ORACLE University

Integrated Cloud Applications & Platform Services

Oracle Application Express Workshop I

Student Guide – Volume II

D79653GC30 | D106872

ORACLE

Authors

Sayantani Ghosh
Moitreyee Hazarika

Technical Contributors and Reviewers

Suresh Rajan
Anjani Pothula
David Peake
Chaitanya Koratamaddi
Joe Greenwald

Editors

Adrita Biswas
Aju Kumar
Moushmi Mukherjee

Graphic Editors

Prakash Dharmalingam
Anne Elizabeth
Pushparaj Kundar

Publishers

Pavithran Adka
Veena Narasimhan
Giri Venugopal

3009092019

Contents

1 Course Overview

Objectives 1-2
Course Goals 1-3
Course Road Map 1-4
Course Environment 1-8
Workspace Details 1-9
Accessing the labs Directory 1-10
Meet Course Persona: Stella 1-11
Meet Course Persona: Steve 1-12
Introduction to the Demo Application: Project Tracking System 1-13
Project Tracking System: Basic Reports 1-14
Project Tracking System: Interactive Grid 1-16
Project Tracking System: Supporting Reports 1-17
Snapshot of the Project Tracking System 1-18
Introduction to the Practice Application: GlobalMart Management Tool 1-19
Additional Resources – Oracle Application Express 19.1 1-20
Summary 1-21

I Unit I Introduction: Getting Started with Application Express

Stella Expresses Her Concern: Driving Factor Behind PTS I-2
Unit 1 Road Map I-3

2 Oracle Application Express: Introduction

You Are Here in This Course 2-2
Objectives 2-3
Steve Decides to Explore Oracle Application Express 2-4
Lesson Agenda 2-5
What Is Oracle Application Express? 2-6
Why Use Oracle Application Express? 2-7
Types of Applications 2-8
Applications Developed by Using Oracle Application Express 2-9
High-Level Architecture 2-10
Types of Installations 2-12
Quiz 2-13
Lesson Agenda 2-14

What Is a Workspace? 2-15
What Is an Internal Workspace? 2-16
Defining Roles 2-17
Quiz 2-19
Lesson Agenda 2-20
Signing In to a Workspace 2-21
Creating a Developer User 2-22
Using Application Express in Dark Mode 2-23
Practice 2-1 Overview: Using Oracle Application Express as a Workspace Administrator 2-24
Workspace Home Page 2-25
What Is App Builder? The First Component of Oracle Application Express 2-26
Types of Applications 2-27
What Is SQL Workshop? The Second Component of Oracle Application Express 2-28
Accessing SQL Workshop 2-29
Running SQL Commands 2-30
Importing and Running a SQL Script 2-31
What Is Team Development? The Third Component of Oracle Application Express 2-32
What Is App Gallery? The Fourth Component of Oracle Application Express 2-33
Accessing the App Gallery 2-34
Installing a Productivity Application 2-35
Running an Installed Productivity Application 2-36
Configuring the Setup 2-37
Unlocking an Installed Productivity Application 2-38
Exporting an Application 2-39
Importing an Application 2-40
Sample Datasets 2-42
Installing a Sample Dataset 2-43
Quiz 2-45
Practice 2-2 Overview: Using Oracle Application Express as a Developer 2-47
Summary 2-48

3 Creating a Database Application
You Are Here in This Course 3-2
Objectives 3-3
Steve Thinks of Developing PTS for Different Interfaces 3-4
Lesson Agenda 3-5
Accessing App Builder 3-6
App Builder Home Page 3-7

Lesson Agenda 3-8
Components of a Database Application 3-9
Database Application User Interfaces 3-10
Database Application Home Page 3-11
Page Designer: The World's Most Advanced IDE 3-12
Page Designer: Drag and Drop from Gallery 3-13
Page Definition: Overview 3-14
Different Views of a Page 3-15
Switching Between Pages and View Types 3-16
Quiz 3-17
Lesson Agenda 3-19
Create Application Wizard 3-20
Accessing the Create Application Wizard 3-21
Different Ways of Creating a Database Application 3-22
Creating Database Application 3-23
Project Tracking System Home Page 3-24
Creating a Database Application from a File 3-25
Running an Application 3-26
Using the Developer Toolbar 3-27
Practice 3 Overview: Creating Database Applications 3-28
Summary 3-29

4 Working with Reports
You Are Here in This Course 4-2
Objectives 4-3
Steve Designs Reports for Tracking Projects 4-4
Lesson Agenda 4-5
Types of Reports 4-6
Different Ways of Creating a Report 4-7
Quiz 4-8
Lesson Agenda 4-9
Selecting the Appropriate Report Type 4-10
When Is Classic Report the Best Option? 4-11
Creating Classic Report with Form 4-12
Creating Classic Reports with Form 4-13
Classic Report 4-14
Practice 4-1 Overview: Creating and Modifying a Classic Report 4-15
List View Reports 4-16
Creating a List View Report 4-17
Modifying a List View Report 4-19
Practice 4-2 Overview: Creating and Modifying a List View Report 4-20

Column Toggle Reports 4-21
Creating a Column Toggle Report 4-22
Practice 4-3 Overview: Creating a Column Toggle Report 4-24
Quiz 4-25
Summary 4-26

5 Working with Interactive Reports
You Are Here in This Course 5-2
Objectives 5-3
When Is an Interactive Report the Best Option? 5-4
Lesson Agenda 5-5
How to Create an Interactive Report 5-6
Creating an Interactive Report 5-7
Interactive Report 5-9
Interactive Report with Link to Single Row View 5-10
Practice 5-1 Overview: Creating and Manipulating an Interactive Report 5-11
Lesson Agenda 5-12
Accessing Report Attributes 5-13
Editing Report Attributes 5-14
Customizing the Search Bar and Actions Menu 5-15
Specifying the Download Formats 5-16
Icon and Detail Views 5-17
Using the Link Column 5-18
Modifying the Interactive Report Query 5-19
Practice 5-2 Overview: Customizing an Interactive Report 5-20
Quiz 5-21
Summary 5-22

6 Working with Interactive Grids
You Are Here in This Course 6-2
Objectives 6-3
Steve Explores the Interactive Grid 6-4
Lesson Agenda 6-5
What Is an Interactive Grid? 6-6
An Interactive Grid with Form 6-7
Interactive Grid Versus Interactive Report 6-8
Lesson Agenda 6-9
Ways to Create an Interactive Grid 6-10
Creating an Interactive Grid (without Form) 6-11
Interactive Grid (without Form) 6-13
Creating an Interactive Grid (with Form) 6-14

Creating an Interactive Grid 6-15
Interactive Grid (with Form) 6-16
Key Attributes of an Interactive Grid: Region Attribute 6-18
Key Attributes of an Interactive Grid: Report Attribute 6-19
Key Attributes of an Interactive Grid: Column Attribute 6-20
Lesson Agenda 6-21
Customization: Rendering the Interactive Grid Editable 6-22
Interactive Grid: Editable 6-23
Customization: Creating Column Links 6-24
Customization: Modifying the Form 6-27
Customization: Modifying Fields on the Form 6-28
Customization: Creating a List of Values on a Column 6-29
Practice 6 Overview: Creating and Manipulating an Interactive Grid 6-31
Quiz 6-32
Summary 6-33

I Unit I Summary: Getting Started with Application Express
Unit I Road Map I-2

II Unit II Introduction: Building User-Friendly Web Applications
Steve Explores Oracle Application Express Further II-2
You Are Here in This Course II-3

7 Working with Pages and Regions
You Are Here in This Course 7-2
Objectives 7-3
Steve Works with Pages and Regions 7-4
Lesson Agenda 7-5
What Is a Page? 7-6
Accessing Page Definition 7-7
Page Modes: Normal, Modal, and Nonmodal 7-8
Page Designer Window 7-9
Page Designer UI Elements 7-10
Page Designer Toolbar 7-11
The Left Pane of Page Designer 7-12
The Central Pane of Page Designer 7-13
The Central Pane of Page Designer: Gallery 7-14
The Central Pane of Page Designer: Messages 7-15
The Central Pane of Page Designer: Page Search and Help 7-16
The Right Pane of Page Designer: Property Editor 7-17
Page Designer UI Elements: Property Editor: Code Editor 7-18

Editing Page Attributes 7-19
Running a Page 7-20
Runtime Developer Toolbar 7-21
Quiz 7-22
Lesson Agenda 7-23
Region and Region Types 7-24
Adding a Region: From the Rendering Tree 7-25
Adding a Region: From the Gallery 7-26
Positioning the Region 7-28
Example: Creating a Region Type (Region Display Selector) 7-29
Example: Creating a Region Display Selector 7-30
Conditional Display of Regions 7-32
Editing a Region 7-34
Editing a Region at Runtime (using Quick Edit option) 7-35
Editing a Region at Runtime (using Live Template Options) 7-36
Specifying a Region Header and Footer 7-37
Copying a Region 7-38
Example: Copying Regions 7-39
Creating a Subregion 7-40
Practice 7-1 Overview: Creating and Modifying Pages and Regions 7-41
Lesson Agenda 7-42
Global Page 7-43
Creating a Global Page Region 7-44
Example: Creating a Global Page Region 7-45
Practice 7-2 Overview: Working with Global Pages 7-46
Viewing Pages of the Application 7-47
Creating a Page Group 7-48
Example: Creating a Page Group 7-49
Copying a Database Application Page 7-51
Deleting a Page 7-52
Quiz 7-53
Summary 7-55

8 Managing Forms

You Are Here in This Course 8-2
Objectives 8-3
Stella Has a New Requirement: Wants a Solution 8-4
Steve Plans Data Entry Through Forms 8-5
Lesson Agenda 8-6
Forms: Introduction 8-7
Types of Forms 8-8

Lesson Agenda 8-9
Accessing the Create Form Wizards 8-10
ROWID Versus Primary Key 8-11
Creating a Form 8-12
Example: Creating a Form 8-13
Example: Adding a Form as a Region 8-16
Creating a Report with Form 8-17
Example: Creating a Report with Form 8-18
Creating a List View with Form 8-21
Example: Creating List View with Form 8-22
Creating a Stacked Master Detail Form 8-23
Example: Creating a Stacked Master Detail Form 8-24
Example: Creating a Stacked Master Detail 8-25
Quiz 8-26
Practice 8-1 Overview: Creating a Report with Form 8-28
Practice 8-2 Overview: Creating a List View with Form 8-29
Practice 8-3 Overview: Creating a Master Detail Form (Side By Side) 8-30
Lesson Agenda 8-31
Linking a Report to a Form 8-32
Example: Linking a Report to a Form 8-33
Practice 8-4 Overview: Creating a Master Detail Form (Stacked) 8-35
Practice 8-5 Overview: Creating a Master Detail Form (Drill-Down) 8-36
Using Quick Edit 8-37
Reordering Items 8-38
Changing Item Display Type 8-39
Customizing Forms 8-40
Quiz 8-41
Summary 8-42

9 Adding Items and Buttons
You Are Here in This Course 9-2
Objectives 9-3
Steve Enhances User Interface—Adds Items and Buttons to Pages 9-4
Lesson Agenda 9-5
Items 9-6
What Are Page Items? 9-7
Types of Page Items 9-8
What Are Application Items? 9-9
Lesson Agenda 9-10
Creating a Page Item 9-11
Example: Creating a Date Picker Item 9-12

Editing an Item 9-13
Examples: Editing an Item 9-14
Creating Quick Picks 9-16
Adding Subtypes 9-17
Quiz 9-18
Lesson Agenda 9-19
What Is an LOV? 9-20
Accessing the Lists of Values Page 9-21
Creating a Static LOV 9-22
Example: Creating a Static LOV 9-23
Associating the Static LOV with an Item 9-24
Creating a Dynamic LOV 9-25
Example: Creating a Dynamic LOV 9-26
Associating the Dynamic LOV with a Select List Item 9-27
Cascading LOV 9-28
Example: Creating a Cascading LOV 9-29
Quiz 9-31
Lesson Agenda 9-32
What Is a Button? 9-33
Creating a Button 9-34
Example: Creating a Button 9-35
Example: Editing Button Attributes 9-36
Modifying a Button to Redirect to a URL 9-37
Quiz 9-38
Practice 9-1 Overview: Adding Items and Buttons 9-39
Practice 9-2 Overview: Manipulating Items 9-40
Summary 9-41

10 Understanding Session State
You Are Here in This Course 10-2
Objectives 10-3
Steve Understands Session State 10-4
Lesson Agenda 10-5
What Is a Session State? 10-6
Session ID 10-7
Session Timeout 10-8
Setting Session Timeout 10-9
How Does Oracle Application Express Implement Session State? 10-10
Identifying the Parts of an Oracle Application Express URL 10-11
Quiz 10-13
Lesson Agenda 10-14

Viewing Session State 10-15
Referencing Session State 10-16
Example: Referencing Session State by Using Bind Variables 10-18
Example: Referencing Session State in Static Text 10-19
Clearing the Cache 10-20
Example: Clearing the Cache 10-21
Quiz 10-23
Practice 10-1 Overview: Understanding Session State 10-25
Summary 10-26

