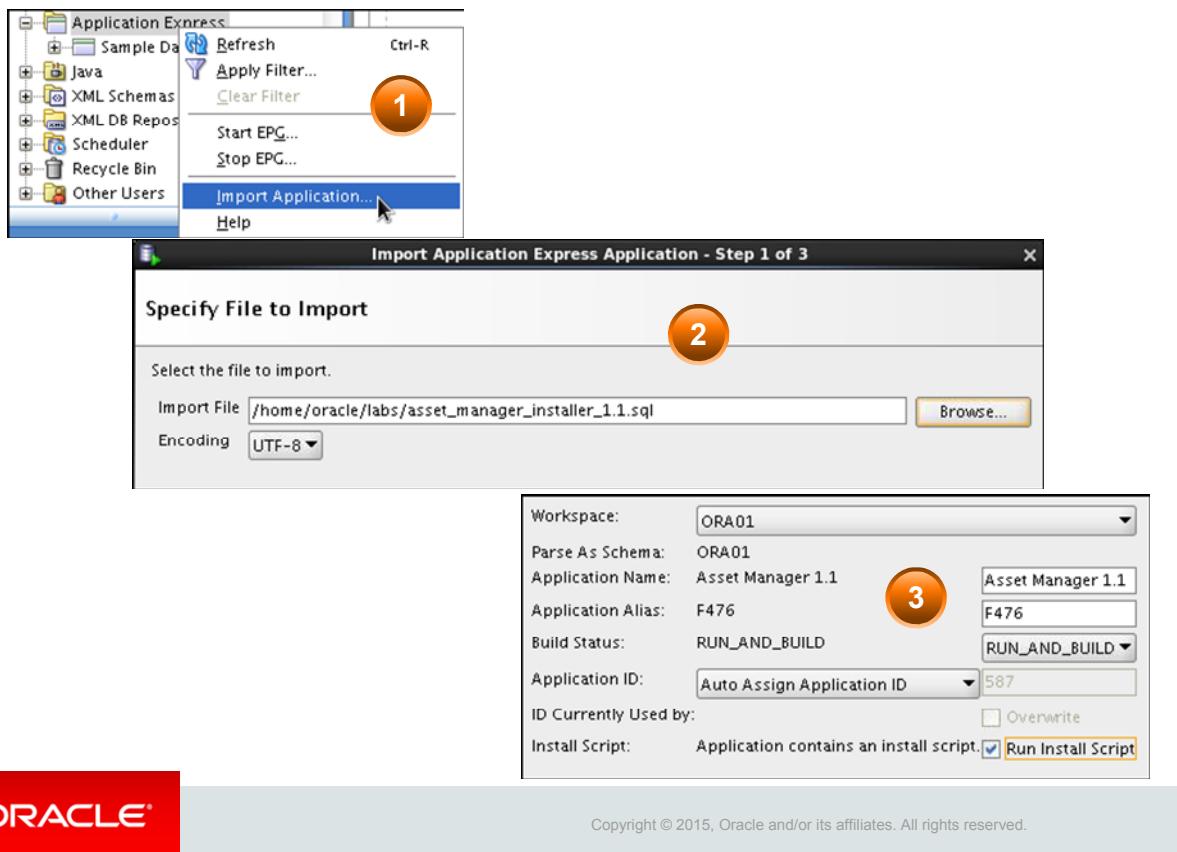


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

SQL Developer can be used to remotely debug PL/SQL, which is very useful when an application is not producing expected results. Remote debugging can be used to step through complex PL/SQL processing that is not performing its intended function. Reviewing data values as the program steps through the code can greatly assist in determining incorrect logic.

Typically, remote debugging is used while in a development or test environment, not in a production environment.

Importing Your Packaged Application



SQL Developer allows you to import a packaged application and run the installation script to create the supporting objects in your database. In this example, you import the Asset Manager packaged application.

To import the packaged application:

1. In SQL Developer, logged in as the APEX user, right-click the Application Express node and click Import Application.
2. Select the SQL file that contains the packaged application and click Next.
3. Specify the workspace that you want to import into and select the Run Install Script check box to install the supporting objects. Click Next.
4. Click Finish.

Setting DEBUG DBA Privileges

In SQL Developer, create a connection to SYSTEM and execute the following commands:

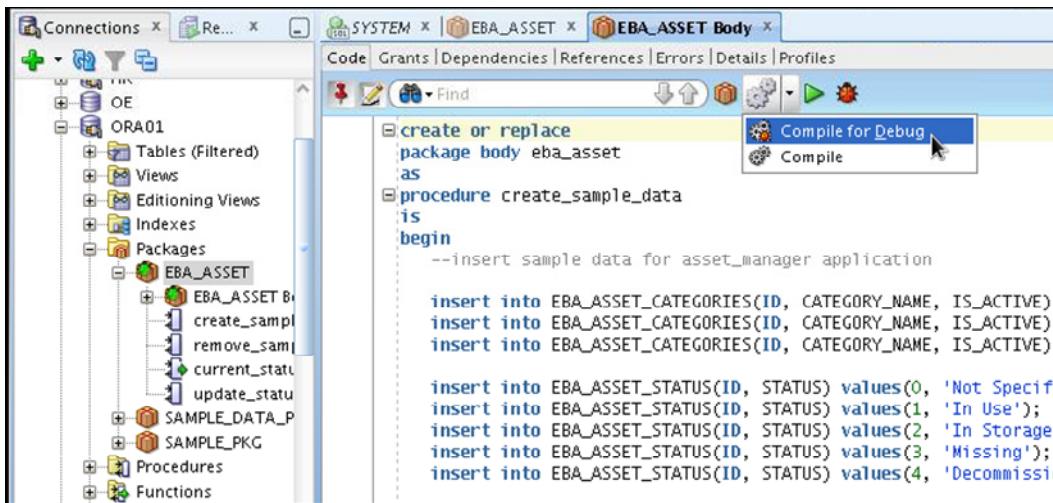
```
grant DEBUG CONNECT SESSION to <APEX_USER>;  
grant DEBUG ON <APEX_USER>. <package name> to PUBLIC;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To use the Debugging facility for a particular package, the connecting user (in this case, APEX_USER) must have the DEBUG CONNECT SESSION privilege. In addition, you must grant DEBUG ON <APEX_USER>. <package name> to PUBLIC.

Compiling the PL/SQL Package and Package Body for Debug



ORACLE®

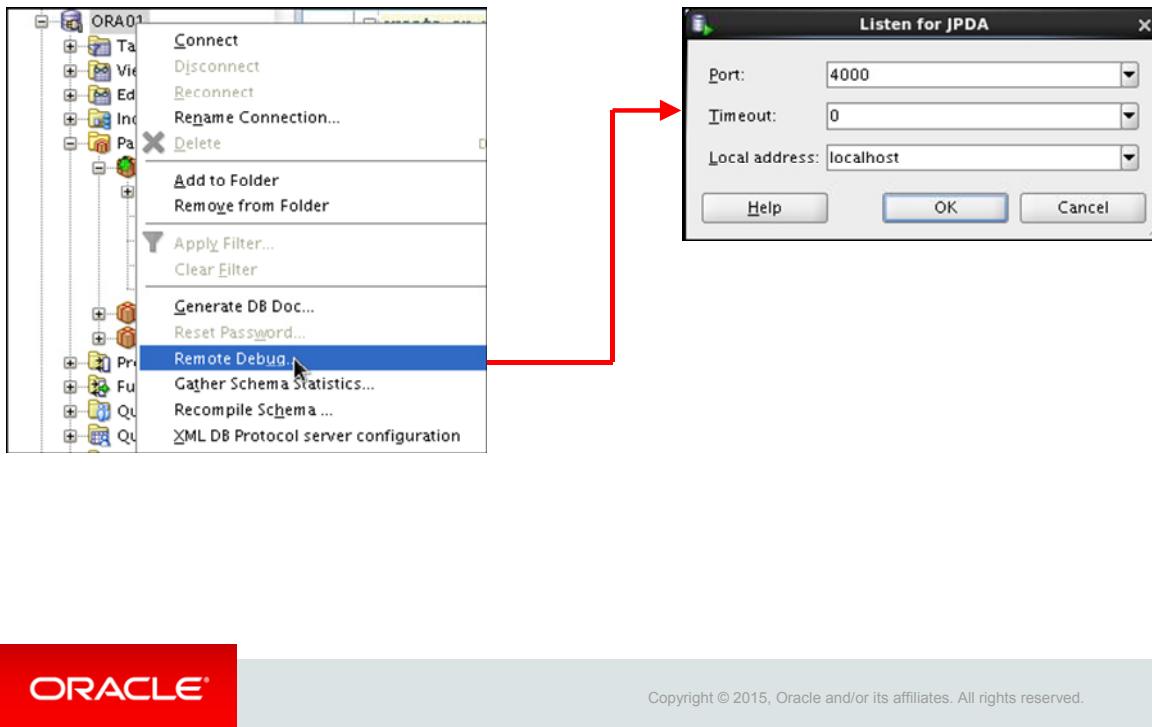
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

For the PL/SQL Debugger to debug your code, you must compile it for debug.

1. Expand Packages.
2. Expand your package and then right-click your package body. Select Edit.
3. Your package body is displayed in edit mode. Click the Compile icon and select Compile for Debug.

Note: If you receive warnings, it is okay to proceed.

Setting the Remote Debug



To debug remotely through SQL Developer, you must set Remote Debug. Right-click the APEX user connection and select Remote Debug. When the Listen for JPDA dialog box appears, enter the listening port number and the IP address of the machine with the database.

You can also set a breakpoint in the code you want to debug so that when the debugger reaches the breakpoint, the debugger stops. You can then review the code in SQL Developer to see where the problem is.

Adding the DEBUG Code to Your Process

The screenshot shows the Oracle Application Express interface for managing processes. The process is named "Update Asset Status" and is of type "PL/SQL anonymous block". The source code contains PL/SQL code that checks if a debug parameter is set and performs database operations accordingly.

```

* Process [Download Source]
IF :DEBUG='YES' THEN
    dbms_debug_jdwp.connect_tcp('localhost',4000);
END IF;
eba_asset.update_status(:P11_ASSET_ID);
IF :DEBUG='YES' THEN
    dbms_debug_jdwp.disconnect;
END IF;

```

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

For SQL Developer to be used remotely, you must turn it on and off where you want the debugger to be used. Add the following code to start and turn off the debugger:

```

IF :DEBUG='YES' THEN
    dbms_debug_jdwp.connect_tcp('localhost',4000);
END IF;
<statement goes here>
IF :DEBUG='YES' THEN
    dbms_debug_jdwp.disconnect;
END IF;

```

In Oracle Application Express, a URL parameter REMOTE is available. If REMOTE is set, you do not need to specify the code surrounding the statements. The debugger will be run for the entire page.

When the page is executed in Application Express, you start DEBUG in the Developer toolbar. Make a change to the field in the code where the DEBUG code appears—in this case, the Inventory Status—and click Apply Changes. At this point, SQL Developer takes over and if there is a breakpoint in the SQL Developer code, you can now step through the code in SQL Developer to see where the problem is.

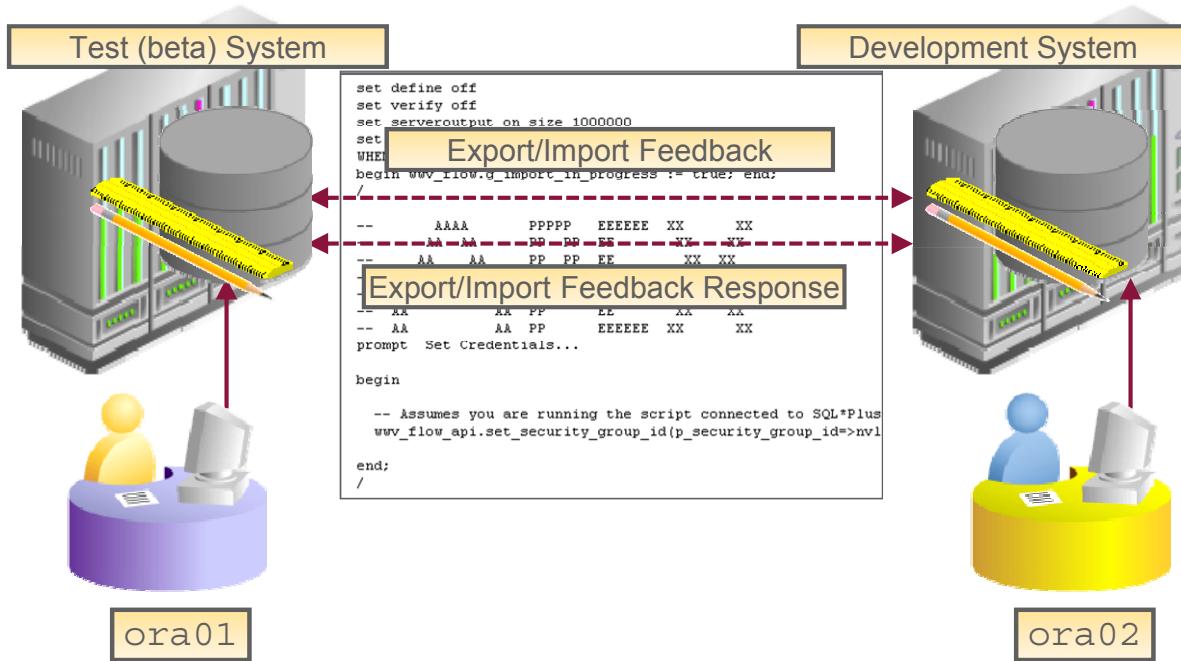
Agenda

- Debug PL/SQL Remotely in SQL Developer
- Synchronize Beta and Development Feedback



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Synchronizing Beta and Development Feedback



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You use the Export Feedback utility to synchronize feedback from a beta or test system to the development system.

The example in the slide shows the feedback synchronization process between a test system and a development system. In the example, ora01 is the workspace created in the test system and ora02 is the workspace created in the development system. You export feedback from the test system to your development system and respond to the feedback. You want the test system end users to view the developers' responses to their feedback. Therefore, you export the feedback response from the development system and import it into the test system. Test system users can simply query the APEX_TEAM_FEEDBACK view for responses.

Note that the export feedback feature is used to synchronize feedback between different APEX installations. You do not use it to copy feedback from one workspace to another on the same APEX installation.

Reviewing the Feedback Synchronization Source Identifier

The screenshot shows the 'Manage Workspaces' interface in Oracle Application Express. It displays two workspaces side-by-side:

- Workspace in the test system:** Workspace Identifier: 2268532096926535, Workspace Status: ASSIGNED, Workspace Name: ORA01, First Schema Provisioned: ORA01, Feedback Synchronization Source Identifier: ORA01, Allow workspace to be automatically purged: Yes, Resource Consumer Group: (empty), Builder Notification Message: (empty).
- Workspace in the development system:** Workspace Identifier: 2382700459930938, Workspace Status: ASSIGNED, Workspace Name: ORA02, First Schema Provisioned: ORA02, Feedback Synchronization Source Identifier: ORA02, Allow workspace to be automatically purged: Yes, Resource Consumer Group: (empty), Builder Notification Message: (empty).

Both workspaces have their 'Feedback Synchronization Source Identifier' set to 'ORA01'. The 'Workspace in the test system' has its name set to 'ORA01' and the 'Workspace in the development system' has its name set to 'ORA02'. The 'Feedback Synchronization Source Identifier' field is highlighted with a yellow box.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Before you export or import feedback, make sure that the workspace names used in the production and development systems are different. If the workspace names are the same, you need to modify the Feedback Synchronization Source Identifier for one of the workspaces. To change the identifier:

1. Log in as the admin user.
2. Navigate to Manage Workspaces > Workspace Reports > Existing Workspaces.
3. Click your workspace name.
4. Review the Source Identifier. If you need to change it, click Edit.
5. Enter a different value for Feedback Synchronization Source Identifier.
6. Click Apply Changes.

Synchronizing Beta and Development Feedback: Example

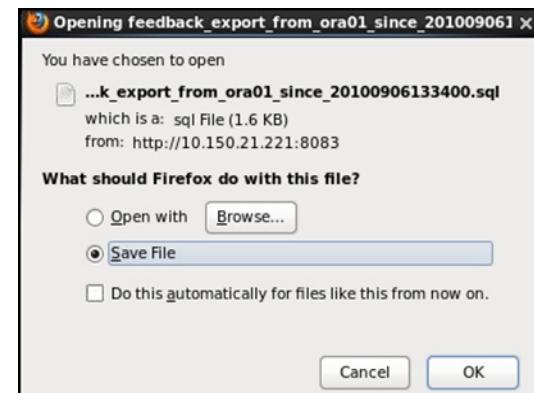
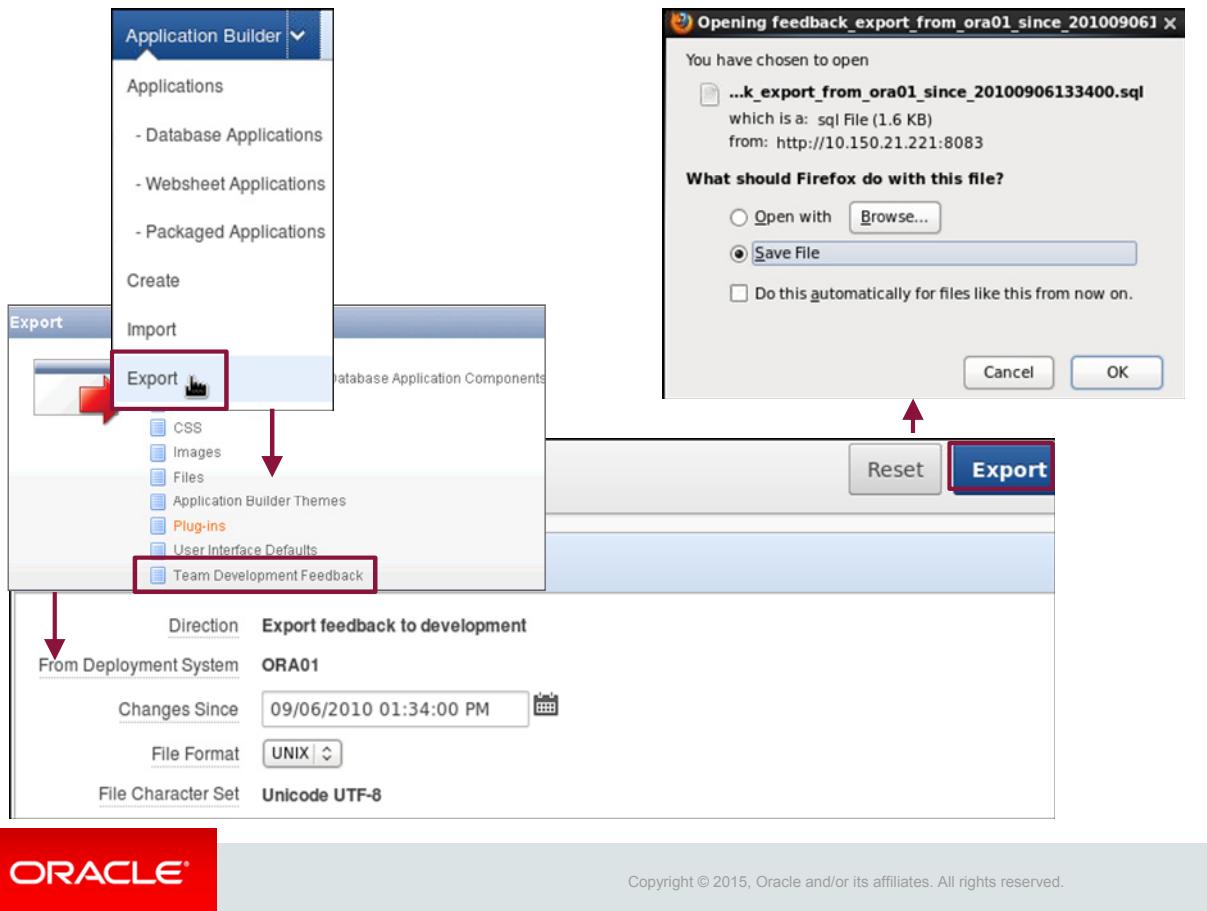
- On the test (beta) system:
 1. Create and customize a feedback page in an application.
 2. Review the submit feedback process.
 3. Submit feedback by using the Feedback navigation bar entry.
 4. Export the feedback.
- On the development system:
 1. Import the feedback from production.
 2. Review feedback and follow up.
 3. Export the feedback response.
- On the test system, import the feedback response from development and create a report to show the feedback list.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The slide shows the steps involved in the example.

Exporting the Feedback from Beta System

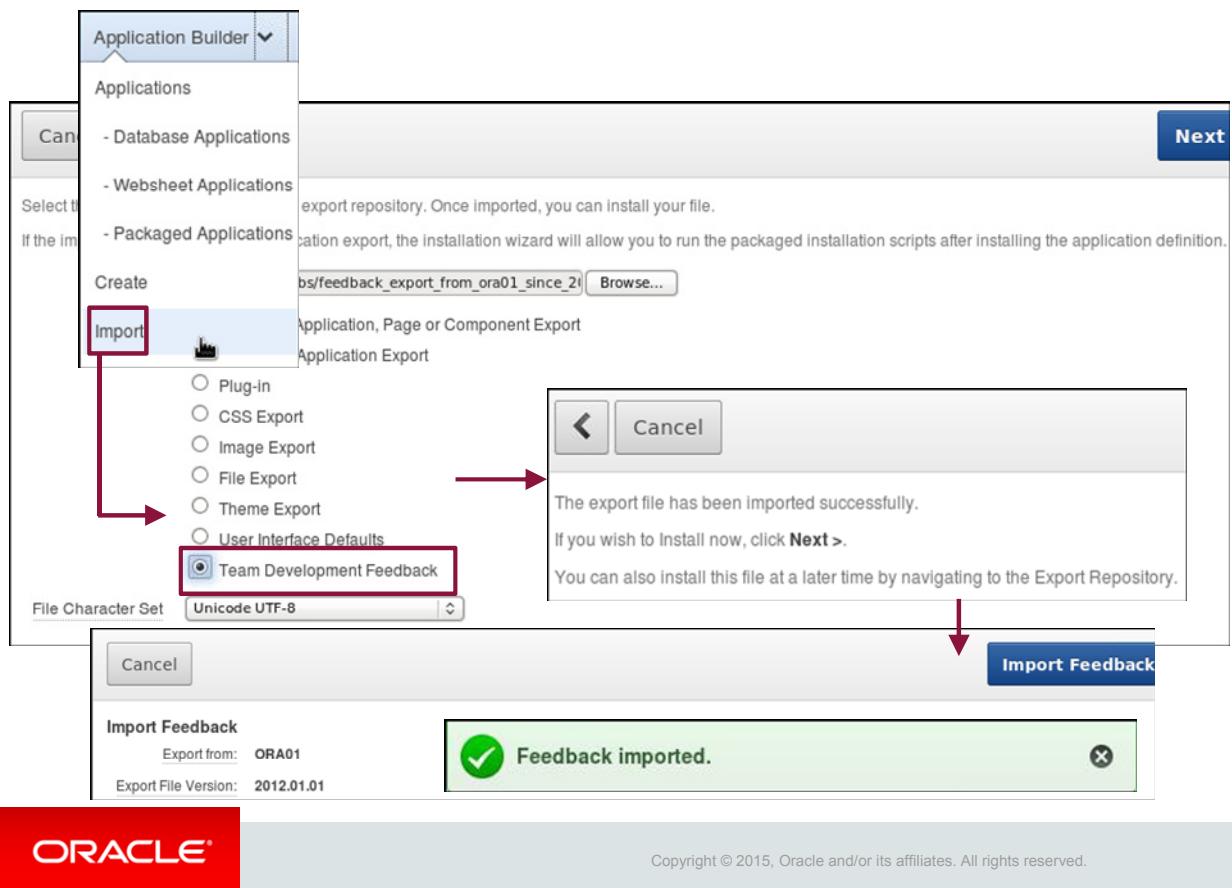


You want developers to review and follow up on the feedback entered by users in the beta system. To facilitate this, you use the Export Feedback utility to export feedback into a .sql file. After the feedback is exported, you import it back to the development system.

The example in the slide shows how to export feedback from the ora01 beta system. To do so, perform the following steps:

1. On the Application Builder home page, click Export.
2. Select Team Development Feedback.
3. Note that the Export Feedback page shows From Deployment System ORA01. For Changes Since, select the date of the oldest feedback to export. All feedback from this selected date until the current date is exported.
4. Click Export.
5. Follow on-screen instructions to save the exported .sql file.

Importing the Feedback into Development System



After you export the feedback from the deployment system, you import it into the target Oracle Application Express instance for development. To do so, perform the following steps:

1. On the Application Builder home page, click Import.
2. Click Browse and select the feedback file that you exported from the deployment system. Select the Feedback option button and click Next.
3. Click Next.
4. On the Import Feedback page, note Export from: ORA01. Click Import Feedback.

The feedback is now available on the development system. Developers can now edit, follow up, and respond to the feedback.

Responding to the Feedback

The screenshot shows the Oracle Application Express Feedback page. A specific feedback item for 'ORA01-6' is selected, indicated by a red box around its row in the grid. A red arrow points from this row to a callout box containing the 'Developer Comment' and 'Public Response' sections. The 'Developer Comment' section contains the text 'Acknowledged, will be fixed in the next build.' The 'Public Response' section also contains the same text. The Oracle logo is visible at the bottom left, and the copyright notice 'Copyright © 2015, Oracle and/or its affiliates. All rights reserved.' is at the bottom right.

The example in the slide shows how to respond to feedback on the development system. To respond to feedback, navigate to Team Development > Feedback and click the Feedback tab.

Developers can perform the following:

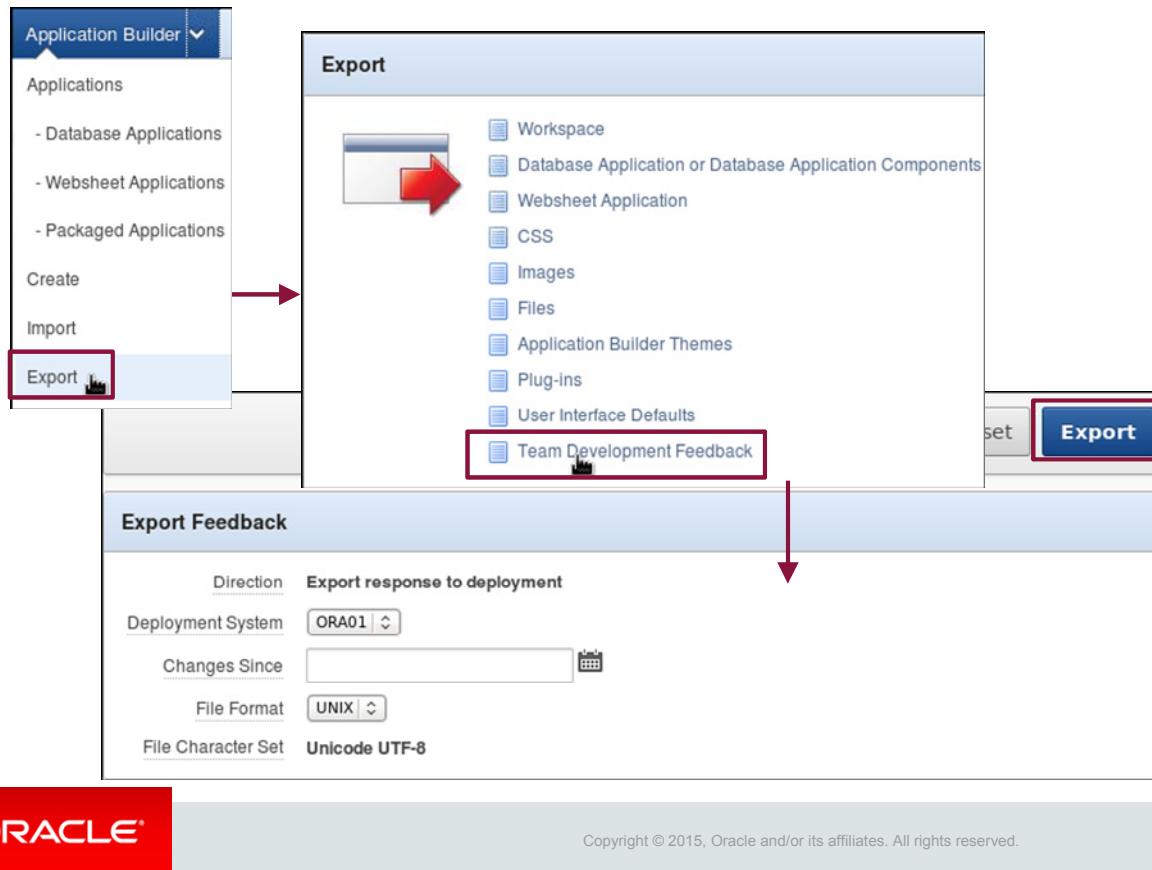
- Edit the feedback
- Follow up
- Log feedback as bug
- Log feedback as Todo
- Log feedback as feature

In the example in the slide, click Edit. Under Developer Comment and Public Response, enter appropriate text and click Apply Changes.

Note that developer comments are considered as internal comments and are not synchronized. That is, they are not visible to the user on the deployment system. But public response gets synchronized and is visible to the deployment system users.

For more details about managing bugs and feedback, see *Oracle Application Express Application Builder User's Guide Release 4.2*.

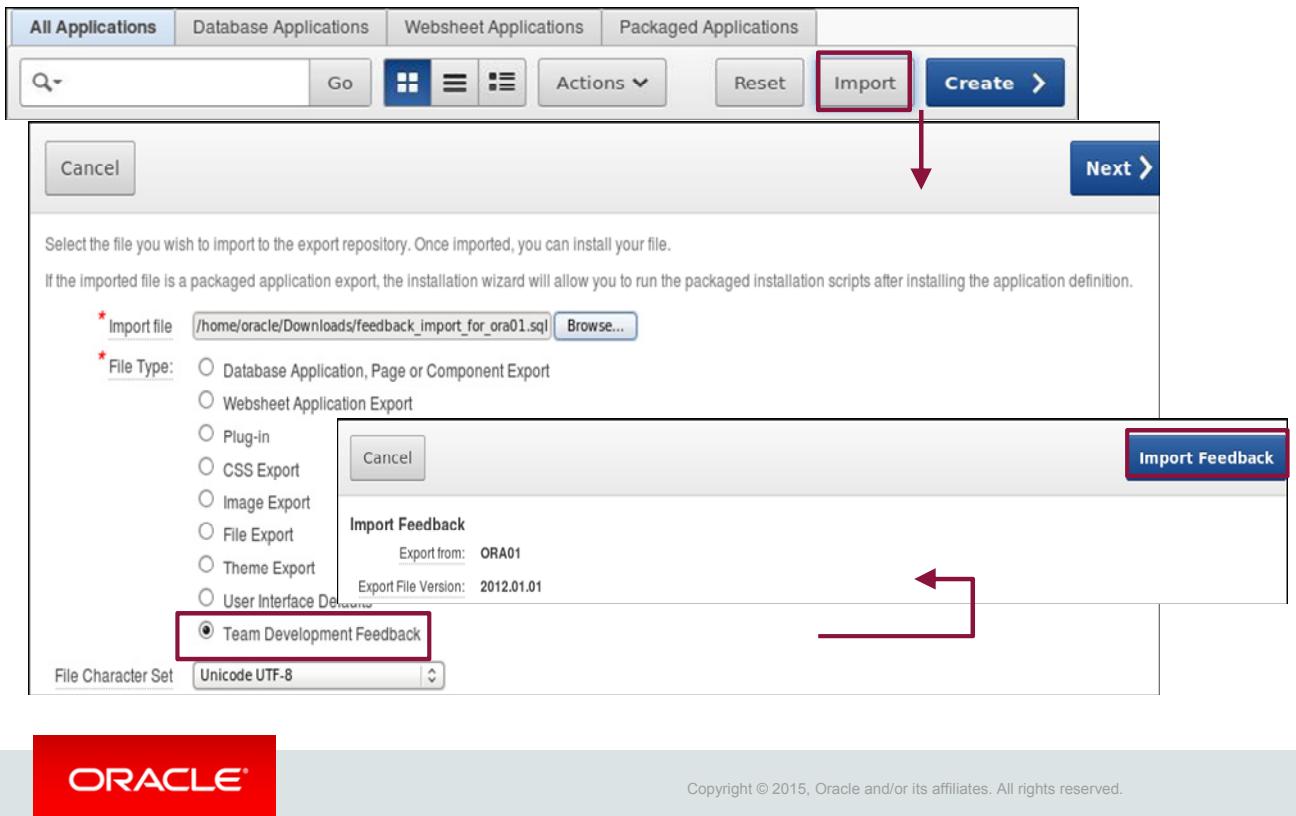
Exporting the Feedback Response from Development System



After the developers review and respond to feedback, you want to export the responses to the deployment system. Perform the following steps to create an export file from the development system:

1. On the Application Builder home page, click Export.
2. Select Team Development Feedback.
3. Note that the Export Feedback page now shows “Export response to deployment.” If exporting from a development system, the export process has an additional attribute called Deployment System where you specify which deployment system you want to synchronize the updated feedback to. For Changes Since, select the date of the oldest feedback to export. All feedback and response from this selected date until the current date is exported.
4. Click Export.
5. Follow on-screen instructions to save the exported .sql file.

Importing the Feedback Response into Deployment System



Perform the following steps to import the feedback response into the deployment system:

1. On the Application Builder home page, click Import.
2. Click Browse and select the feedback file that you exported from the development system. Select the Feedback radio button and click Next.
3. Click Next.
4. Click Import Feedback.

The feedback response is now available on the deployment system.

Creating a Feedback Review Report

Use this query