11 Including Page Processing
You Are Here in This Course 11-2
Objectives 11-3
Steve Includes Page Processing in the Application 11-4
Lesson Agenda 11-5
Page Rendering Versus Page Processing 11-6
Example: Page Rendering 11-7
Example: Page Processing 11-8
Types of Logic 11-10
Lesson Agenda 11-11
What Is a Computation? 11-12
Computation Examples 11-13
Creating Computations 11-14
Example: Creating Page-Processing Computations 11-15
Quiz 11-16
Lesson Agenda 11-17
What Is a Page Process? 11-18
Automatic Processing Processes 11-19
Reviewing an Automated Row Fetch Process 11-20
Reviewing an Automatic Row (DML) Processing Process 11-21
Creating an On Submit Process 11-22
Creating an On Load Process 11-23
Quiz 11-25
Lesson Agenda 11-26
What Are Validations? 11-27
Creating a Validation 11-28
Using the Create Validation Feature 11-29
Example: Creating an Item String Comparison Validation 11-30
Example: Creating a Regular Expression Validation 11-31
Example: Creating a SQL Validation 11-32
Example: Creating a PL/SQL Validation 11-33

Quiz 11-34
Lesson Agenda 11-35
What Is Branching? 11-36
Example: Creating a Branch 11-37
Practice 11-1 Overview: Creating and Manipulating Computations, Processes, and Validations 11-40
Summary 11-41

12 Using Dynamic Actions and Plug-Ins
You Are Here in This Course 12-2
Objectives 12-3
Steve Uses Dynamic Actions and Plug-ins 12-4
Lesson Agenda 12-5
Examples: Creating and Using Dynamic Actions 12-6
What Is a Dynamic Action? 12-7
Example 1: Enabling and Disabling an Item 12-8
General Steps to Create a Dynamic Action 12-9
Example 2: Setting the Value of an Item When Another Item Changes 12-11
Example 3: Clearing All Items on the Click of a Button 12-14
Example 3: Clearing All Items When a Button Is Clicked 12-15
Example 3: Clearing All Items on the Click of a Button 12-16
Example 4: Refreshing the Data in a Report Using Custom Filters 12-17
Example 4: Refreshing the Data in a Report Using Custom Filters: Overview 12-18
Example 4: Refreshing the Data in a Report Using Custom Filters 12-19
Example: Disabling a Button and Submitting the Page When Clicking a Button 12-21
Quiz 12-23
Practice 12-1 Overview: Creating and Using Dynamic Actions 12-25
Lesson Agenda 12-26
What Is a Plug-In? 12-27
Steps to Use a Plug-in in Your Application 12-28
Accessing the Plug-in Repository 12-29
Importing a Plug-In 12-31
Installing a Plug-In 12-32
Reviewing a Plug-in Definition 12-33
Using an Item Plug-in on a Page 12-35
Quiz 12-37
Additional Plug-in Examples 12-38
Adding the Modern Star Rating Item 12-39
Adding the Modern Star Rating Plug-in—Add Column 12-40
Adding the Modern Star Rating Item—Import and Install the Plug-in 12-41

Adding the Modern Star Rating Item Plug-in to a Form 12-42
Practice12-2 Overview: Importing and Using Plug-Ins 12-43
Quiz 12-44
Summary 12-45

13 Validating and Debugging Your Application
You Are Here in This Course 13-2
Objectives 13-3
Steve Validates and Debugs the Application 13-4
Lesson Agenda 13-5
Using the Advisor 13-6
Resolving Advisor Errors/Warnings 13-7
Practice 13-1 Overview: Using the Advisor 13-8
Lesson Agenda 13-9
Using User Interface Defaults 13-10
Creating User Interface Defaults for a Table 13-11
Managing Your Table Dictionary 13-12
Managing Your Attribute Dictionary 13-13
Reviewing Items and Report Columns 13-14
Modifying Attributes in the Dictionary 13-15
Quiz 13-16
Practice 13-2 Overview: Modifying the Attribute Dictionary 13-19
Lesson Agenda 13-20
What Is the Debug Option? 13-21
Enabling and Disabling Debug Mode 13-22
Debugging an Application 13-23
Viewing Debug Reports 13-24
Viewing Debug Reports (at Runtime) 13-25
Quiz 13-26
Summary 13-27

II Unit II Summary: Building User-Friendly Web Applications
Unit II Road Map II-2

III Unit III Introduction: Customizing Your Web Application
Steve Designs Application Navigation III-2
Unit III Road Map III-3

14 Adding Shared Components That Aid Navigation
You Are Here in This Course 14-2
Steve Builds a Navigation System into PTS 14-3

Objectives 14-4
Lesson Agenda 14-5
Shared Components: Examples 14-6
What Are Shared Components? 14-7
Lesson Agenda 14-8
Accessing Navigation Menu Page 14-9
Creating Navigation Menu Entries 14-10
Lesson Agenda 14-11
Lists 14-12
Accessing the Lists Page 14-13
Creating a Static List 14-14
Creating a Static List Region 14-15
Creating a Dynamic List 14-16
Creating a Dynamic List Region 14-18
Creating a List Region on Global Page 14-19
Lesson Agenda 14-20
Viewing a Breadcrumb 14-21
Creating Breadcrumb Entries 14-22
Reparenting Breadcrumbs 14-24
Lesson Agenda 14-25
Accessing the Navigation Bar Entries Page 14-26
Creating a Help Page 14-27
Creating a Navigation Bar Entry 14-28
Quiz 14-29
Practice14 Overview: Adding Shared Components That Aid Navigation 14-30
Summary 14-31

15 Working with Themes, Templates, and Files
You Are Here in This Course 15-2
Objectives 15-3
Steve Works with Application User Interface 15-4
Lesson Agenda 15-5
What Is a Theme? 15-6
Accessing the Themes Page 15-7
Creating a New Theme 15-8
Creating a Copy of an Existing Theme 15-9
Editing a Theme 15-10
Switching Between Themes 15-11
Example: Switching Between Themes 15-12
Changing the Theme Style 15-13
Using Universal Theme and Theme Roller 15-14

Accessing the Theme Roller 15-15
Customizing Your Theme Style Using Theme Roller 15-16
Customizing Your Theme Style Using Theme Roller – Changing Header 15-17
Customizing Your Theme Style Using Theme Roller – Changing Title Bar Background 15-18
Customizing Your Theme Style Using Theme Roller – Updating Navigation 15-19
Saving Theme Style in Theme Roller 15-20
Quiz 15-21
Lesson Agenda 15-22
What Are Templates? 15-23
Types of Templates 15-24
Accessing the Templates Page 15-25
Replacing a Template 15-26
Example: Replacing a Template 15-27
Creating a Copy of an Existing Template 15-28
Editing a Template 15-29
Applying a Template 15-30
Example: Applying a Template 15-31
Changing Default Templates in a Theme 15-32
Example: Changing Default Templates in a Theme 15-33
Overriding Application Defaults at the Page Level 15-34
Using Substitution Strings in Templates 15-35
Lesson Agenda 15-36
Uploading a Cascading Style Sheet 15-37
Referencing a Cascading Style Sheet 15-38
Uploading an Image 15-39
Example: Using an Uploaded Image 15-40
Quiz 15-42
Practice 15 Overview: Working with Themes, Templates, and Files 15-43
Summary 15-44

16 Implementing Security

Steve Implements Security in the Application 16-2
You Are Here in This Course 16-3
Objectives 16-4
Lesson Agenda 16-5
Securing an Application: Overview 16-6
Accessing Security Tasks 16-7
Lesson Agenda 16-8
Using Authentication 16-9
Accessing Authentication Schemes Page 16-10

Using Preconfigured Authentication Schemes 16-11
Examples: Preconfigured Authentication Schemes 16-12
Example: Creating Authentication Based on Preconfigured Schemes 16-13
Copying an Authentication Scheme 16-14
Quiz 16-15
Practice 16-1 Overview: Creating an Authentication Scheme 16-16
Lesson Agenda 16-17
Using Authorization 16-18
Where Can You Implement Authorization? 16-19
Methods to Implement Authorization 16-20
Creating an Authorization Scheme from the Scratch 16-21
Creating Users with Roles – Example: Create User 1 16-23
Creating Users with Roles – Example: Create User 2 16-24
Creating an Access Control Page 16-25
Setting Access Control Mode 16-26
Creating Access Control List 16-27
Applying an Authorization Scheme to an Application 16-29
Applying an Authorization Scheme to a Page 16-30
Applying an Authorization Scheme to a Column in a Report 16-31
Example: Applying an Authorization Scheme to a Column in a Report 16-32
Quiz 16-33
Practice 16-2 Overview: Restricting Users by Using Access Control 16-34
Lesson Agenda 16-35
Session State Protection 16-36
Enabling Session State Protection from the Edit Application Page 16-37
Enabling Session State Protection from the Session State Protection Page 16-38
Configuring Session State Protection 16-39
Configuring Session State Protection by Using a Wizard 16-40
Configuring Session State Protection for Pages and Items 16-41
Configuring Session State Protection for Application Items 16-42
Practice 16-3 Overview: Enabling Session State Protection 16-43
Summary 16-44

17 Managing Application Navigation
Steve Tries to Improve Application Navigation 17-2
You Are Here in This Course 17-3
Objectives 17-4
Lesson Agenda 17-5
Building a Hierarchical List with Images 17-6
Building a Hierarchical List 17-8
Example: Building a Hierarchical List 17-9

Building a Hierarchical List with Images 17-10
Practice 17-1 Overview: Building a Hierarchical List with Images 17-13
Lesson Agenda 17-14
Building a Database-Driven Navigation Report 17-15
Quiz 17-19
Practice17-2 Overview: Building a Database-Driven Report 17-20
Lesson Agenda 17-21
Building a Site Map 17-22
Adding Site Map as a Navigation Bar Entry 17-27
Example: Adding Site Map as a Navigation Bar Entry 17-28
Quiz 17-29
Practice17-3 Overview: Building a Site Map 17-30
Lesson Agenda 17-31
Enforcing Authorization on Your Site Map 17-32
Practice 17-4 Overview: Enforcing Authorization on the Site Map 17-35
Summary 17-36

III Unit III Summary: Customizing Your Web Application
Unit III Road Map III-2

IV Unit IV Introduction: Enhancing Your Web Application
Steve Enhances the Project Tracking System Application IV-2
Unit IV Road Map IV-3

18 Extending Your Application
Steve Extends the PTS Application 18-2
You Are Here in This Course 18-3
Objectives 18-4
Lesson Agenda 18-5
Data Load Wizard 18-6
Creating Data Load Wizard Pages 18-7
Data Load Wizard Pages 18-9
Practice 18-1: Creating Data Load Wizard Pages 18-10
Lesson Agenda 18-11
Contact Us Page 18-12
Creating a Send E-Mail Process 18-14
Lesson Agenda 18-15
Creating an Upload and Download Page 18-16
Adding a File Browse Item on the Upload Download Page 18-17
Practice 18-2: Adding an Upload and Download Page 18-18
Summary 18-19

19 Creating and Editing Charts
Introducing Visual Aids for Representing Data 19-2
You Are Here in This Course 19-3
Objectives 19-4
Lesson Agenda 19-5
Search a Workspace for AnyChart Using SQL Commands 19-6
Upgrading from AnyCharts to JET Charts 19-8
Lesson Agenda 19-9
Create Charts 19-10
Creating SQL Queries for Charts 19-11
Creating a Bar Chart 19-12
View and Edit Chart Attributes 19-14
Lesson Agenda 19-15
View and Edit Chart Attributes 19-16
Visualizing Data in a Line with Area Chart and Pie Chart 19-17
Visualizing Data in a Line with Area Chart 19-18
Visualizing Data in a Pie Chart 19-19
Practice 19-1 Overview: Creating and Editing Charts 19-20
Practice19-2 Overview: Creating a Pie and a Donut Chart 19-21
Lesson Agenda 19-22
Create a Pyramid Chart 19-23
Create a Status Meter Gauge Chart 19-26
Viewing and Analyzing a Combined Chart 19-27
Practice19-3 Overview: Enhanced Charting Examples 19-28
Quiz 19-29
Summary 19-30

20 Adding Calendars and Trees
Explores Tools to Organize Project Activities 20-2
You Are Here in This Course 20-3
Objectives 20-4
Lesson Agenda 20-5
Creating a Calendar 20-6
Creating a Calendar on Home Page 20-7
Editing Calendar Attributes 20-8
Dragging and Dropping Calendar Entries 20-9
Defining Links in a Calendar 20-10
Linking to the Calendar from a Button 20-11
Practice 20-1 Overview: Creating a Calendar 20-13
Lesson Agenda 20-14

What Is a Tree? 20-15
Creating a Tree 20-16
Exploring a Tree 20-21
Practice 20-2 Overview: Creating a Tree Whose Nodes Link to a Different Page 20-23
Summary 20-24