```
select FEEDBACK_NUMBER,FEEDBACK,APPLICATION_NAME,PAGE_NAME,PUBLIC_RESPONSE  
from APEX_TEAM_FEEDBACK
```

Interactive report that shows feedback details and response

FEEDBACK_NUMBER▼	FEEDBACK	APPLICATION_NAME	PAGE_NAME	PUBLIC_RESPONSE
-	Associate this feedback with an application	-	-	You have not logged into the application.
10	There is no Print option available	Checklist Manager	Home	Please provide test environment.
6	There are 2 feedback links, please remove 1	Feedback	Home	Acknowledged, will be fixed in the next build.
8	Good to add images to the page	Feedback	Home	-
9	Please check the Login Page links	Feedback	Home	-



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You may want to create a report on the production system to show feedback list and responses. For the report, you write a query against the `APEX_TEAM_FEEDBACK` view.

To write a query against the `APEX_TEAM_FEEDBACK` view:

1. Navigate to the Application Builder home page. Under Tasks, click Application Express Views.
2. Select the `APEX_TEAM_FEEDBACK` view.
3. Click Select Columns tab.
4. Select the appropriate columns to display in your report. Click Results to see the data result. This is the data you want to include in your report.
5. Click Query to review and copy the query that was executed.

If you want to view the follow-up entered by developers, you can query the `APEX_TEAM_FEEDBACK_FOLLOWUP` view.

Summary

In this lesson, you should have learned how to:

- Debug PL/SQL remotely in SQL Developer
- Synchronize beta and development feedback

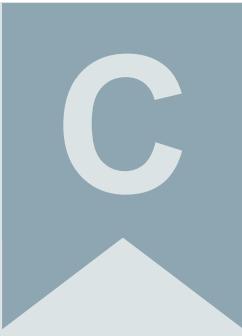


ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only



Extending Your Application with User-Defined Error Handling

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Upload and execute necessary error handling scripts
- Associate error handling function with your page
- Test error handling for your Page



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

This lesson discusses how to extend your application to use the user-defined error handling function.

Error Handling Function

The screenshot shows a web application interface for managing departments. The top navigation bar includes 'Error Handling', 'Home', and 'Maintain Departments'. A yellow warning icon indicates '1 error has occurred' with the message 'Name is already in use! (Row 11)'. Below this, a table titled 'Update Departments' lists eight rows of department data. The last row, with Department ID 101 and Department Name 'Finance', has its 'Department Name' field highlighted with a red border, indicating it is the source of the error. The table columns are: Department ID, Department Name, Manager Id, and Location Id.

	Department ID	Department Name	Manager Id	Location Id
<input type="checkbox"/>	10	Administration	200	1700
<input type="checkbox"/>	20	Marketing	201	1800
<input type="checkbox"/>	30	Purchasing	114	1700
<input type="checkbox"/>	40	Human Resources	203	2400
<input type="checkbox"/>	50	Shipping	121	1500
<input type="checkbox"/>	60	IT	103	1400
<input type="checkbox"/>	70	Public Relations	204	2700
<input type="checkbox"/>	90	Executive	100	1700
<input type="checkbox"/>	100	Finance	108	1700
<input type="checkbox"/>	101	Finance	108	1700

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Error Handling Function is a feature that can be used when you want to have full control over your application when an error occurs. It can include errors raised by validation, process, and the Application Express engine itself. It can be used to log the error, modify the error message text, and define where it can be displayed, such as Inline with Field and in Notification, Inline with Field, Inline in Notification, and On Error Page.

1. Creating Constraint Lookup Table

Stores the constraints and messages for user-defined violations.

```
create table CONSTRAINT_LOOKUP
(
  CONSTRAINT_NAME VARCHAR2(30) primary key,
  MESSAGE VARCHAR2(4000) not null
);
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You need to create a table where you are going to store the name of the constraint and the message to show when the constraint is violated.

2. Creating Error Handling Function

```
create or replace function apex_error_handling_example (
    p_error in apex_error.t_error )
    return apex_error.t_error_result
is
    l_result          apex_error.t_error_result;
    l_reference_id    number;
    l_constraint_name varchar2(255);
begin
    l_result := apex_error.init_error_result (p_error => p_error );
    if p_error.is_internal_error then
        if p_error.apex_error_code <> 'APEX.AUTHORIZATION.ACCESS_DENIED' then
            l_result.message := 'An unexpected internal application error has occurred. ' ||
                'Please get in contact with XXX and provide ' ||
                'reference# '||to_char(l_reference_id, '999G999G999G990') ||
                ' for further investigation.';
            l_result.additional_info := null;
        end if;
    else
        l_result.display_location :=
            case when l_result.display_location = apex_error.c_on_error_page then
                apex_error.c_inline_in_notification
            else l_result.display_location end;
    .....
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

1. If it is an internal error raised by APEX, such as an invalid statement or code which cannot be executed, the error text might contain security sensitive information. To avoid this security problem, the error is rewritten to a generic error message and the original error message is logged for further investigation by the help desk.
2. Access Denied errors raised by application or page authorization should still show up with the original error message.
3. Log error for example with an autonomous transaction and return l_reference_id as reference#

```
l_reference_id := log_error (p_error => p_error );
```
4. Change the message to the generic error message, which does not expose any sensitive information.

2. Create Error Handling Function

```
if p_error.ora_sqlcode in (-1, -2091, -2290, -2291, -2292) then
    l_constraint_name := apex_error.extract_constraint_name (p_error => p_error );
    begin
        select message
        into l_result.message
        from constraint_lookup
        where constraint_name = l_constraint_name;
        exception when no_data_found then null;
    end;
end if;
if p_error.ora_sqlcode is not null and l_result.message = p_error.message then
    l_result.message := apex_error.get_first_ora_error_text (p_error => p_error );
end if;
if l_result.page_item_name is null and l_result.column_alias is null then
    apex_error.auto_set_associated_item (
        p_error          => p_error,
        p_error_result => l_result );
end if;
end if;

return l_result;
end apex_error_handling_example;
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

5. If the constraint violation is like the following:

- ORA-00001: Unique constraint violated
- ORA-02091: Transaction rolled back (can hide a deferred constraint)
- ORA-02290: Check constraint violated
- ORA-02291: Integrity constraint violated - parent key not found
- ORA-02292: Integrity constraint violated - child record found

A friendly error message is used from the constraint lookup configuration. If the constraint is not in the lookup table, the original ORA error message is used.

6. If an ORA error has been raised, for example a `raise_application_error(-20xxx, '...')` in a table trigger or in a PL/SQL package called by a process, the error is not in the lookup table, then the actual error text and not the full error stack with all the ORA error numbers is displayed.
7. If no associated page item/tabular form column has been set, the `apex_error.auto_set_associated_item` is used to automatically guess the affected error field by examining the ORA error for constraint names or column names.