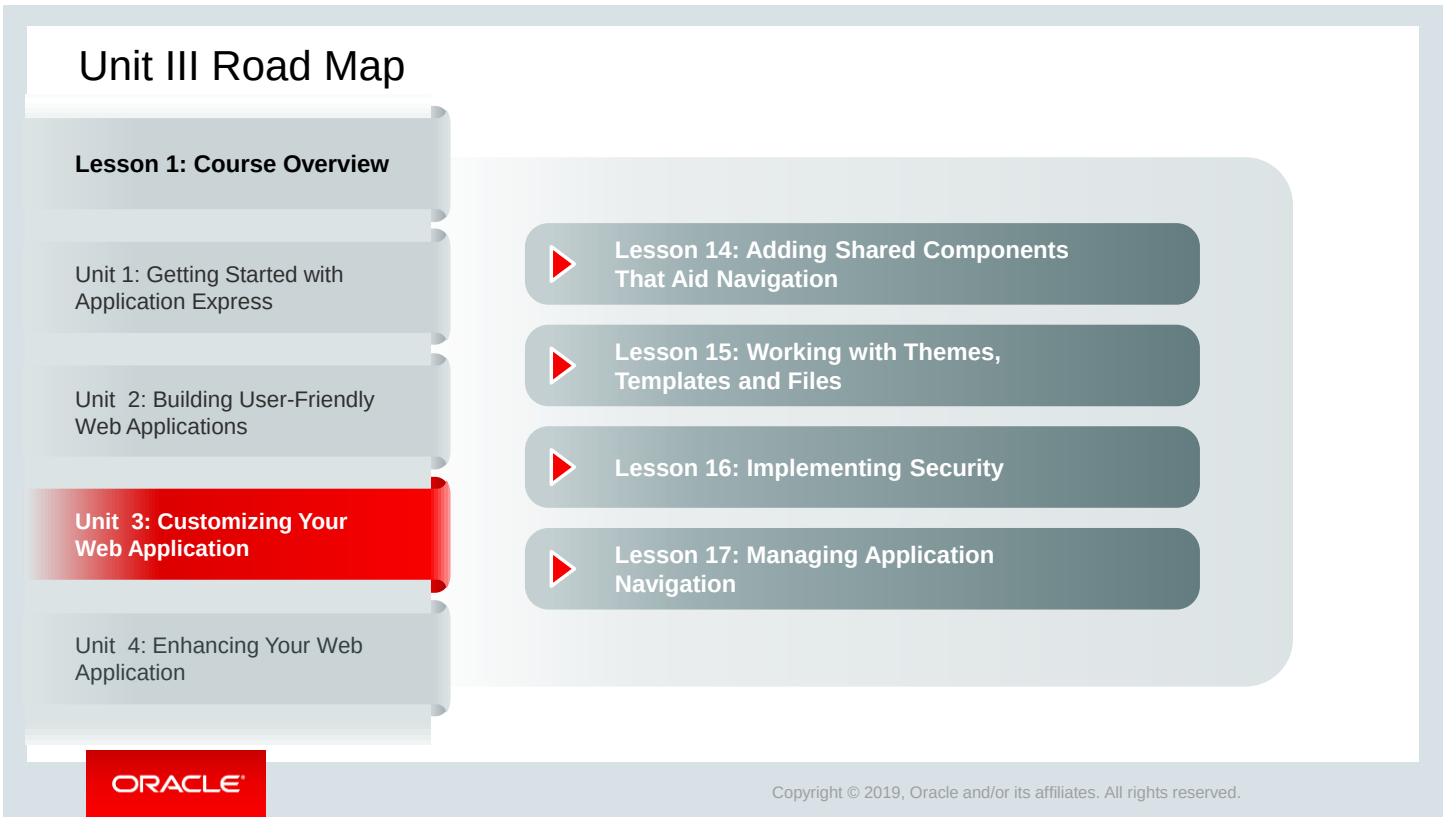
21 Managing Application Feedback
Introducing Feedback Mechanism 21-2
You Are Here in This Course 21-3
Objectives 21-4
Lesson Agenda 21-5
Managing Application Feedback 21-6
Lesson Agenda 21-7
What Is Team Development? 21-8
Tracking the Progress of Your Application Development Project 21-9
Creating Features 21-10
Creating Milestones 21-11
Creating Bugs 21-12
Creating To Dos 21-13
Quiz 21-14
Lesson Agenda 21-17
Review the Progress of Your Milestones and Features 21-18
Lesson Agenda 21-19
Enabling Feedback in your Application 21-20
Step 1: Enabling Feedback in Application Properties 21-21
Step 2: Creating a Feedback Page 21-22
Step 3: Submitting Feedback 21-23
Step 4: Accessing Feedback in Team Development 21-24
Step 4.1: Accessing Feedback as an End User 21-25
Quiz 21-26
Practice 21 Overview: Adding and Monitoring Feedback in Your Application 21-28
Summary 21-29

IV Unit IV Summary: Enhancing Your Web Application
Unit IV Road Map IV-2

Unit III Introduction: Customizing Your Web Application

Steve Designs Application Navigation

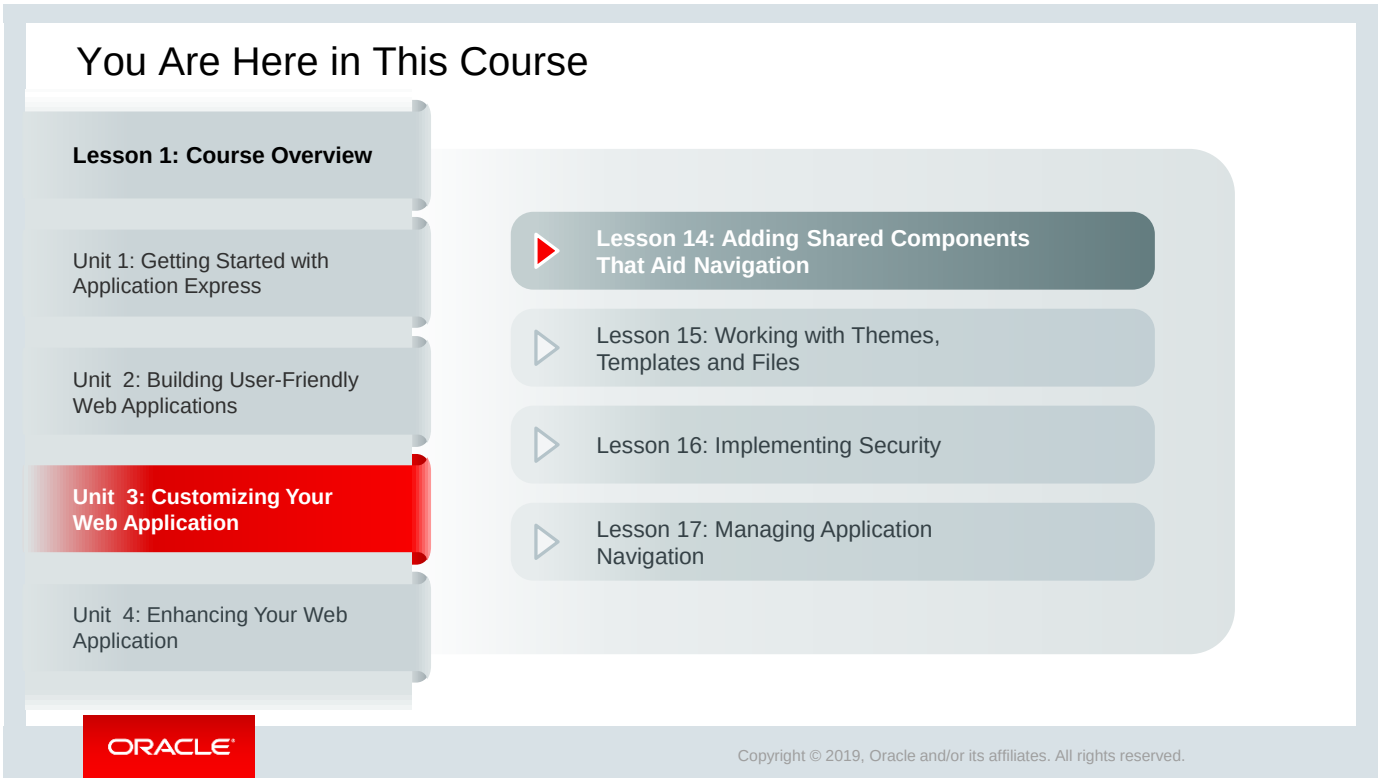
Steve created the *PTS* application, which provides all the pages like Forms and Reports to meet Stella's requirements. But while running the application he feels that it is not too user-friendly when it comes to accessing those features. Therefore, Steve starts customizing *PTS* with Oracle Application Express to get a better user experience before he presents it to Stella and other project managers.



In Unit 3, you include navigation in your application with the help of shared components. This unit also explains how to implement page-level authorization to make your application highly secure.

14

Adding Shared Components That Aid Navigation



This slide is a graphical depiction of the course, particularly highlighting Unit 3 - Lesson 14, which is dealt with in these slides.

Steve Builds a Navigation System into PTS

Steve is very happy with the way the *PTS* application has taken shape in recent days. But he is finding it quite difficult to run the various pages that are built into *PTS*. Steve wants to make navigating through the different pages easy and user-friendly, so that Stella and other project managers do not waste time searching for their relevant pages.

Objectives

After completing this lesson, you should be able to:

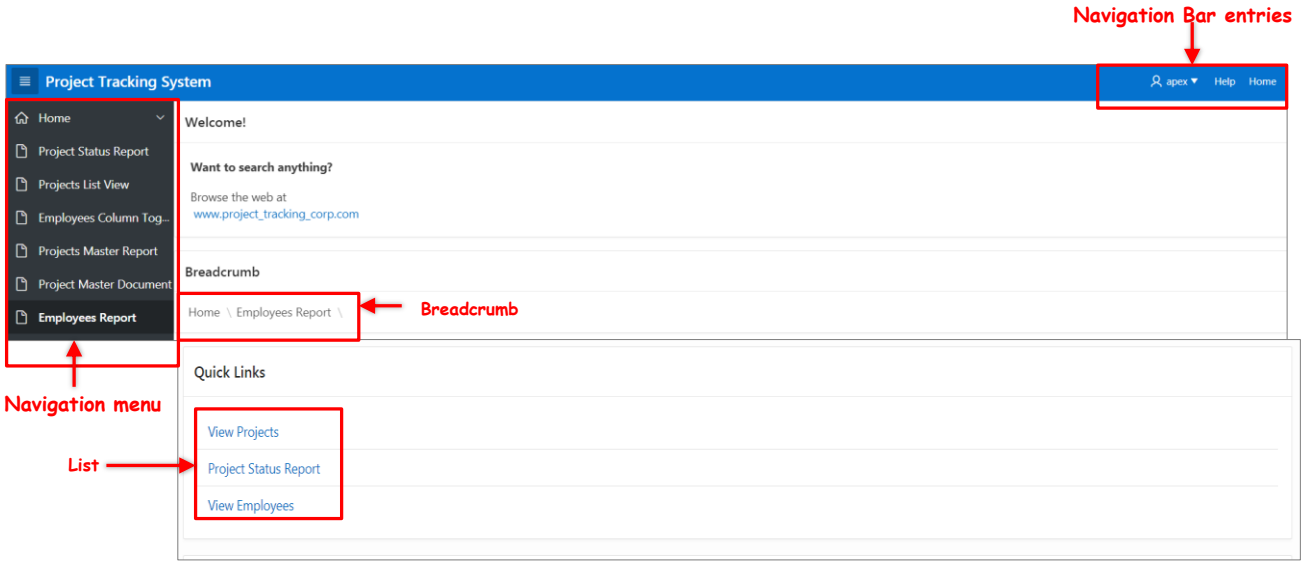
- Explain the use of shared components in an application
- Create and edit the following navigational shared components in an application:
  - Navigation menu entries
  - Lists
  - Breadcrumbs
  - Navigation bar entries

In this lesson, you learn how to create, edit, and use navigational shared components (navigation bars, lists, and breadcrumbs) in your application.