Setting Up an Example to Test Error Handling

1. Add a Unique Constraint.

```
alter table DEPARTMENTS  
add constraint DEPT_DNAME_UK unique (DEPARTMENT_NAME);
```

2. Insert the Constraint and Message into the Constraint Lookup Table.

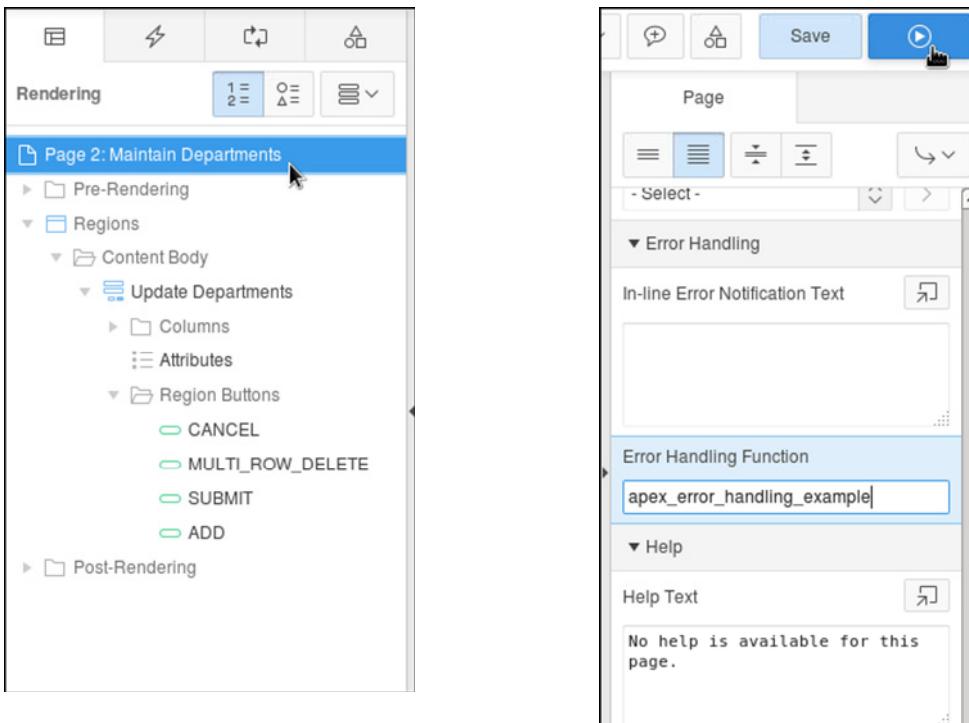
```
insert into CONSTRAINT_LOOKUP  
  (CONSTRAINT_NAME, MESSAGE)  
values  
  ('DEPT_DNAME_UK', 'Name is already in use!');
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The example in the slide adds a unique constraint to the DEPARTMENTS table and then inserts the constraint with a message into the constraint lookup table.

Associating the Error Handling Function with Your Page



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this example, you edit the page definition for Maintain Departments and enter the function `apex_error_handling_example` in the Error Handing Function area, and click Save and Run.

Testing the Page for Error Handling

The screenshot shows two views of the 'Maintain Departments' page. The top view displays an error message: '1 error has occurred' with the detail 'Name is already in use! (Row 11)'. The bottom view shows the data entry screen where a new row is being added. A red box highlights the last row, which has a Department ID of 1700, a Department Name of Finance, and a Manager ID of 201. This row violates the constraint because Manager 201 is already assigned to Department 108.

Department Id	Department Name	Manager Id
30	Purchasing	114
40	Human Resources	203
50	Shipping	121
60	IT	103
70	Public Relations	204
80	Sales	145
90	Executive	100
100	Finance	108
(null)	Finance	201
		1700

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this example, you run the Maintain Departments page and add a row to the tabular form. Enter a Department Name and Location, select a Manager that is already assigned to another department, and click Submit. Notice that the message you entered into the constraint lookup table is used.

Summary

In this lesson, you should have learned how to:

- Upload and execute necessary error handling scripts
- Associate error handling function with your page
- Test error handling for your Page



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.



Migrating a Desktop Application to a Responsive Theme

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Explain Responsive Web Design
- Create and switch to Theme 25
- Modify application components to take advantage of different responsive options



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learn what responsive web design is and switch the existing theme in an application to Theme 25, which is a responsive theme. You modify the application components to take advantage of different responsive options.

What Is Responsive Web Design?

Responsive Web Design (RWD) is an approach to web design in which a designer intends to provide an optimal **viewing experience**—easy reading and navigation with a minimum of resizing, panning, and scrolling—**across a wide range of devices** (from desktop computer monitors to mobile phones).
(Wikipedia)



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

What Is Responsive Web Design?

Desktop UI

The screenshot shows a desktop application window titled "Order Management Application". The window has a blue header bar with the title and navigation links for "Home", "Customers", and "Orders". Below the header is a search bar with a dropdown menu, a "Go" button, and an "Actions" dropdown. A "Create" button is also present. The main area contains a table with customer data, including columns for Customer Id, Cust First Name, Cust Last Name, Street Address, Postal Code, City, State Province, and Country. The table has 8 rows of data. At the bottom of the application window is a toolbar with various icons and labels such as "Home", "Application 108", "Edit Page 2", "Session", "View Debug", "Debug", "Show Grid", "Quick Edit", and "Help".

Customer Id	Cust First Name	Cust Last Name	Street Address	Postal Code	City	State Province	Country
216	Orson	Koirala	810 Race St	19107	Philadelphia	PA	US
217	Bryan	Huston	4901 Locust Ln	17109	Harrisburg	PA	US
218	Bryan	Dvrrie	3376 Perrysville Ave	15214	Pittsburgh	PA	US
219	Ajay	Sen	220 Penn Ave # 300	18503	Scranton	PA	US
220	Carol	Jordan	135 S 18Th St # 1	19103	Philadelphia	PA	US
221	Carol	Bradford	522 Swede St	19401	Norristown	PA	US
222	Cary	Stockwell	7708 City Ave	19151	Philadelphia	PA	US
223	Cary	Olin	1801 Lititz Pike	17601	Lancaster	PA	US
224	Clara	Krige	101 E Olney	19120	Philadelphia	PA	US

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

ORACLE

In the example in the slide, you see the user interface of the application when on a desktop.

What Is Responsive Web Design?

Tablet UI

The screenshot shows the Order Management Application interface on a tablet. At the top, there's a navigation bar with 'apex' and 'Log Out' buttons, followed by tabs for 'Home', 'Customers', and 'Orders'. Below the navigation is a search bar with a dropdown arrow and a 'Go' button. A 'Actions' dropdown menu is open. Underneath is a 'Create' button. The main area contains a grid of customer data with the following columns: Customer Id, Cust First Name, Cust Last Name, Street Address, and Postcode. There are 11 rows of data, with the last row partially visible. Each row includes a small edit icon. At the bottom, there's a toolbar with various icons and buttons like 'Home', 'Edit Page 2', 'Session', 'View Debug', 'Debug', 'Show Grid', 'Quick Edit', and a gear icon.

ORACLE®

Mobile UI

The screenshot shows the same Order Management Application interface on a mobile device. The layout is significantly different due to the smaller screen. The grid of customer data is stacked vertically, requiring users to scroll down to view all the rows. The columns remain the same: Customer Id, Cust First Name, Cust Last Name, Street Address, and Postcode. The rest of the interface elements (navigation bar, search bar, actions, create button, toolbar) are also present but adapted to fit the mobile screen.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When on a Tablet or Mobile device, the user interface can look similar but with minor differences. The difference between devices is handled through CSS3 Media Queries and a Grid Layout structure. In the example in the slide, on the Tablet UI on the left, you see two columns of regions and two rows of regions. In the Mobile UI, all the regions are stacked and you need to scroll down to see all the regions.

What Is Responsive Web Design?

Tablet UI

The screenshot shows a tablet displaying the 'Order Management Application'. The top navigation bar includes 'apex' and 'Log Out' links, and tabs for 'Home', 'Customers', and 'Orders'. A 'Customers' button is highlighted. Below the navigation is a form titled 'Edit CUSTOMERS' with fields for 'Cust First Name' (Orson), 'Cust Last Name' (Koirala), 'Street Address' (810 Race St), 'Postal Code' (19107), 'City' (Philadelphia), 'State Province' (PA), and 'Country Id' (US). Buttons for 'Cancel', 'Delete', and 'Save' are at the top right. The footer contains standard application controls like 'Home', 'Edit Page 3', 'Session', 'View Debug', 'Debug', 'Show Grid', 'Quick Edit', and a phone number (+1 215 123 4738).

Mobile UI

The screenshot shows a mobile device displaying the same 'Order Management Application' as the tablet version. The layout is more compact, reflecting the smaller screen size. The 'Edit CUSTOMERS' form fields are displayed in a single column, and the overall interface is simplified for mobile use.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The example in the slide shows a form page. In the Tablet UI, you see that the buttons are side by side and the fields are of variable lengths.

Responsive Design Versus Mobile

Responsive

- Develop a single application, one source for application logic, one set of pages
- Similar UI
- Full functionality of application

jQuery Mobile

- Framework designed for mobile devices
- Lightweight and faster loading
- Native look and feel
- Uses wizards to create page components (more comfortable for APEX developers)
- Easily integrated in PhoneGap to develop Hybrid-Native applications



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The main difference between the two approaches is that responsive design allows for using a single web page with any device, while developers building mobile applications with jQuery Mobile should do so in addition to the desktop-oriented pages. Having only a single page to maintain using responsive design techniques might sound more appealing than maintaining two separate pages. The upfront cost of designing a truly responsive page are much higher and require a much higher level of understanding of Grid Layout, HTML, and CSS. So for most Application Express developers, it would probably be easier to simply rely on the built-in wizards to produce desktop pages and separate mobile pages. Another critical difference is that jQuery Mobile-based pages are generally very small and load a minimal amount of static content. Alternatively, responsive sites always load the full page, including content that might only be shown in desktop browsers. Using CSS sprites, minified JavaScript code and other techniques may help reduce the page size. However, content such as larger headers, sidebars, rich page content, and so on would still have to be loaded, only to be selectively hidden on the client later.

The main deciding factor should be the type of site or application that is being developed and the content that is being presented. A marketing site for instance, such as a company's home page, catalogs, libraries, and wikis, are well suited for responsive design. Productivity applications, such as customer management applications, business intelligence tools, and inventory applications are good candidates for a jQuery Mobile-based user interface.

These types of applications require quick response time, easy-to-use data entry forms, and would benefit from utilizing the device's native controls. Customers who are used to quickly flipping through data using touch controls in native applications will get much the same experience with their jQuery Mobile applications, such as paginating through content using swipe, orientation change to make better use of space, and context-sensitive soft keypads. Ideally of course, these types of applications would still have their desktop-orientated pages be somewhat responsive so that they work well on desktops as well as tablets.

Responsive Design in Oracle APEX

Theme 25

- Support for desktop, tablet, and mobile screen sizes
- Uses flexible Grid Layout that can be used on screen sizes from the largest monitors to mobile devices
- Interactive Reports are displayed appropriately for smaller screens using a scroll bar
- Form labels shift above and item width is standardized when using mobile devices



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Oracle APEX handles responsive design through Theme 25 in release 4.2. Theme 25 has support for all screen sizes and uses a flexible Grid Layout that adjusts based on the device you are using.

Theme 25: CSS3 Media Queries

- Styles are defined based on conditions, such as screen size or resolution.
 - Relies on browser width to determine layout/device type.
- ```
@media screen and (min-width: 320px) and (max-width: 479px)
{ ... }
```
- Multiple CSS3 media queries are defined to target “cut-off points” and appropriately adjust UI for a given screen size.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Theme 25 is based on CSS3 media queries, which allow you to define various styles based on screen size and orientation. If on a mobile phone, you want the look and feel to be different from what it would be on a tablet. So, Theme 25 uses different CSS3 media queries for each device.

### Mobile Portrait

- @media screen and (min-width: 320px) and (max-width: 479px)

### Mobile Landscape

- @media only screen and (min-width: 480px) and (max-width: 767px)

### Mobile Portrait/Landscape

- @media only screen and (max-width: 767px)

### Tablet Portrait

- @media only screen and (min-width: 768px) and (max-width: 959px)

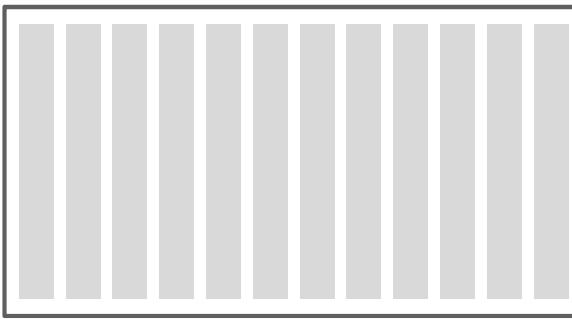
### Tablet Landscape

- @media only screen and (min-width: 960px) and (max-width: 1024px)

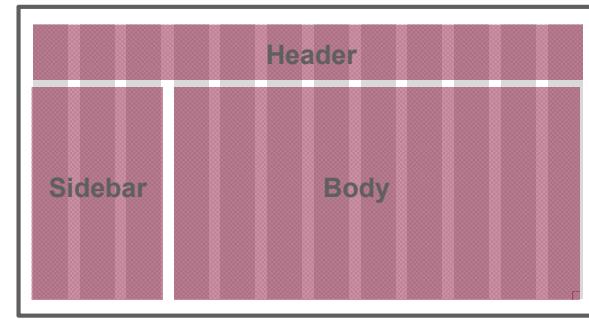
## Grid-Based Layout

- Provides a HTML structure to organize page components onto a grid
- Using a grid makes it easier to align and layout page components.

Page with 12-column Grid Layout



Page components positioned on grid

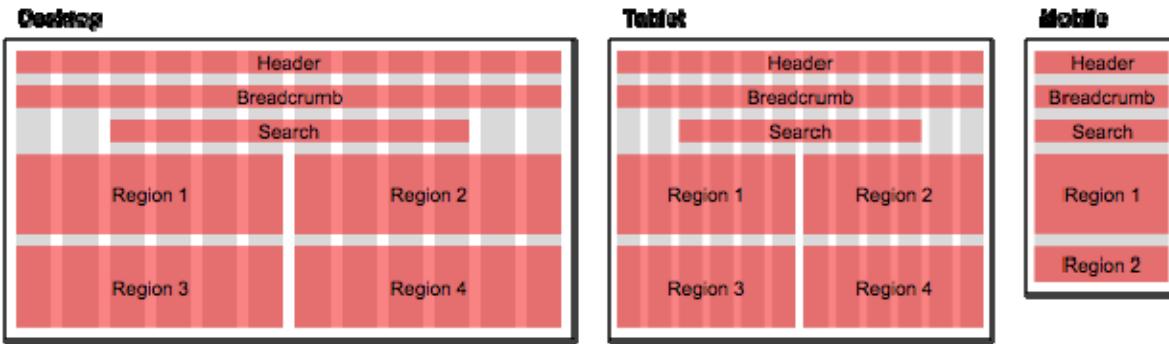


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Themes 21-26 are based on a Grid Layout. Each page has 12 columns. The page components can be positioned on the grid. For example, the sidebar can take up a certain number of columns, such as 3 in the example in the slide, and the body would take up a certain number of columns, in this case, 9.

## Grid-Based Layout

In Theme 25, CSS3 media queries can then easily shift or reposition these components.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

As a result of the Grid Layout, in Theme 25, the CSS3 media queries can shift or reposition themselves based on the components you define. In the example in the slide, you can reposition the regions on the page differently based on the screen size and orientation of the device. So, the regions are side by side for Desktop and Tablet devices, and stacked on a mobile device.

## Migrating an Application to Theme 25

1. Create and Switch to Theme 25 (Review template compatibility)
2. Revise Report Regions: Classic and Interactive
3. Revise Buttons: Large/Icons
4. Revise Forms: Region Template and Page Items
5. Review Grid Layout
6. Change Page Template on Page
7. Use Class Utilities



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The steps to migrate an application to Theme 25 are outlined in the slide. In this section, you examine some of the tasks you could perform to optimize the capabilities of the new theme.

# Creating Theme 25

The screenshot shows the Oracle Application Express User Interface. In the top left, there's a sidebar titled "User Interface" with icons for "User Interface Attributes", "Themes", "Templates", "Lists of Values", "Shortcuts", "Plug-ins", and "Component Settings". Below this is the main content area. At the top of the content area, there are tabs for "Themes", "Reports", and "History", with "Themes" being the active tab. There's also a search bar with a magnifying glass icon, a "Go" button, a "Reset" button, a "Switch Theme" button, and a "Create >" button. Below the search bar, there are two theme cards: "Scarlet - 21" and "Blue Responsive - 25 \*". The card for "Blue Responsive - 25 \*" has a small asterisk next to it. At the bottom right of the content area, there's a page number "1 - 2".

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

To migrate your application to Theme 25, you need to add it to your application. This is done through Shared Components.

# Switching to Theme 25: Template Conversion Compatibility

| Template Type | From Template Class                              | To Template Class                     | Status                                                   |
|---------------|--------------------------------------------------|---------------------------------------|----------------------------------------------------------|
| Breadcrumb    | Breadcrumb Menu                                  | Breadcrumb Menu                       | ✓                                                        |
| Button        | Button                                           | Button                                | ✓                                                        |
| Label         | Required with help                               | Required (Horizontal - Right Aligned) | Warning: no template found with matching template class. |
|               | Optional with help                               | Optional (Horizontal - Right Aligned) | Warning: no template found with matching template class. |
| List          | Vertical Unordered List with Bullets             | Vertical Unordered List with Bullets  | Warning: multiple matching templates found               |
| Page          | Printer Friendly                                 | Printer Friendly                      | ✓                                                        |
|               | No Tabs - Right Sidebar (optional / table-based) | No Tabs - Right Sidebar               | Warning: multiple matching templates found               |
|               | Login                                            | Login                                 | ✓                                                        |
| Region        | Report Filter - Single Row                       | Standard Region                       | Warning: no template found with matching template class  |
|               | Form Region                                      | Form Region                           | ✓                                                        |
|               | Region without Buttons and Titles                | Region without Buttons and Titles     | ✓                                                        |
|               | Sidebar Region                                   | Sidebar Region                        | ✓                                                        |
|               | Reports Region                                   | Standard Region                       | Warning: multiple matching templates found               |
|               | Breadcrumb Region                                | Breadcrumb Region                     | ✓                                                        |
|               | Navigation Region                                | Standard Region                       | Warning: no template found with matching template class  |
|               | Hide and Show Region                             | Hide and Show Region (Expanded)       | Warning: multiple matching templates found               |
|               | Report                                           | Standard                              | ✓                                                        |



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

When Theme 25 is added to your application, you need to switch to this theme to make it the current theme. When you switch to a new theme, a Template conversion capability report will appear. **You should review this page thoroughly because the decisions you make on this page will save you a lot of time later when you modify the resulting application.** Before you go through the switch, it is suggested that you make a copy of your application in case you need to go back and perform the process again.

In the example in the slide, the following choices were made:

- Label > Required with Help > Required Horizontal – Right Aligned
- Label > Optional with Help > Optional Horizontal – Right Aligned
- Page > No Tabs Right Sidebar (optional/table-based) > No Tabs – Right Sidebar
- Region > Report Filter – Single Row > Standard Region
- Region > Reports Region > Standard Region
- Region > Navigation Region > Standard Region