Lesson Agenda

- Introducing Shared Components
  - Shared Components: Examples
  - What Are Shared Components?
- Creating Navigation Menu entries
- Creating Lists
- Creating Breadcrumbs
- Creating Navigation Bar entries

Shared Components: Examples



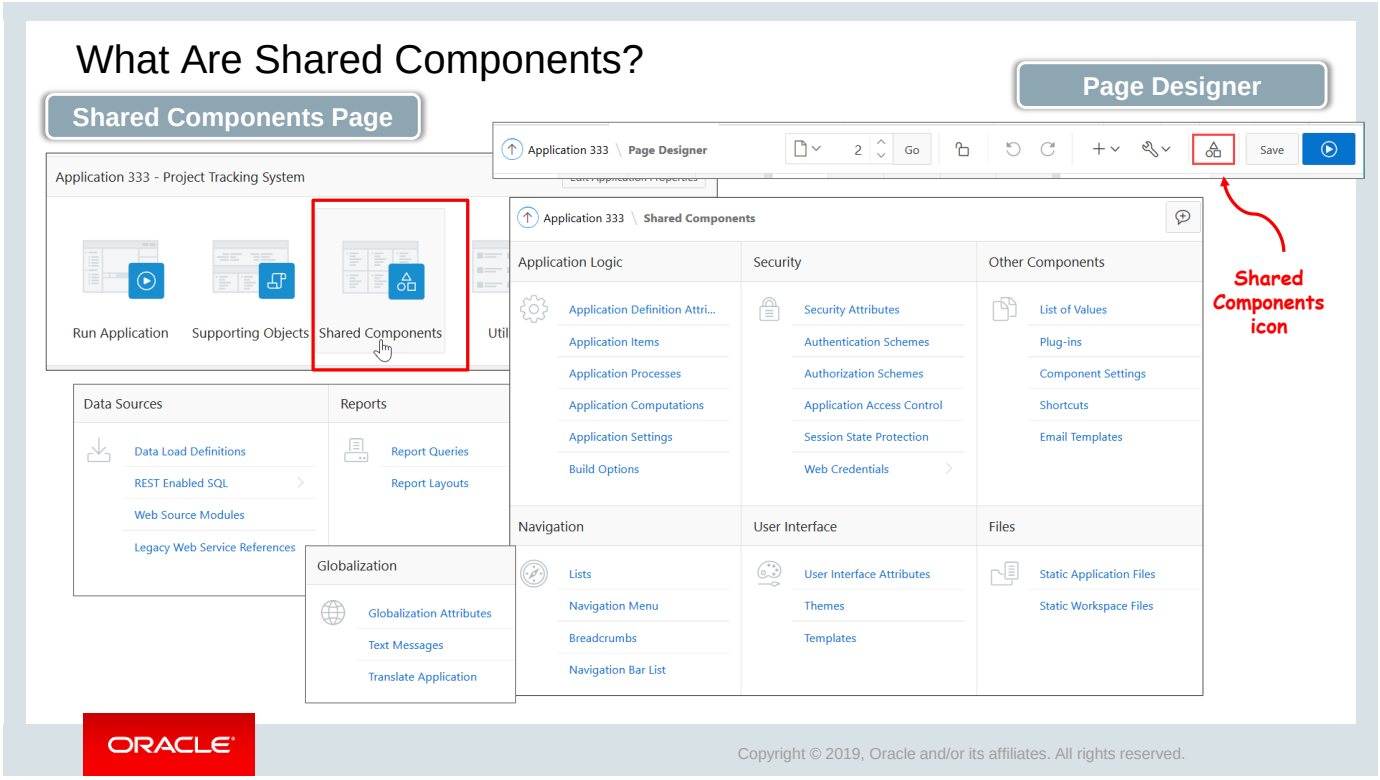
The *PTS* Application interface screen in the slide shows that an application typically uses a combination of navigation menus, lists, navigation bars, and breadcrumbs.

The *Home*, *Project Status Report*, *Projects List View*, *Employees Column Toggle*, *Projects Master Report*, *Projects Master Document*, and *Employees Report* pages are the **Navigation Menu** entries. *Help*, *Username*, and *Home* links at the top right of the page are the **Navigation Bar** entries. *Home* > is the **Breadcrumb** used to go back and forth between the pages within the application's major components. The *Quick Links* on the top of the page is a **List**.

Thus, you can use a combination of navigation menu entries, lists, navigation bar entries, and breadcrumbs to navigate within an application.

- Navigation menu is provided with Universal Theme by default, and it is used to provide navigation between major components of the application.
- A list is a collection of links. Each list entry is associated with a page.
- Breadcrumbs are a hierarchical list of links. They show you where you are within the application.
- A navigation bar is used to link text or an image to a page. You need not reference it on every page (as you must do with the other navigational shared components). An application can have only one navigation bar.

All of these are *Shared Components*. So, what is a Shared Component? Let's learn in the next slide.



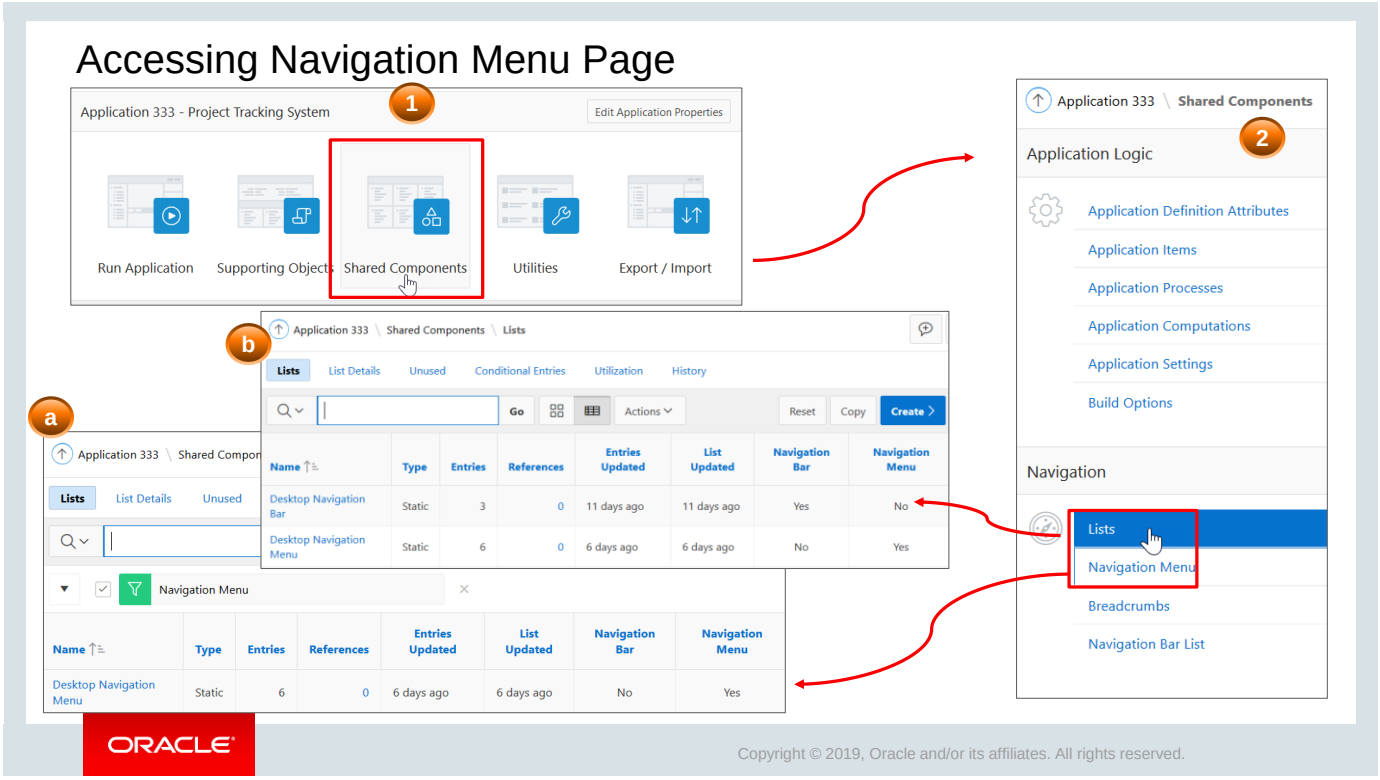
Shared components are components that can be included on one or more pages of your application. The Shared Components Page screenshot in the slide shows the categories of shared components that you can include in your application.

In the Shared Components section of a page's definition (see the screenshot under Page Designer), you can view the shared components that are included on that page.

In the next few slides, you learn how to create navigational shared components: lists, breadcrumbs, and navigation bar entries.

Lesson Agenda

- Using Shared Components
- Creating Navigation Menu Entries
  - Accessing Navigation Menu page
  - Creating Navigation Menu entries
- Creating Lists
- Creating Breadcrumbs
- Creating Navigation Bar Entries



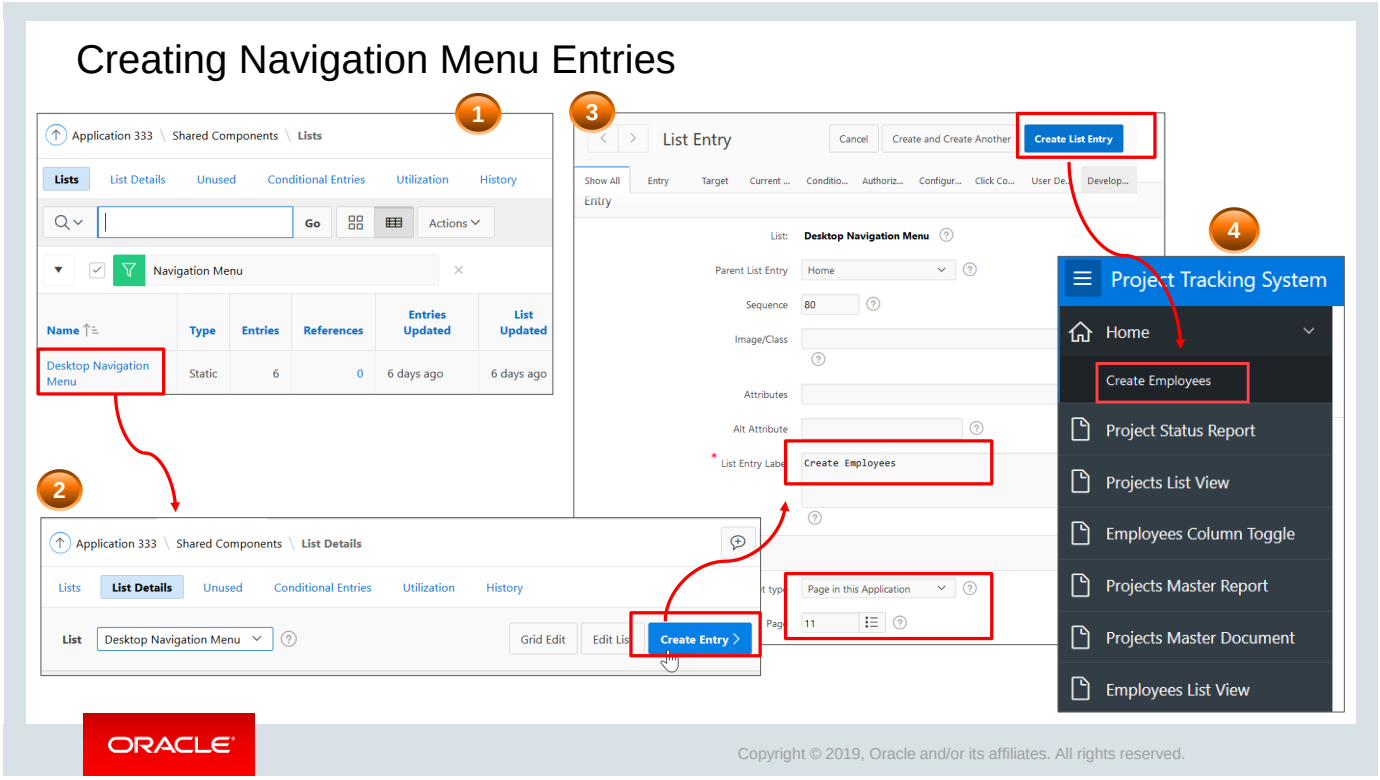
Because Steve created the *PTS* application as a desktop application, you can see that Desktop Navigation Menu already created in PTS shared components. So, how do you access the Navigation Menu page?

To access Navigation Menu shared components:

1. In the development workspace, open your application's home page (screenshot 1).
2. Click the **Shared Components** icon on the application's home page.
3. Locate the **Navigation** group and click **Navigation Menu.**

Alternatively, you can access it by performing the following steps:

1. Navigate to the application home page.
2. Click the **Shared Components** icon on the application's home page.
3. Locate **Navigation** group and click **Lists**.



Now, Steve wants to create new Navigation Menu entries for the PTS application such that all the forms have an entry in the Navigation Menu. Project managers can click any menu item to access the corresponding form to manage details about employees, projects, and so on. And he starts with the *Create Employees f*orm page. Let's see how.

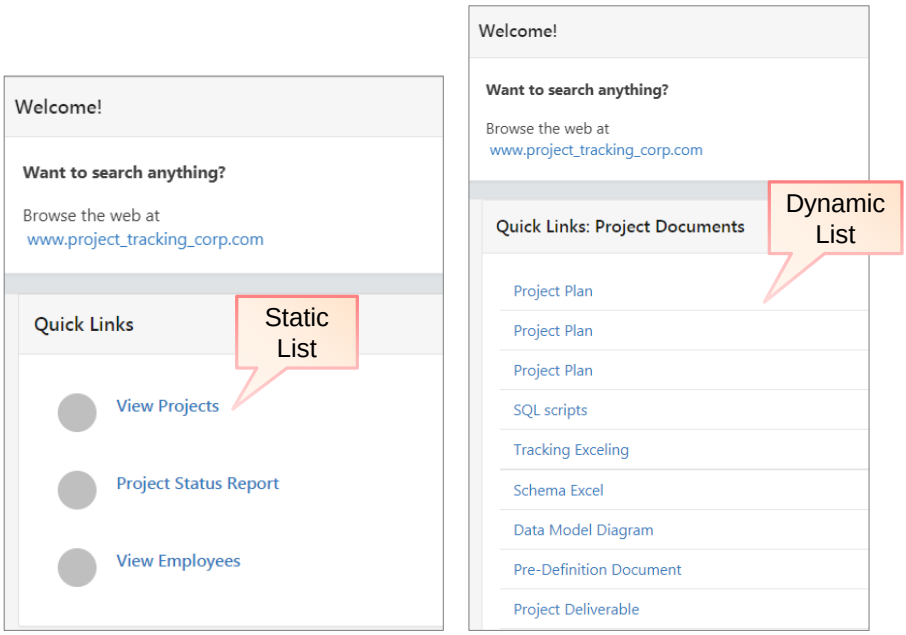
To create Navigation Menu entries, in the development workspace, open your application's home page, and under **Shared Components,** locate **Navigation** group and click **Navigation Menu**. You then perform the following steps:

1. Click **Desktop Navigation Menu** and then **Create List Entry.**
2. Enter the following values and click **Save**:
   - **Parent List Entry**: Select **Home.**
   - **List Entry Label**: Enter `Create Employees.`
   - **Target Type**: Select **Page in this Application.**
   - **Target Page**: Select **Page 11** (this is the *Create Employees* page) from the pop-up LOV.
3. Run the Home Page to see the *Create Employees* form listed under *Home* in the Navigation Menu. Click the *Create Employees* link to load the *Create Employees* form.

Lesson Agenda

- Using Shared Components
- Creating Navigation Menu Entries
- Creating Lists
  - Accessing the Lists Page
  - Creating a Static List
  - Creating a Static List Region
  - Creating a Dynamic List
  - Creating a Dynamic List Region
  - Creating a List Region on Global Page
- Creating Breadcrumbs
- Creating Navigation Bar Entries

Lists

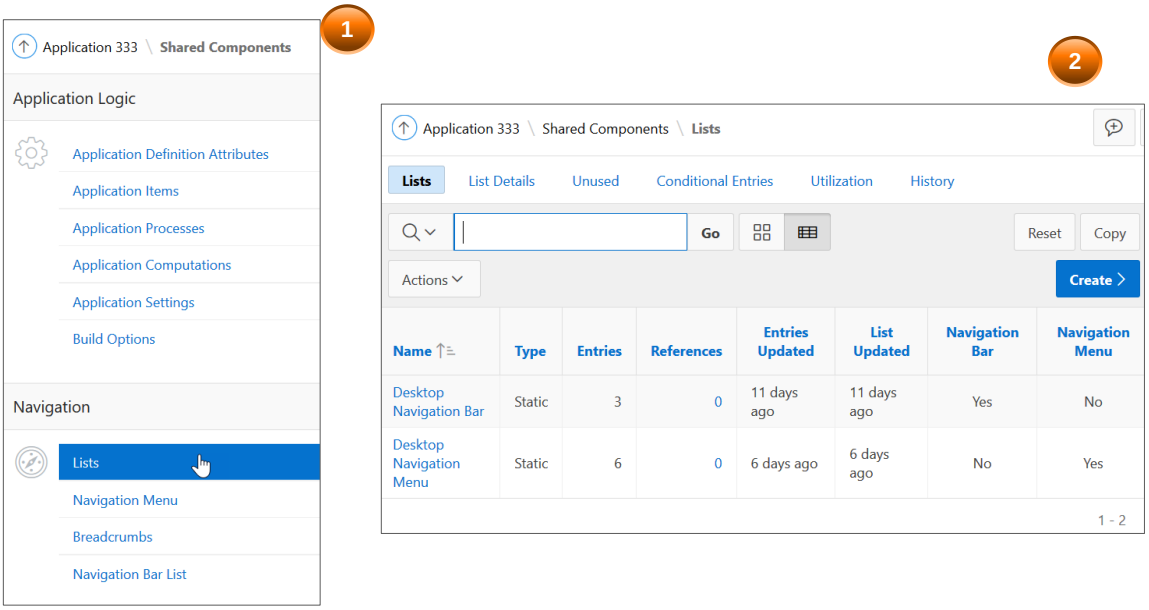


See the two list images in the slide: one a static list (based on predefined display and return values) and the other a dynamic list (based on a SQL query or a PL/SQL function executed at run time). Steve wants to create these two list types for his *Home Page* and his *Project Master Document* page so that all the important pages and documents can be easily accessed by his project managers.

So, what is a list?

A list is a collection of links. Each link is called a list entry. For each list entry, you must specify the display text, a target URL, and other attributes that control when and how the entries in the list are to be displayed. In the next few slides, you learn how to access the Lists Page and create different types of lists.

Accessing the Lists Page



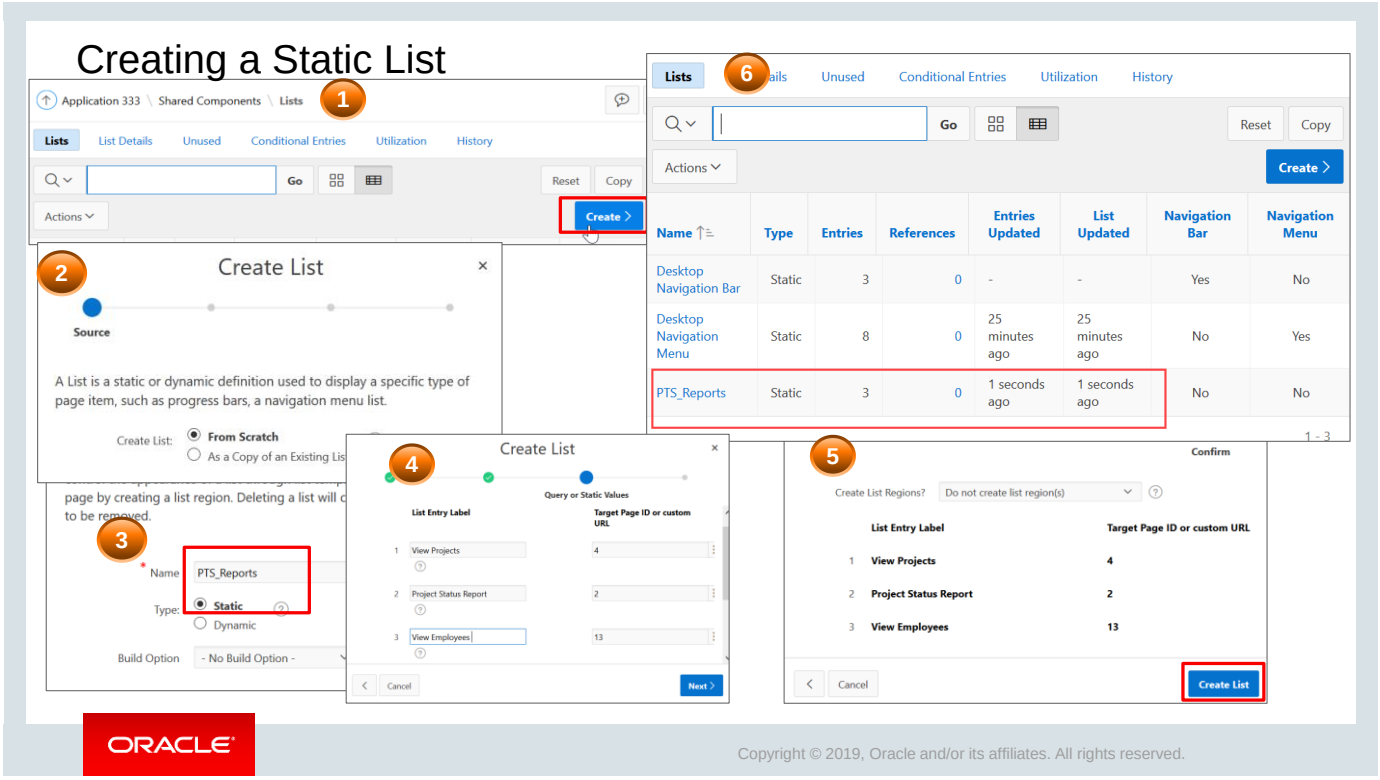
To access the Lists page, on the application home page, click the **Shared Components** icon and perform the following steps:

1. On the **Shared Components** page, click the **Lists** link under Navigation.
2. The Lists page is displayed. Existing Lists, if any, are displayed on the Lists tab. You can create a new list or copy a list from another application. (The other application must reside in the same workspace.)

Alternatively, perform the following steps:

1. On the application home page, click a page.
2. On the **Shared Components** tab in the page definition, right-click the **Lists** node and select **Create**.

   The Create List Wizard opens.



Now, Steve starts creating a static list with links for all the reports generated from *PTS* and creates a list region to display this list on the Home page. This enables project managers to access any report by clicking the links on the list directly. Let's see how.

To create a static list (for *PTS Reports*), perform the following steps:

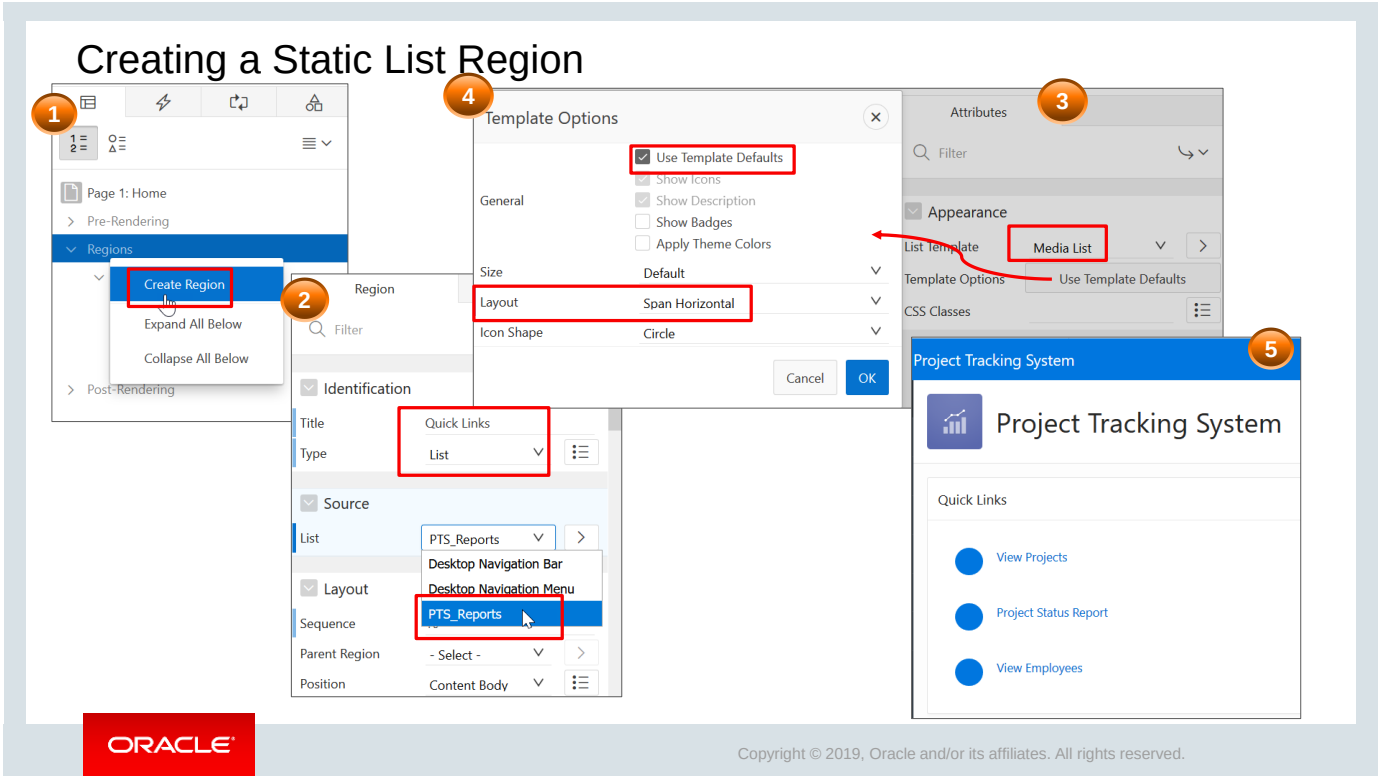
1. Click the **Create** button on the Lists page.
2. Select **From Scratch** and click **Next** (screenshot 2).
3. Enter a name for the list. (Here Steve enters *PTS_Reports* (screenshot 3).)
4. Enter following values for List Entries. **Target Page ID** should be selected from the LOV and click **Next** (screenshot 4):
   - `View Projects - 4`
   - `Project Status Report - 2`
   - `View Employees - 13`
5. Select **Do not create List Region**. (You will learn to create a list region to display your list in the next slide.) Click **Create List**. The *PTS_Reports* list is created. You can edit the list to add additional list entries.

You can also create new list entries in lists that are already populated. To create a list entry, perform the following steps:

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.