**Note:** If there is no corresponding Region Template, select Standard Region.

# Page Template Changes

The screenshot shows the Oracle Application Express Page 6: Products page template editor. On the left, the page structure tree shows 'Page 6: Products' selected under 'Regions > Content Body > Products'. The main area displays a table of products with columns: Product ID, Description, Category ID, Weight Class, and List Price. A red arrow points from the page structure tree to the 'Page' tab of the right-hand configuration panel. The 'Page' tab shows the current template is 'One Level Tabs - Right Side'. Other tabs include 'Normal' and 'Two Level Tabs - Left Side'. The 'Template Options' section includes 'Use Template Defaults' and 'CSS Classes'. The 'Media Type' dropdown is set to 'Standard Tab Set' (TS1). The 'Navigation' section includes 'Cursor Focus' (set to 'Do not focus cursor') and 'JavaScript' (disabled). The bottom of the editor shows various toolbar icons and the Oracle logo.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The default Page Template for Theme 25 is One Level Tabs – No Sidebar. Note that in the slide, you have a region in Position 3. This region will not display by default. You need to change the page template to One Level Tabs – Right Sidebar.

# Revising Report Regions: Classic Report

The screenshot shows two panels side-by-side. The left panel is titled 'Region' and contains fields for 'Sequence' (set to 10), 'Parent Region' (set to '- Select -'), and 'Position' (set to 'Content Body'). Under 'Appearance', the 'Template' dropdown is set to 'Standard Region - No Paddi' and is highlighted with a red box. The right panel is titled 'Attributes' and contains fields for 'Number of Rows Type' (set to 'Static Value') and 'Number of Rows' (set to 15). Under 'Appearance', the 'Template' dropdown is set to 'Standard - Alternative' and is also highlighted with a red box. Both panels have a toolbar at the top with icons for copy, paste, and other operations.

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.



It is recommended that for Classic Reports, you select the Region Template as Standard Region – No Padding and the Report Template as Standard – Alternative.

# Revising Report Regions: Interactive Report

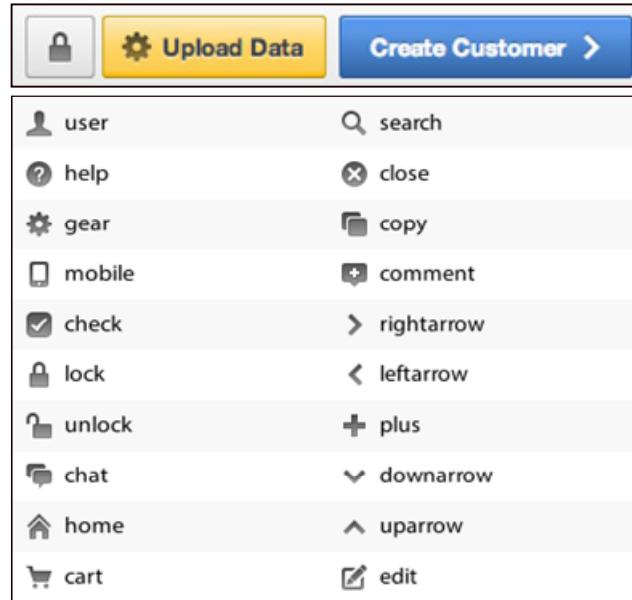
The screenshot shows the Oracle Application Express 'Region' configuration page. On the left, there's a sidebar with various settings like 'Page Items to Submit', 'Layout Sequence', 'Parent Region', 'Position', and 'Appearance'. Under 'Appearance', the 'Template' section is expanded, showing 'Interactive Report Region' selected. This section is highlighted with a red box. Below it are 'Template Options' (with 'Use Template Defaults' checked) and 'CSS Classes'. The main area displays a grid of customer data with columns: Cust First Name, Cust Last Name, Street Address, Postal Code, City, State Province, and Country. At the bottom of the grid, there are edit icons for each row. The bottom navigation bar includes links for Home, Application 108, Edit Page 2, Session, View Debug, Debug, Show Grid, Quick Edit, and a help icon.

| Cust First Name | Cust Last Name | Street Address | Postal Code          | City         | State Province | Country |
|-----------------|----------------|----------------|----------------------|--------------|----------------|---------|
| John            | Kolala         | 810 Race St    | 19107                | Philadelphia | PA             | US      |
| 217             | Bryan          | Huston         | 4901 Locust Ln       | 17109        | Harrisburg     | PA      |
| 218             | Bryan          | Dvrrie         | 3376 Perrysville Ave | 15214        | Pittsburgh     | PA      |
| 219             | Ajay           | Sen            | 220 Penn Ave # 300   | 18503        | Scranton       | PA      |
| 224             | Clara          | Krige          | 101 E Olney          | 19120        | Philadelphia   | PA      |

For Interactive Regions, select the Region Template Interactive Report Region.

## Theme 25: Icons

- Developers can easily create buttons with icons without creating custom templates.
- Simply select an icon-compatible button template and specify an icon class.



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are many icons that you can take advantage of as outlined in the slide. The name of the button is next to the icon. For example, if you wanted to use the gear icon, you specify gear for the button class. An example is in the next slide.

## Revising Buttons

The diagram illustrates the Oracle APEX button configuration process and its resulting application output. It consists of two main parts: the configuration interface and the application results.

**Configuration Interface:**

- The top part shows the 'Button' configuration page. It includes settings for Region (CUSTOMERS), Button Position (Right of Interactive Report Search B), Appearance (Hot status set to 'Yes'), and various CSS classes like 'rightarrow' and 'Icon CSS Classes'.
- The bottom part shows the same configuration page with the 'Large Button - Icon' template selected and the 'rightarrow' CSS class applied to the button.

**Application Output:**

- The top application output shows a table of customer data with a standard 'Create' button.
- The bottom application output shows the same table of customer data, but the 'Create' button now features a right-pointing arrow icon.

**ORACLE**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The Create button uses an icon. The definition for the Create button is highlighted in the slide. You specify the **Large Button – Icon** for the Button Template and enter **rightarrow** as the Button CSS Classes. Notice that the button is highlighted in blue, which indicates that this button has the type Hot. It is recommended that you only have one hot button on the page at a time. It is specified to indicate the most common choice based on the page you are on.

## Theme 25: Responsive Classes

Easily hide/show content depending on device type.

| Class            | Phones<br>767px and below | Tablets<br>979px to 768px | Desktops<br>Default |
|------------------|---------------------------|---------------------------|---------------------|
| .visible-phone   | Visible                   | Hidden                    | Hidden              |
| .visible-tablet  | Hidden                    | Visible                   | Hidden              |
| .visible-desktop | Hidden                    | Hidden                    | Visible             |
| .hidden-phone    | Hidden                    | Visible                   | Visible             |
| .hidden-tablet   | Visible                   | Hidden                    | Visible             |
| .hidden-desktop  | Visible                   | Visible                   | Hidden              |

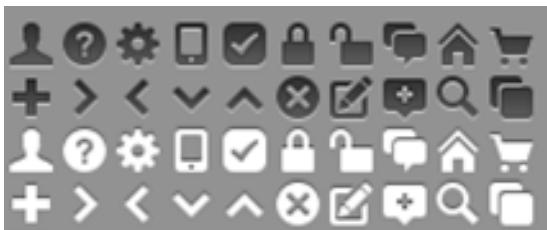


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

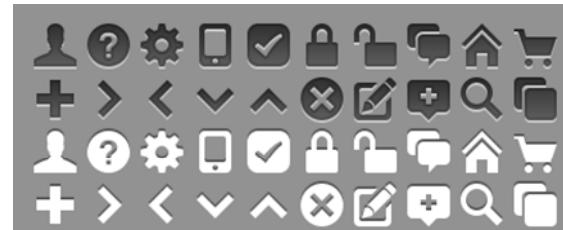
With Theme 25, you can hide and show various application components on different devices using a class. The table in the slide indicates the classes available. For example, you can specify that a region be only visible on a tablet but not on a phone or Desktop using the .visible-tablet class.

## Theme 25: High Resolution Display Support

- Support for “Retina Display” devices
- Graphics utilized by the theme are automatically upscaled for devices that use ultra high resolution displays to render web content.



Standard Display



High Resolution Display

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

On devices that support high resolution display, such as the latest iPad, iPhone, and other Android devices, you see that the icons that are displayed are slightly brighter. When using Theme 25, this capability is automatically detected and no changes to the application need to be made.

# Explanation of Grid Layout Definition

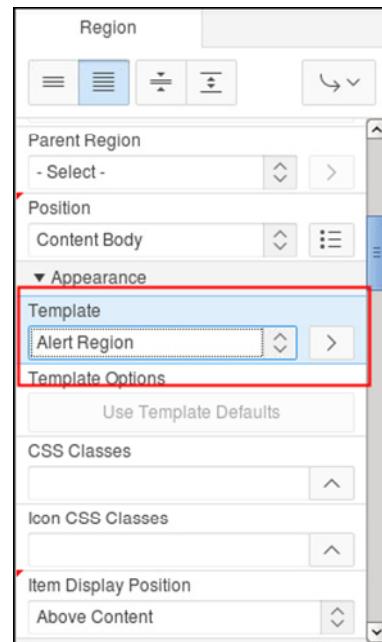
| Field         | Value     | Description                                                                                                                                                     |
|---------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Start New Row | Yes       | Position the region in a new row.                                                                                                                               |
| Column        | Automatic | Position the region in a specific grid column. Selecting Automatic uses the next available grid column in the region.                                           |
|               | 1-12      | Position the region at this specific grid column.                                                                                                               |
| New Column    | Yes       | Position this region in the next column in the same row.                                                                                                        |
|               | New       | Position this region in the same column as the previous region.                                                                                                 |
| Column Span   | Automatic | Determines how many grid columns should be used by the region. Selecting Automatic balances the available grid columns within all regions in the same grid row. |
|               | 1-12      | Use this specific number of grid columns for this region.                                                                                                       |



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

The table in the slide explains what the different options are in the Grid Layout. This is applicable in themes 21-26.

## Revising Forms: Region Template



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

For Form Pages, it is recommended that you use an Alert Region.

## Revising Forms: Regions

The screenshot shows the Oracle Application Express Order Management Application. On the left, the 'Edit CUSTOMERS' form is displayed with fields for Cust First Name (Orson), Cust Last Name (Koirala), Street Address (810 Race St), Postal Code (19107), City (Philadelphia), State Province (PA), Country Id (US), Phone Number (+1 215 123 4738), and Nls Language (us). On the right, the 'Grid' configuration panel is open, showing settings for 'Start New Row' (Yes), 'Column' (Automatic), 'Column Span' (Automatic), and 'Advanced' options like 'Static ID' and 'Custom Attributes'.

By default, the Grid Layout for your region is set to Start a New Row and automatically adjust the columns and column span.

## Revising Forms: Regions

The screenshot shows the 'Edit CUSTOMERS' page of the Order Management Application. The page has a blue header bar with tabs for Home, Customers, Orders, and Products. Below the header is a breadcrumb trail: 'Customers > Customers'. The main area contains a form with the following fields:

|                 |                         |
|-----------------|-------------------------|
| Cust First Name | Orson                   |
| Cust Last Name  | Koirala                 |
| Street Address  | 810 Race St             |
| Postal Code     | 19107                   |
| City            | Philadelphia            |
| State Province  | PA                      |
| Country Id      | US                      |
| Phone Number    | +1 215 123 4738         |
| Nls Language    | us                      |
| Nls Territory   | AMERICA                 |
| Credit Limit    | 1900                    |
| Cust Email      | Orson.Koirala@PIPIT.COM |

A context menu is open over the 'Street Address' and 'Postal Code' fields, showing options for 'Start New Row', 'Column', and 'Column Span'. The 'Column Span' option is set to 10.

At the bottom left is the Oracle logo, and at the bottom right is the copyright notice: 'Copyright © 2015, Oracle and/or its affiliates. All rights reserved.'

You can change the column and column span. In the example in the slide, you change the column to 2 and the column span to 10. Notice the difference when the page is run. It starts in column 2 and then spans 10 columns.

## Revising Forms: Page Items

The screenshot shows the 'Edit CUSTOMERS' page of the Order Management Application. It features two main input fields: 'Cust First Name' containing 'Orson' and 'Cust Last Name' containing 'Koirala'. Below these are several form items with associated grid layout settings. For 'Cust First Name', the grid layout is set to start a new row ('Start New Row: Yes') and have one column ('Column: 1'). For 'Cust Last Name', the grid layout is set to not start a new row ('Start New Row: No'), have five columns ('Column: 5'), and span three columns ('Column Span: 3'). Arrows point to the 'Start New Row' and 'Column' fields for both names.

Change the Grid Layout for page items (as required) in your form region. You can also delete any Stop and Start page items as they are not needed in Theme 25 and will be ignored. In the example in the slide, you want to place the first and last names next to each other. To do this, you must change the Grid Layout options. For First Name, change the Column to 1 and Column Span to 2. For Last Name, change the Start New Row to No, and Column to 5 and Column span to 3. In addition, change the Label Template for Last Name to Hidden Label (Read by Screen Reader). The ? icon indicates there is help for the Page item. To remove the icon, you need to delete the Help text.

**Note:** In other themes, you can also remove them by changing Start New Grid = Yes on the item below where the Stop and Start page items were. You can no longer create Stop and Start Page Item types, but the legacy items will still be there.

## Helpful Hints

- Use Automatic for Grid Layout most of the time.
- Use Alert region for small forms.
- Minimize number of columns on a report.
- Minimize use of Popup LOVs and Shuttles.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

There are a few things to keep in mind when migrating to Theme 25:

- Use automatic Grid Layout whenever possible so that the page adjusts to the size of the screen appropriately.
- Use the Alert region template for small forms. This is because this template adjusts the buttons and the text field sizes when on different devices.
- Minimize the number of columns on a report because when displaying on a mobile device, you will receive a horizontal scroll bar to see all the columns because of the screen size.
- Minimize the use of Popup LOVs and Shuttles whenever possible and use Select Lists. If you use Popup LOVs, another window opens, you select what you want, and then you are returned to the originating page. When using select lists on a mobile device, it uses the select list widget.

## Quiz



Theme 25 uses CSS3 Media queries to handle how your application will look on different devices and orientations.

- a.** True
- b.** False

**ORACLE®**

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

## Summary

In this lesson, you should have learned how to:

- Explain what Responsive Web Design is
- Create and switch to Theme 25
- Modify application components to take advantage of different responsive options



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you should have learned about what a Responsive Web Design is and how to create and switch to Theme 25. You should have also learned about modifying application components to take advantage of different responsive options.



# Making Your Application Accessible

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

## Objectives

After completing this lesson, you should be able to:

- Describe what Web Accessibility means
- Identify issues in applications that have accessibility issues
- Identify changes to your application that will improve its web accessibility



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you learn about Web Accessibility and identify the changes that you can make to your application that will improve its web accessibility.

# What Is Web Accessibility?

- Provides equal access to users with disabilities
- Includes four principles:
  - Perceivable
  - Operable
  - Understandable
  - Robust



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Web Accessibility provides equal access to users with disabilities. The four principles of web accessibility from the Introduction to Understand WCAG 2.0 reference are:

- **Perceivable:** Information and user interface components must be presentable to users in ways they can perceive. This means that the user must be able to perceive the information being presented (it cannot be invisible to all of their senses).
- **Operable:** User interface components and navigation must be operable. This means that users must be able to operate the interface (the interface cannot require interaction that a user cannot perform).
- **Understandable:** Information and the operation of user interface must be understandable. This means that users must be able to understand the information as well as the operation of the user interface (the content or operation cannot be beyond their understanding).
- **Robust:** Content must be interpreted reliably by a wide variety of user agents, including assistive technologies. This means that users must be able to access the content as technologies advance (as technologies and user agents evolve, the content should remain accessible).

## Why Is Web Accessibility Important?

- Up to 16% of the population is disabled
- Removes obstacles between a company and its existing or potential customers
- Allows employers to recruit from a broader pool of talent
- Improves products for everyone (Universal Design)
- New/Existing Procurement and Discrimination Laws in effect



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Web Accessibility is important due to the following reasons:

- Up to 16% of the population is disabled, either through birth, aging, illness, or the result of an accident.
- Accessible products remove obstacles between a company and its existing or potential customers.
- It allows employers to recruit from a broader pool of talent.
- The design of the product improves due to the changes made for accessibility.
- There are numerous Procurement and Discrimination Laws in effect that require that web applications adhere to certain accessibility standards.

There are two accessibility modes in Application Express:

- Screen Reader Mode
- High Contrast Mode

# Screen Readers

- What are they?
  - A software application that attempts to identify and interpret what is being displayed on the screen
  - Interpretation is presented to the user with text-to-speech, sound icons, or a Braille output device.
  - Useful for people who are blind, visually impaired, illiterate, or learning disabled
- JAWS from Freedom Scientific is the most popular screen reader.



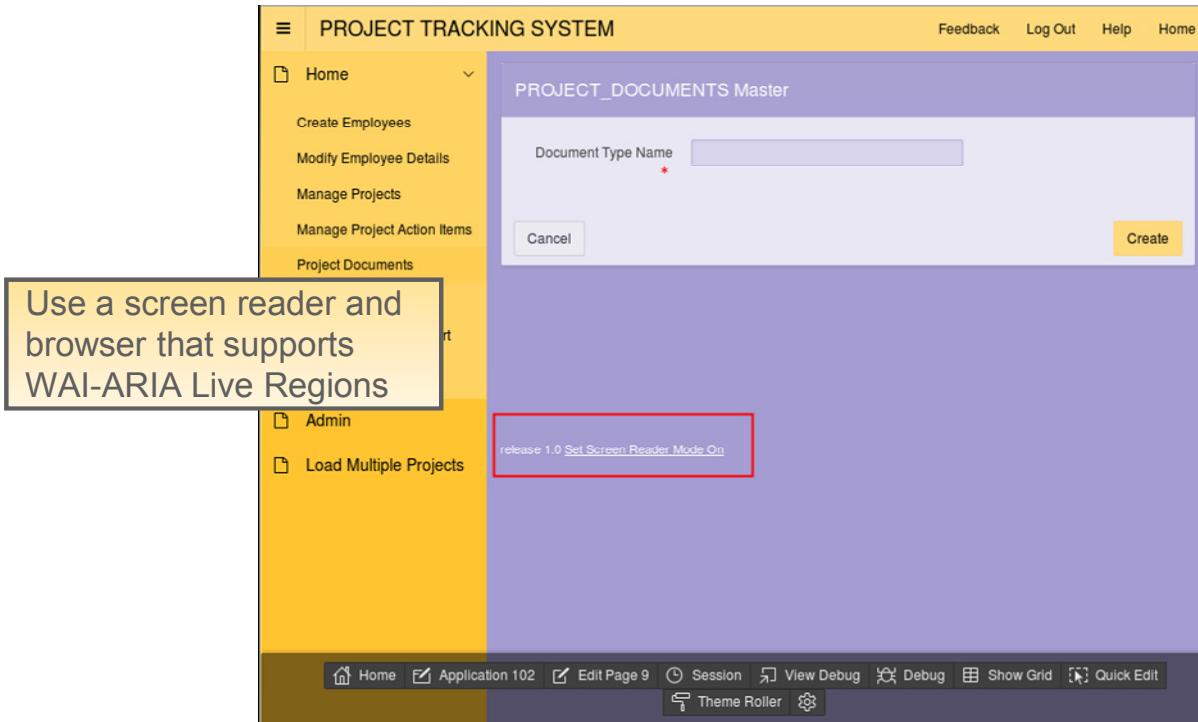
Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

A Screen Reader is a software application that attempts to identify and interpret what is being displayed on the screen. It interprets what is presented to the user in a variety of ways, such as text-to-speech, sound icons, or a Braille output device. A Screen Reader is useful for people who are blind, visually impaired, illiterate, or learning disabled.

There are many third-party screen readers in the market. Oracle Application Express has only been tested with the most recent JAWS 13.0 (by Freedom Scientific), which supports Live Regions. The current list of browsers that support this include:

- Mozilla Firefox 3.0 and later
- Microsoft Internet Explorer version 8

# Turn Screen Reader Mode On



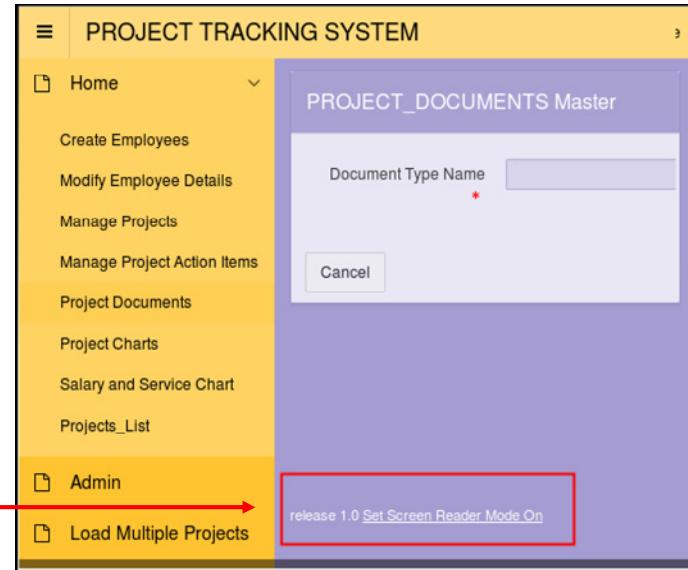
The screen reader must be turned on in your APEX Application to take advantage of the accessibility features you defined in your application. You use a screen reader and browser that supports WAI-ARIA (Web Accessibility Initiative-Accessible Rich Internet Applications) live regions.

# Themes and Templates: Page Template

## Page Template:

- #SCREEN\_READER\_TOGGLE# in Footer

```
11 </div>
12 <footer class="t-Footer">
13 #APP VERSION#
14 #CUSTOMIZE#
15 #SCREEN READER TOGGLE#
16 #REGION_POSITION_05#
17 </footer>
18</div>
19</div>
20</div>
21<div class="t-Body-inlineDialogs">
22 #REGION_POSITION_04#
23</div>
```



To turn the screen reader on for your application, the page template should have the substitution variable #SCREEN\_READER\_TOGGLE# in the footer. Note that many of the themes provided with Oracle APEX already have this parameter.

# Changing Text of Toggle Message

Add Text Message with user-defined text.

The screenshot shows two windows. The left window is titled 'Globalization' and has a sidebar with 'Globalization Attributes', 'Text Messages' (which is selected), and 'Translate Application'. The right window is titled 'Create/Edit Text Message' and contains the following fields:

- Application: 102 PROJECT TRACKING SYSTEM
- Name: SET\_SESSION\_SCREEN\_READER\_ON
- Language: English (en)
- Used in JavaScript: No
- Text (Example: Tax: %0 Total amount %1): Screen Reader Mode Off (Enable)

At the bottom are 'Cancel', 'Create And Create Another', and a blue 'Create Text Message' button.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can change the message that appears at the bottom of the window to toggle the screen reader on and off by adding a Text Message for the name of the parameter. Perform the following steps:

1. From Shared Components, under Globalization, click Text Messages.
2. Enter SET\_SESSION\_SCREEN\_READER\_ON for Name, enter the text you want to display in the Text area, and click Create Text Message.

When you run the page now, you will receive the new message.

# Themes and Templates: Region Template

## Region Template:

- <hn> tags
- class="visuallyhidden"
- role="main"

```
Template ?
1 <div id="#REGION_STATIC_ID#" #REGION_ATTRIBUTES# class="t-IRR-region #REGION_
2 role="main" aria-labelledby="#REGION_STATIC_ID# heading">
3 <h2 class="u-VisuallyHidden" id="#REGION_STATIC_ID#_heading">#TITLE#</h2>
4 #PREVIOUS##BODY##SUB_REGIONS##NEXT#
</div>
```

	Employee Id	First Name	Last Name	Email	Phone Number	Mobile Number	Address
531	Deepthi	Rao		DSR@ORACLE.COM	7654567898	5546897432	Colorado, Texas
530	Nag	Roberts		nr@oracle.com	7865431235	7865687808	New York, US
505	Fiorello	LaGuardia		fiorello.laguardia@pts.com	2125553923	1235342653	Hangar Center, Third Floor, Flushing, NY
504	Frank	O'Hare		frank.ohare@pts.com	7735557693	3157862405	10000 West O'Hare Chicago, IL

Title not displayed  
but read by screen reader

```
<div class="col col-12 " >
<div id="R22292197871948275" class="t-IRR-region " role="main" aria-labelledby="R22292197871948275_heading">
<h2 class="u-VisuallyHidden" id="R22292197871948275_heading">Employees Report</h2>
<div id="R22292197871948275_ir" class="a-IRR-container"><div id="R22292197871948275_worksheet_r
</stvle>
```



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the Region Template, you can specify the <h2> tag with the #TITLE# substitution but then specify that it uses the visuallyhidden class, which will be used by the screen reader. In addition, you can explicitly specify the ARIA Landmark support for role="main" in the template, which indicates that this region is the main region on the page.

# Themes and Templates

## Implicit Landmark Support for HTML5 tags in Page Template

Footer Landmark  
implicit from  
<footer> tag