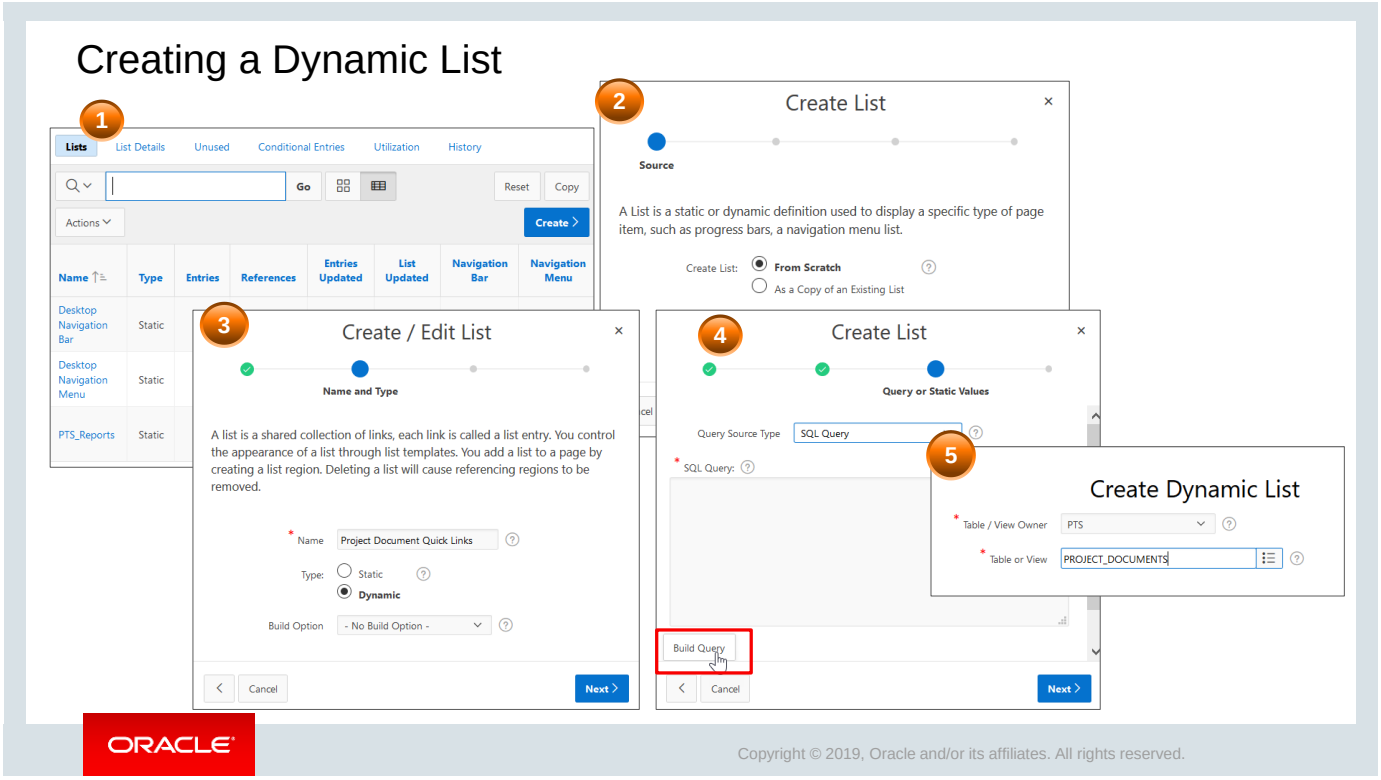
In the next slide, you will create List Region on Home Page to make the *PTS_Reports* list visible on the Home Page.



Although, in the previous slide, you selected *Do not create List Region*, you need to create a list region to display your list. Usually, it is done on the Home Page. You can create a list region on the current page also while creating the list (in the previous slide). Alternatively, you can also create a list region separately on the page where you want to display the list. In this slide, you will learn how.

To create a list region on the Home Page to make the *PTS_Reports* list visible on the Home Page, open the *Home* Page in Page Designer view and perform the following steps:

1. Under Rendering, right-click **Regions** and click **Create Region**.
2. Enter the following values in the Properties Pane of the new Region on the right side:
   - **Title**: `Quick Links`
   - **Type**: `List`
   - **List** (under Source): `PTS_Reports` (select *PTS_Reports* from select list)
3. Click **Attributes** under **Quick Links** to see its properties in Property Editor.
4. Select **Media List** for **List Template** and select **Use Template Defaults** from **General**.
5. Select **Span Horizontal** for **Layout** and click **OK**.
6. Click the **Save and Run** icon to see the list in the center of the Home page.



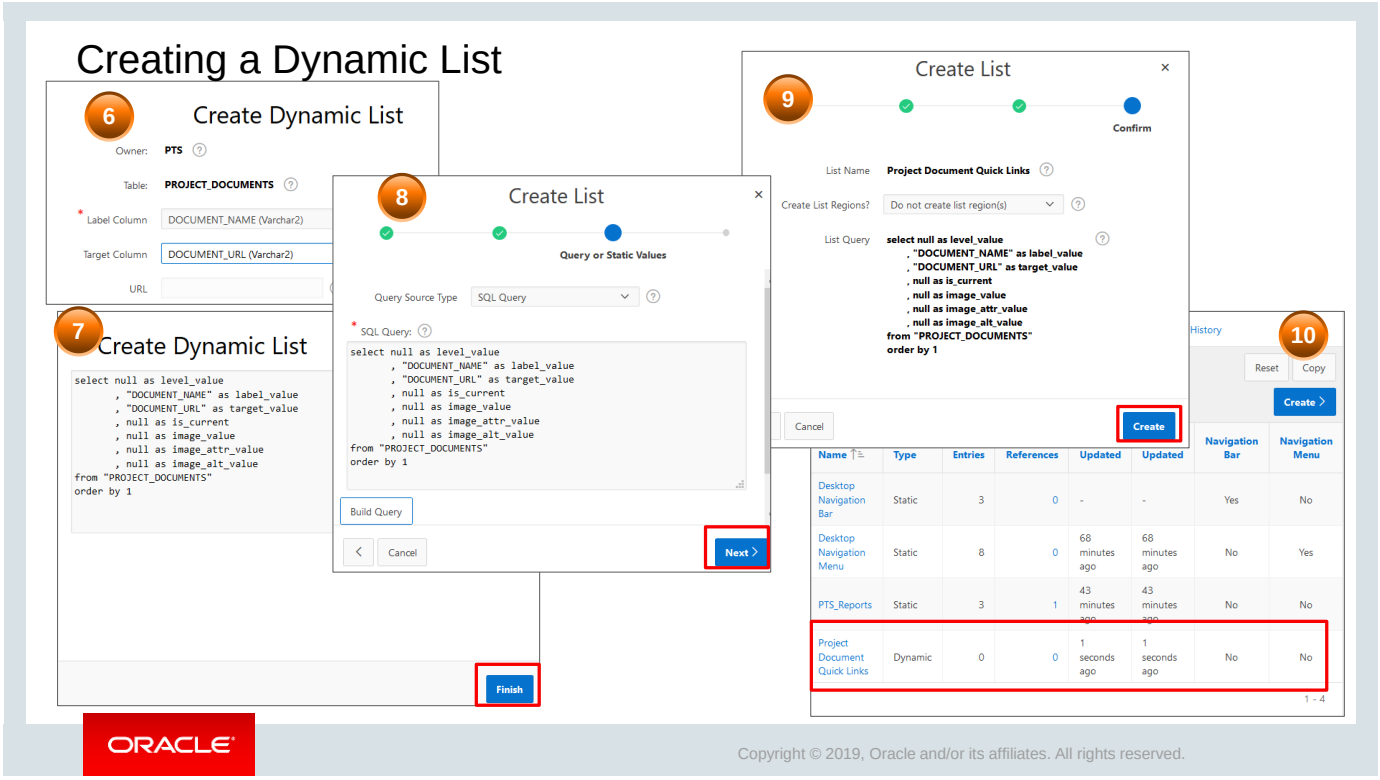
Steve now starts creating another list with links for all the project documents maintained in PTS applications. He creates a list region to display this list on the *Project Master Documents* page. This will help project managers access any project document by clicking the links on the list directly.

This time Steve chooses a Dynamic List, because dynamic lists query the database at run time and displays the list. So, any new entry or updates done to the underlying table will be reflected in the list, and he does not have to go and edit all these pages every time a new project document is added.

Let's see how he creates a Dynamic List – **Quick Links: Project Documents**.

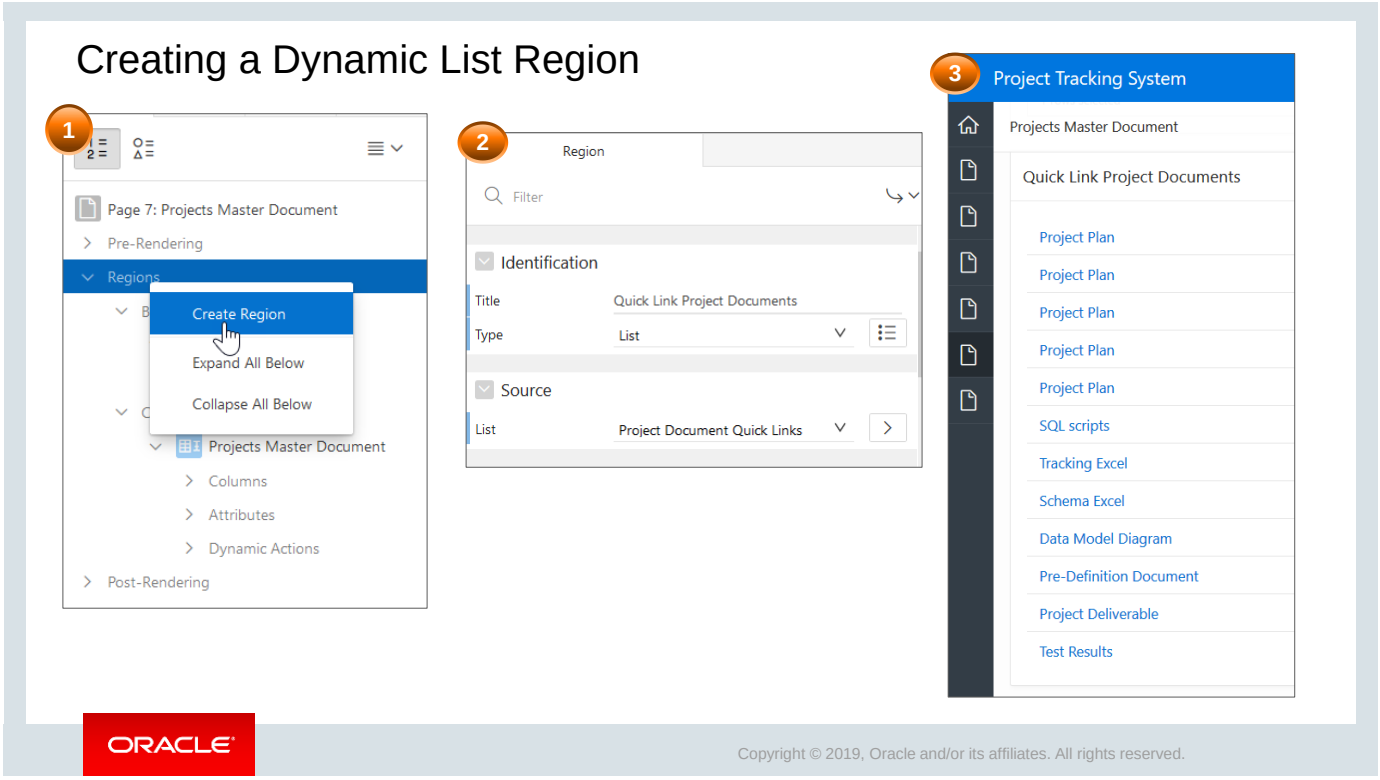
You must first open your application and click the **Shared Components** icon. On the Shared Components page, click **Lists** under **Navigation**, and perform the following steps:

1. Click **Create.**
2. The **Create List** Wizard appears. Select **From Scratch**.
3. Enter the following values (retain other values as default) and click **Next**.
   - **Name** - Enter `Project Document Quick Links`
   - **Type** - Select **Dynamic**
4. Click **Build Query** (screenshot 4).
5. The **Create Dynamic List** window opens. Select `PROJECT_DOCUMENTS` from the pop-up LOV and click **Next** (screenshot 5).



6. Enter the following values, leaving others as default, and click **Next** (screenshot 6).
   - **Label Column**: `DOCUMENT_NAME (Varchar2)`
   - **Target Column**: `DOCUMENT_URL (Varchar2)`
7. The SQL query is created. Click **Finish** (screenshot 7).
8. Click **Next** (screenshot 8).
9. Select **Do not create list region(s)**. You will learn to create a list region to display your list in the next slide. Click **Create** (screenshot 9).