```
<div class="t-Body">
 #SIDE GLOBAL NAVIGATION LIST#
 <div class="t-Body-main">
 <div class="t-Body-title" id="t_Body_title">
 #REGION_POSITION_01#
 </div>
 <div class="t-Body-content" id="t_Body_content">
 #SUCCESS_MESSAGE##NOTIFICATION_MESSAGE##GLOBAL_NOTIFICATION#
 <div class="t-Body-contentInner">
 #BODY#
 </div>
 <div class="t-Footer">
 #APP VERSION#
 #CUSTOMIZE#
 #SCREEN READER TOGGLE#
 #REGION_POSITION_05#
 </div>
 </div>
 <div class="t-Body-inlineDialogs">
 #REGION_POSITION_04#
 </div>
```

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the Page Template, the <footer> HTML5 tag in the footer is used, which implicitly indicates the footer role or landmark on the page.

# Interactive Reports

## Summary on <table> tag

The screenshot shows two versions of an interactive report table side-by-side. The top version is labeled "Screen reader set to off" and the bottom version is labeled "Screen reader set to on". Both versions display the same data from an Employee table:

	Employee Id	First Name	Last Name	Email
	516	Ward		
	517	Martin	Johnson	martin.johnson@pts.com
	519	Adams	Henry	adams.henry@pts.com
	521	Miller	Emanuel	miller.emmanuel@pts.com
	522	Kiranmayil	Adapala	ka@oracle.com
	523	Sowmya	Kiran	sk@oracle.com

Both versions have identical HTML code for the table structure, indicating that the summary information is present in both cases.

With the screen reader set to off, you do not receive the summary information about the report in the HTML. With the screen reader set to on, you receive summary information about the report in the table tag. Note that the reason the screen reader mode is not turned on by default is that it is quite expensive to calculate the statistics regarding the total number of rows in the summary if it is not needed.

# Interactive Reports

## Alt tag on Link Column

Link Column

Link Column: Link to Custom Target

Link Icon: #e2.gif alt="EDIT: #FIRST\_NAME"

Link Attributes:

Target: Page in this Application

Page: 6 Reset Pagination

Request:

Clear Cache:

Name: Value

Item 1: P6\_EMPLOYEE\_ID Value 1: #EMPLOYEE\_ID#

```
<td align="right" headers="EMPNO">

</td>
```

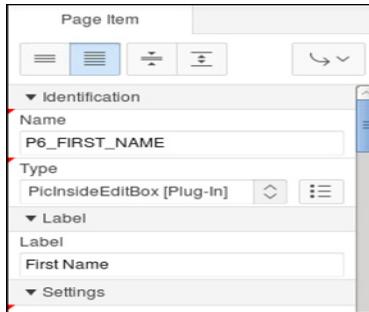


Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In the Link Text, you can include an `<alt>` tag containing the text that the screen reader can read.

# Forms

Label all Page Items.



```
<div class="col col-12 ">
 <div id="P6_FIRST_NAME_CONTAINER" classe="t-Form-fieldContainer rel-col ">
 <div class="t-Form-labelContainer col col-3">
 <label id="P6_FIRST_NAME_LABEL" classe="t-Form-label" for="P6_FIRST_NAME">
 First Name

 </label>
 </div>
 </div>
</div>
```

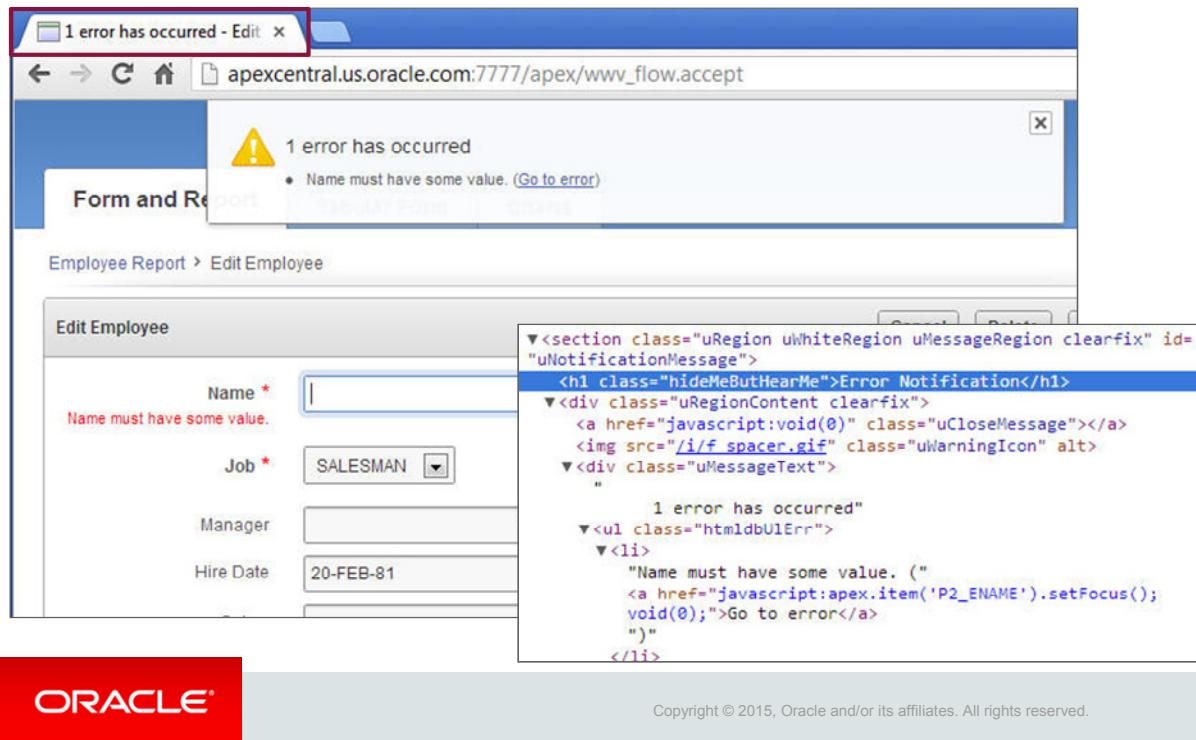
ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Make sure that all your page items have a label so that the screen reader knows how to describe the page item. If you do not want a page item to display a label such as when you want to have a search field, you can use the Hidden label, read by screen reader template.

# Forms

## Validation Errors



With the screen reader set to on, the title of the page will show the message of the error so that the screen reader can relay information to the user that there is an issue right away. In addition, a `<h1>` is inserted with the error to indicate an Error Notification. The `<h1>` tag is getting set from what is contained in the Subtemplate–Notification area in the Page Template.

# Tabular Forms

## Use as Row Header

Column Definition

Column Name	<b>FIRST_NAME</b>
Column Heading	First Name
Show Column	Yes
Heading Alignment	center
Compute Sum	No
Column Alignment	left
Sortable Column	Yes
Column Width	
Use As Row Header	<input checked="" type="checkbox"/> Yes
Include In Export	Yes

```
▼<td headers="ENAME">
 <label for="f03_0001" class="hideMeButHearMe">Name: SMITH</label>
 <input type="text" name="f03" size="16" maxlength="2000" value="SMITH" id="f03_0001" autocomplete="off">
</td>
```

When option set to Yes

```
▼<td headers="ENAME">
 <label for="f03_0001" class="hideMeButHearMe">Name</label>
 <input type="text" name="f03" size="16" maxlength="2000" value="SMITH" id="f03_0001" autocomplete="off">
</td>
```

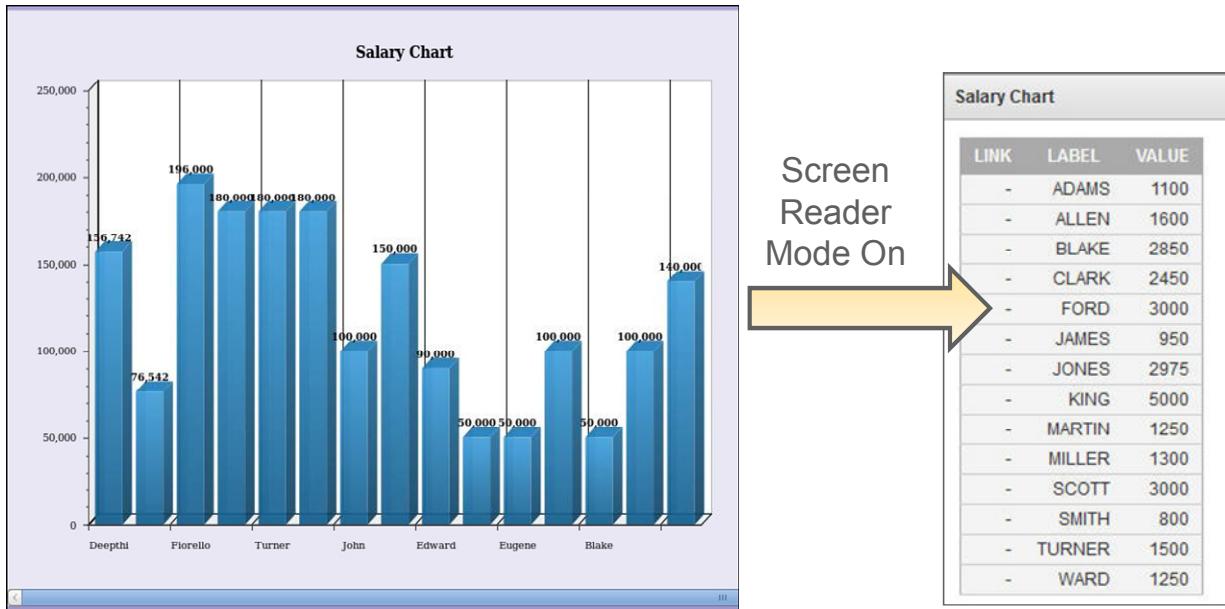
When option set to No



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

With the screen reader set to on, the label for each row in the table will identify the row header and the column value. In this example, the row header is Ename and the value is Smith.

## Charts



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Flash charts are not currently accessible to screen readers. Therefore, when running in screen reader mode, the user will get a report representation of the information conveyed in the chart. A separate report will be generated for each series of a multiple-series chart. When running in screen reader mode, these data tables contain descriptive text.

## High Contrast

To provision High Contrast Mode for users of your own database applications, use:

- Page template #HIGH\_CONTRAST\_TOGGLE# substitution string
- APEX\_UTIL APIs
- f?p syntax REQUEST attribute



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

You can identify sessions as optimized for high contrast in the Oracle Application Express development environment, Websheet run time, and also within your own database applications. Enabling high contrast mode results in the controls in an interactive report region's search bar display without icons, and all controls, including the icons are shown as clear text using a button-like look. There are also numerous reports throughout Application Express that share the same report template, which have contrast issues in the column headers. If high contrast mode is enabled, for those reports, the gradient background is removed and a darker, solid-color background is displayed instead.

- **Page template #HIGH\_CONTRAST\_TOGGLE# substitution string:** Add this substitution string to your page template and Oracle Application Express displays a link to the current page to turn on or turn off (or toggle) the mode. In other words, if you are in standard mode, this procedure generates a link to turn it on.
- **APEX\_UTIL APIs:** There are also APIs that can be used to control this mode. You may want to use the APIs if you only want to render the toggle in one place and do not want to do this at page template level, or if you want more control over the actual displayed link text.
- **f?p syntax REQUEST attribute:** Use this attribute to enable and disable high contrast mode. The general syntax for the f?p syntax REQUEST attribute is:

f?p=application:page:session:request:.... If the request is exactly SET\_SESSION\_HIGH\_CONTRAST\_ON or SET\_SESSION\_HIGH\_CONTRAST\_OFF, then the session is put into or out of high contrast mode. For example:

```
Set High Contrast On
```

## Quiz



There is only one way to make an application accessible.

- a. True
- b. False

ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

**Answer: b**

## Summary

In this lesson, you should have learned how to:

- Describe what Web Accessibility means
- Identify issues in applications that have accessibility issues
- Identify changes to your application that will improve its web accessibility



ORACLE®

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

In this lesson, you should have learned about web accessibility, the various accessibility issues, and also the changes that you can make in your application to improve web accessibility.