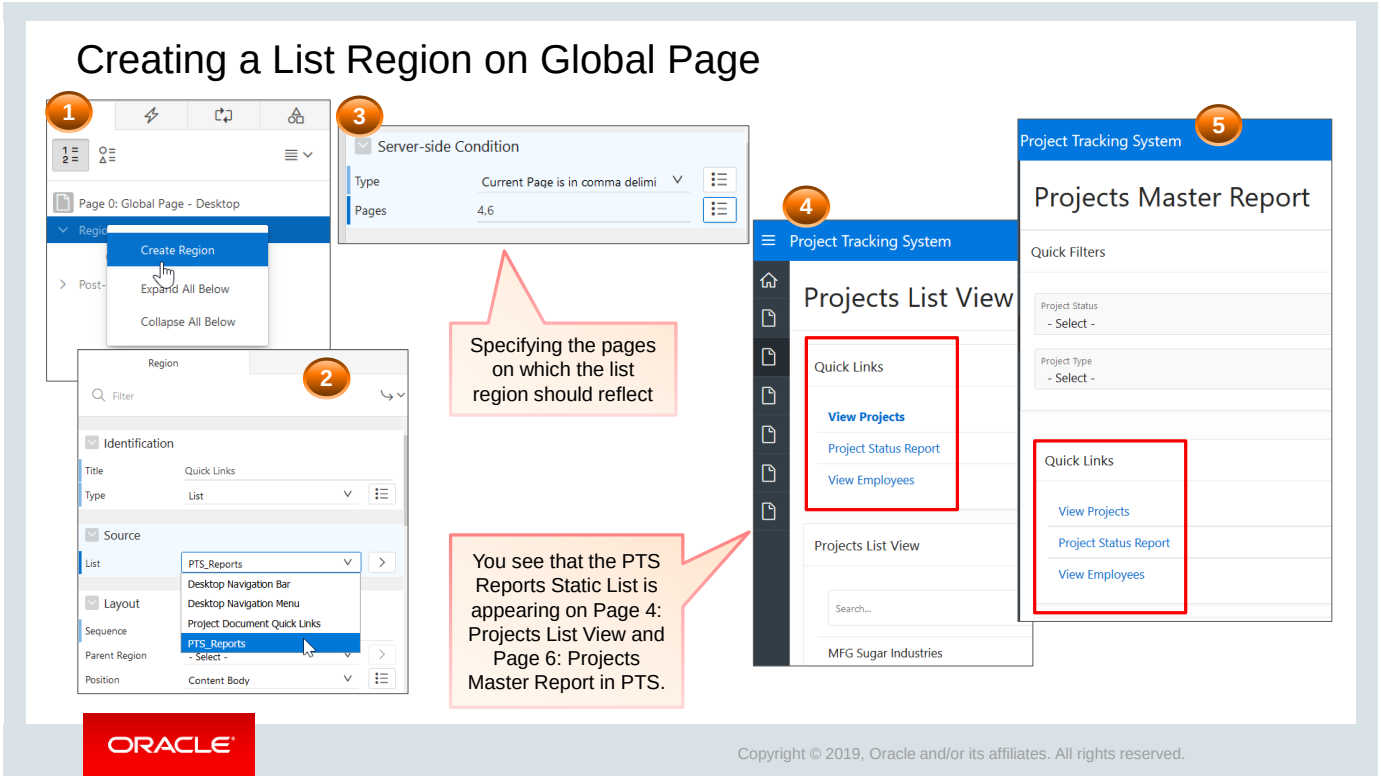
The Dynamic list *Project Document Quick Links* is created and can be seen listed under Lists on Shared Components (screenshot 10).



Now that Steve has created the Dynamic list for his projects documents, he wants to add it to his *Projects Master Document* page. Let's see how he creates a list region on *Page 7: Project Master Document* to make the Dynamic List visible.

Open the *Project Master Document* page in Page Designer view and perform the following steps:

1. Under Rendering, right-click **Regions** and click **Create Region**.
2. Enter the following values in the Properties Pane of the new region on the right:
   - **Title**: Enter `Quick Links: Project Documents.`
   - **Type**: Select **List.**
   - **List** (under Source): Select **Project Documents Quick Links** from the pop-up LOV.
3. Click the **Save and Run** icon to see the list on the Project Master Document Page.



You learned in the previous slides that to display a list on a page, you must create a list region. You can either create separate list regions on individual pages or you can create a list region on a Global page so that it appears on all the pages. You can even specify the pages that should reflect the list region.

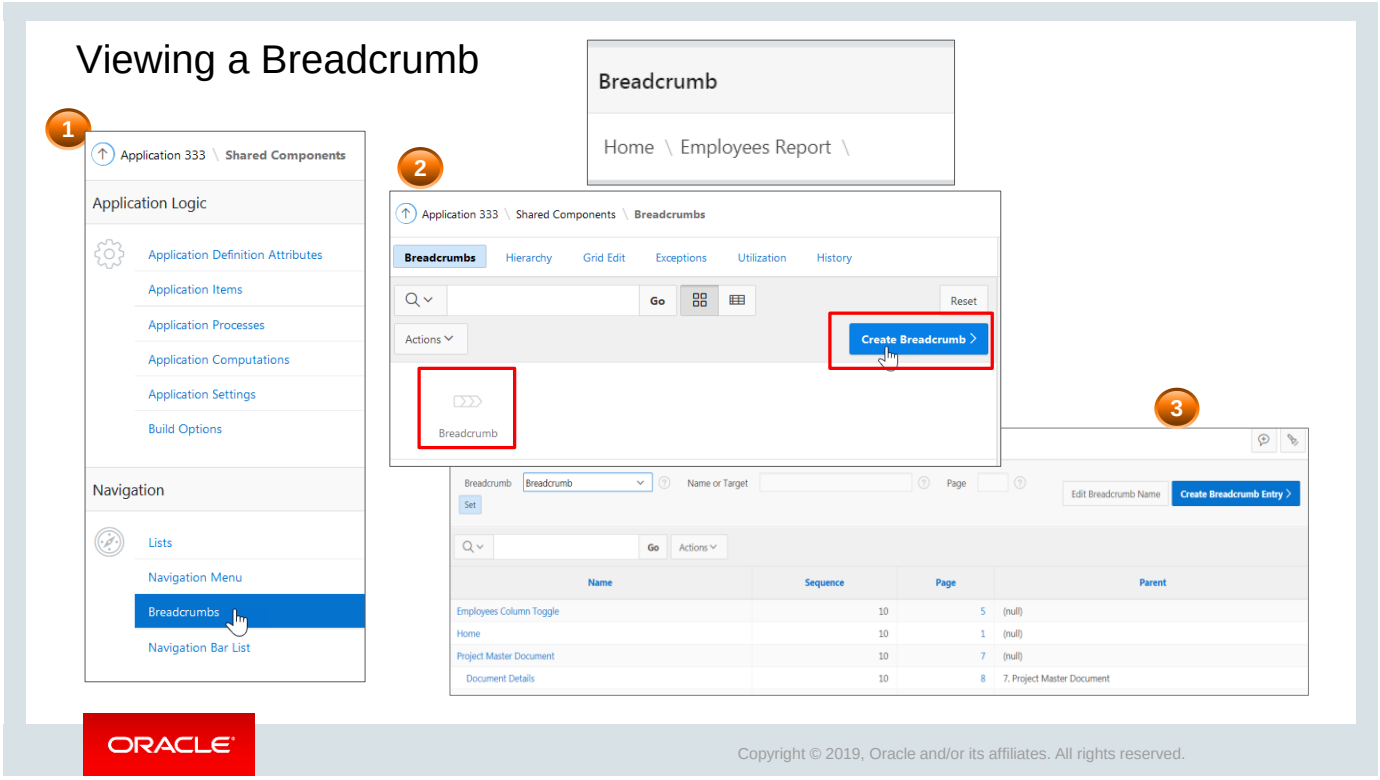
In the slide, Steve adds the static list that he created some time back on the *Global page*. This will enable project managers access the reports quickly from any page in the application.

1. To create a list region, from the page definition for Global page, right-click the **Regions** node and click **Create Region**.
2. Select List in the Property Editor of the new region. Here Steve selects *PTS_Reports*. You can also update other properties such as Title, Position on the page, and so on. The list region is created.
3. To specify the pages on which the list region should be displayed, select the list region node and click the Server-side Condition tab in its Property Editor. Select “Current Page is in comma delimited list” and select the pages in which you want this list region to appear using the pop-up LOV. Here Steve selects page 4 (*Projects List View*) and 6 (*Projects Master Report*).

If you run the application, you should see the list region on the pages that you specified.

Lesson Agenda

- Using Shared Components
- Creating Navigation Menu Entries
- Creating Lists
- Creating Breadcrumbs
  - Viewing a Breadcrumb
  - Creating Breadcrumb Entries
  - Reparenting Breadcrumbs
- Creating Navigation Bar Entries



The slide shows an image in a gray box—a hierarchical list of links or a breadcrumb showing you where you are within the application.

The breadcrumb path is displayed below the Navigation Bar at the top of each page. You can use a breadcrumb path and click a specific page name link to view that page immediately.

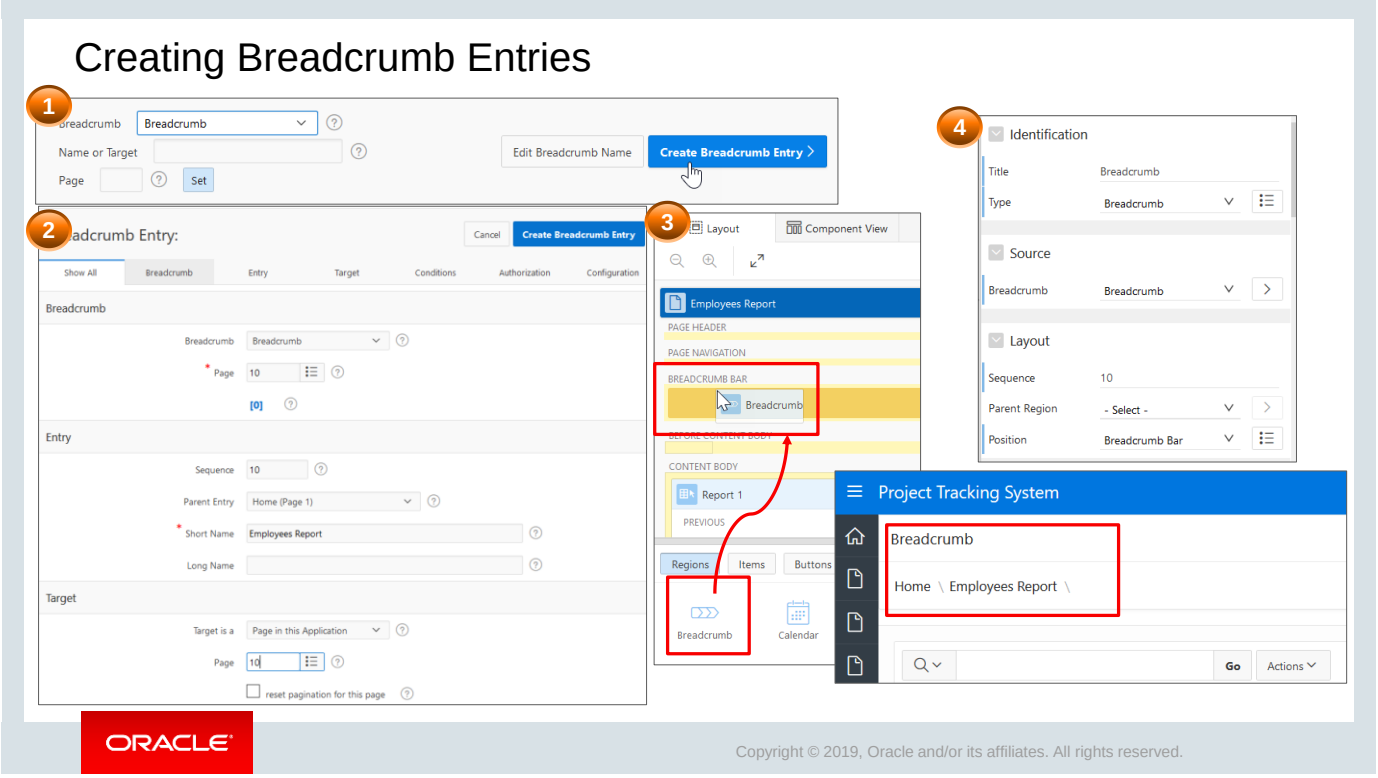
You can define the breadcrumb region on the Global page so that it appears on all pages or on each page individually. You can define conditions to exclude the breadcrumb region from specific pages where they are not to be displayed, such as pop-up LOV pages.

By default, each application contains one breadcrumb. The breadcrumb contains multiple breadcrumb entries.

The Create Page Wizard provides an option to create a breadcrumb entry. To view the breadcrumb for an application, perform the following steps:

1. On the **Shared Components** page, click the **Breadcrumbs** link in the Navigation pane.
2. On the Breadcrumbs page, the existing breadcrumb is listed. Click the icon to view the breadcrumb entries for the breadcrumb. To create a new breadcrumb, click the **Create Breadcrumb** button.
3. The Breadcrumb Entry page appears where you can define the page details for which a breadcrumb entry is required. Alternatively, you can create a breadcrumb entry for a page while creating the page by using the Create Page Wizard itself.

In the next slide, let's see how Steve creates a breadcrumb entry on the *Employees Report* page.



Steve knows that project managers use the *Employees Report* very often to add or update details of the existing employee or when a new employee joins in. Therefore, it will be easy for them if there is a breadcrumb entry created for this report page, which can take them directly to the report. Let's see how Steve creates this breadcrumb entry.

You must first click the **Shared Components** icon and click **Breadcrumbs** under Navigation. You then click **Breadcrumb**, and it opens with the entry for Home in it already. Perform the following steps after that:

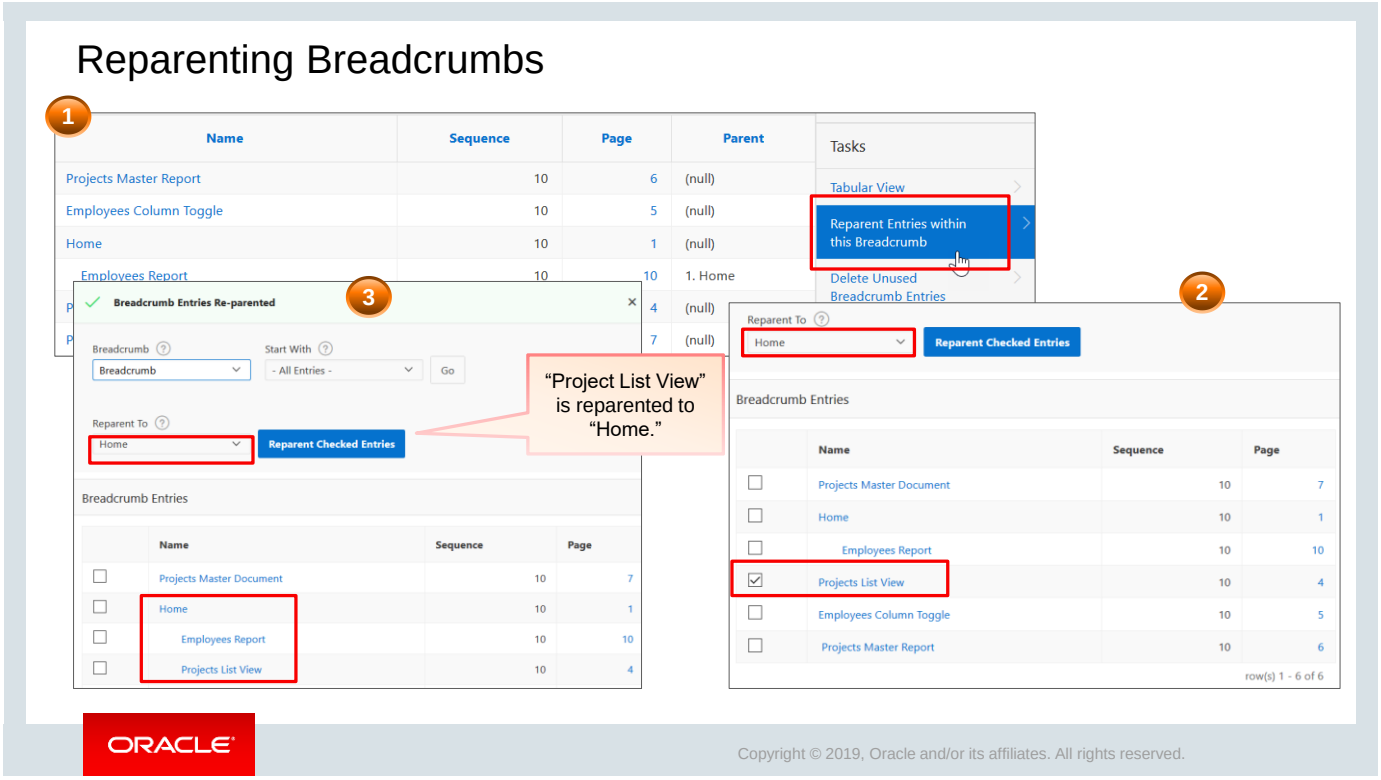
1. On the Breadcrumbs page, the existing breadcrumb is listed. To create a new breadcrumb, click the **Create Breadcrumb** button.
2. Enter the following values (retain defaults for rest of the fields) and click Create Breadcrumb Entry.
    - **Parent Entry:** Select **Home** *(Page 1).*
    - **Page:** Select **10** (from pop-up LOV).
    - **Short Name:** Enter `Employees Report.`
    - **Target is a:** Select **Page in this application.**
    - **Page:** Select **10** (from pop-up LOV).

Now, in order to show the breadcrumb on **Page 10**, you need to create a breadcrumb region on Page 10 as follows:

3. Open *Page 10: Employees Report* in Page designer view and select **Breadcrumb** from **Regions Gallery** and drag it to **Breadcrumb Bar** in Grid Layout.

4. Enter the following values in the Properties Pane of the new region on the right side:
    - **Title:** Enter `Breadcrumb.`
    - **Type:** Select **Breadcrumb.**
    - **Source:** Select **Breadcrumb.**
    - **Position:** Select **Breadcrumb Bar.**
5. Click the **Save and Run** icon to load page 10. You can see the breadcrumb entry for Page 10.

In the next slide, let's see how you can reparent a breadcrumb entry.



You learned how to create a breadcrumb entry in the previous slide. Breadcrumbs, as you know, provide users with hierarchical navigation. You can also select a new parent for selected breadcrumb entries on the Reparent Entries page. Generally, you reparent a breadcrumb when you add additional pages to your application. For example, you initially define a Report (page 3) with a Form (page 4) on the `PROJECTS` table. You then define a new page with Project Details (Page 10) and subregions for the child tables (Milestones and so on). You update the report (Page3) to link to the Project Details page (page 10). Now, on page 10, you include an edit link to go to the form page (Page 4) for projects. So, the breadcrumbs for Page 4 need to be reparented to go to page 10 and not page 3.

Note that you can change the parent entry for one or more breadcrumb entries.

To reparent the breadcrumb entries, perform the following steps:

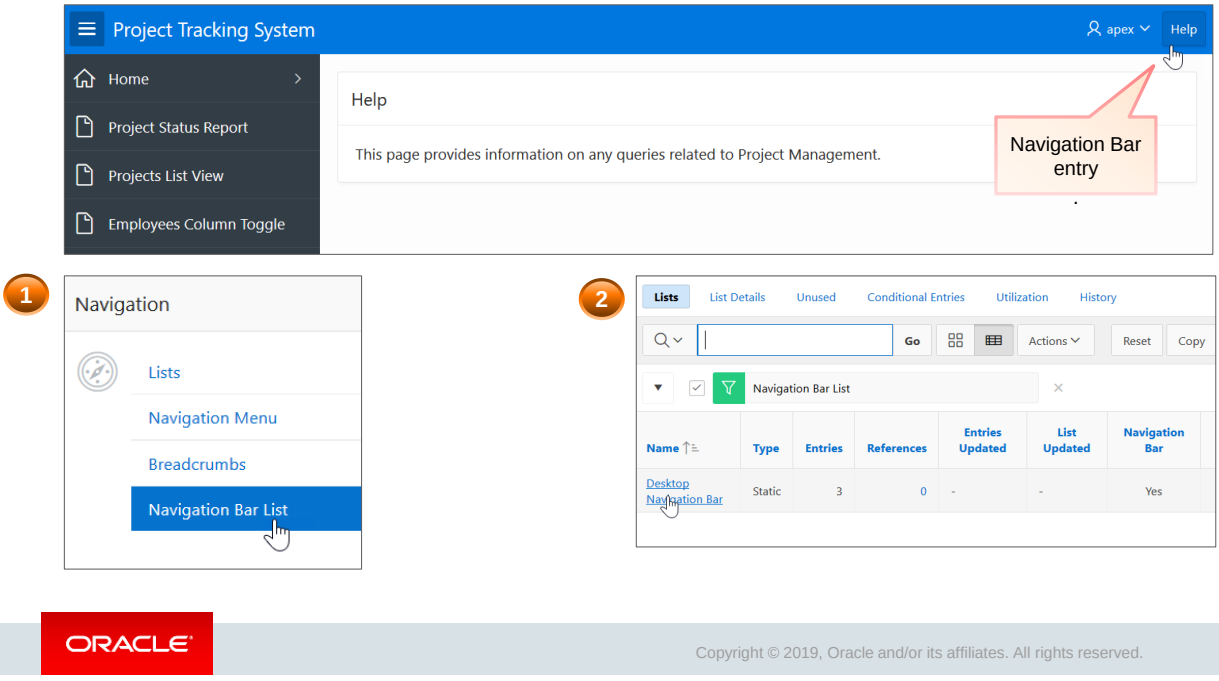
1. On the Breadcrumb page, select **Reparent Entries within this Breadcrumb** from the Tasks menu (in the bottom-right corner of the page).
2. Select a parent entry for the **Reparent To** field. Select the check box for each breadcrumb that you want to reparent. In this slide, Steve selects *Project Details.* He wants to select *Home* as the new parent. Click the **Reparent Checked Entries** button.

   The entry is now listed under the new parent.

Lesson Agenda

- Using Shared Components
- Creating Navigation Menu Entries
- Creating Lists
- Creating Breadcrumbs
- Creating Navigation Bar Entries
  - Accessing the Navigation Bar Entries Page
  - Creating a Help Page
  - Creating a Navigation Bar Entry

Accessing the Navigation Bar Entries Page

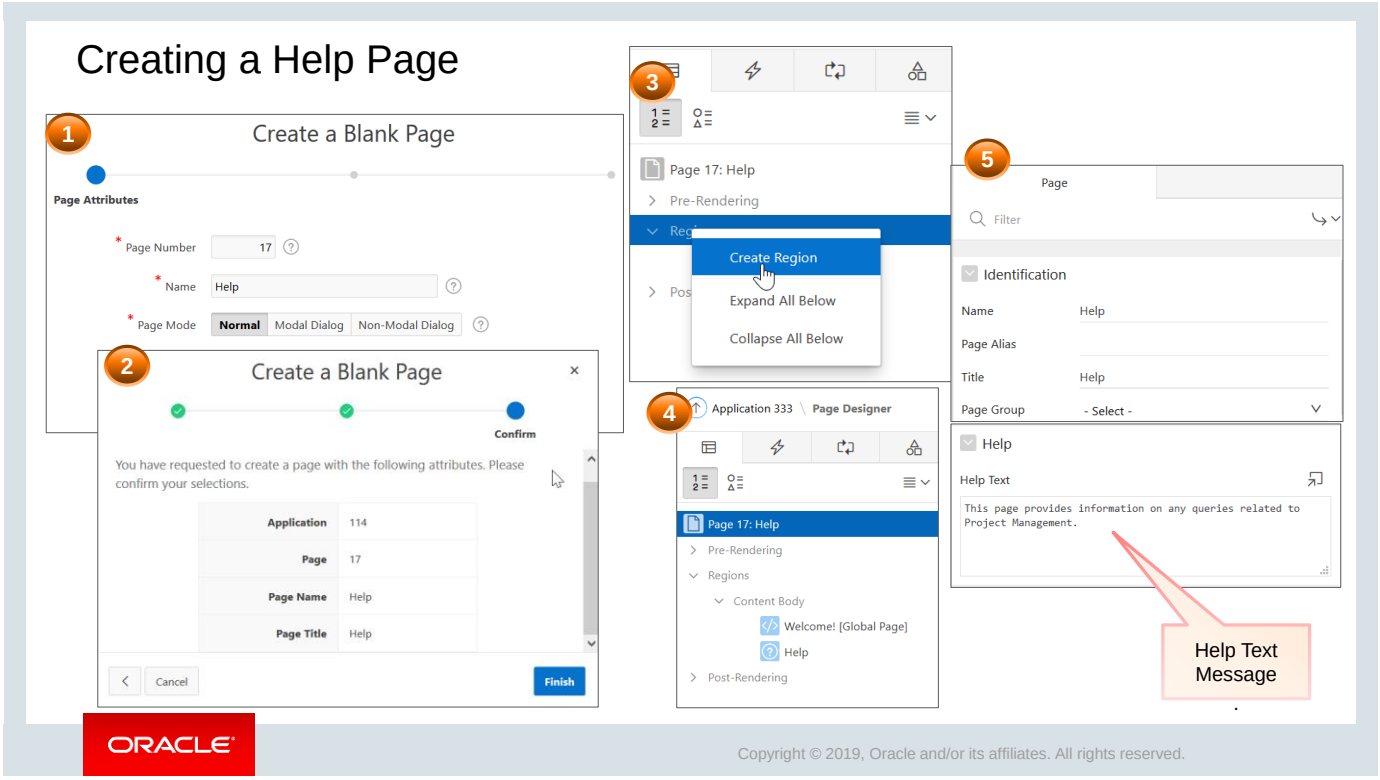


Can you see *Help* and *apex (*username) on the top right-hand side of the page? Both of these are navigation bar entries.

Note that each application can have only one navigation bar. The items inside the navigation bar are called navigation bar entries. Some of the typical situations where you use navigation bars are accessing the Home page and linking to a Help page. The location of the navigation bar depends on the associated page template. You use text or images when you create a navigation bar icon.

If you click the **Navigation Bar List** link from the application's **Shared Components** page, you can view the navigation bar entries for the application.

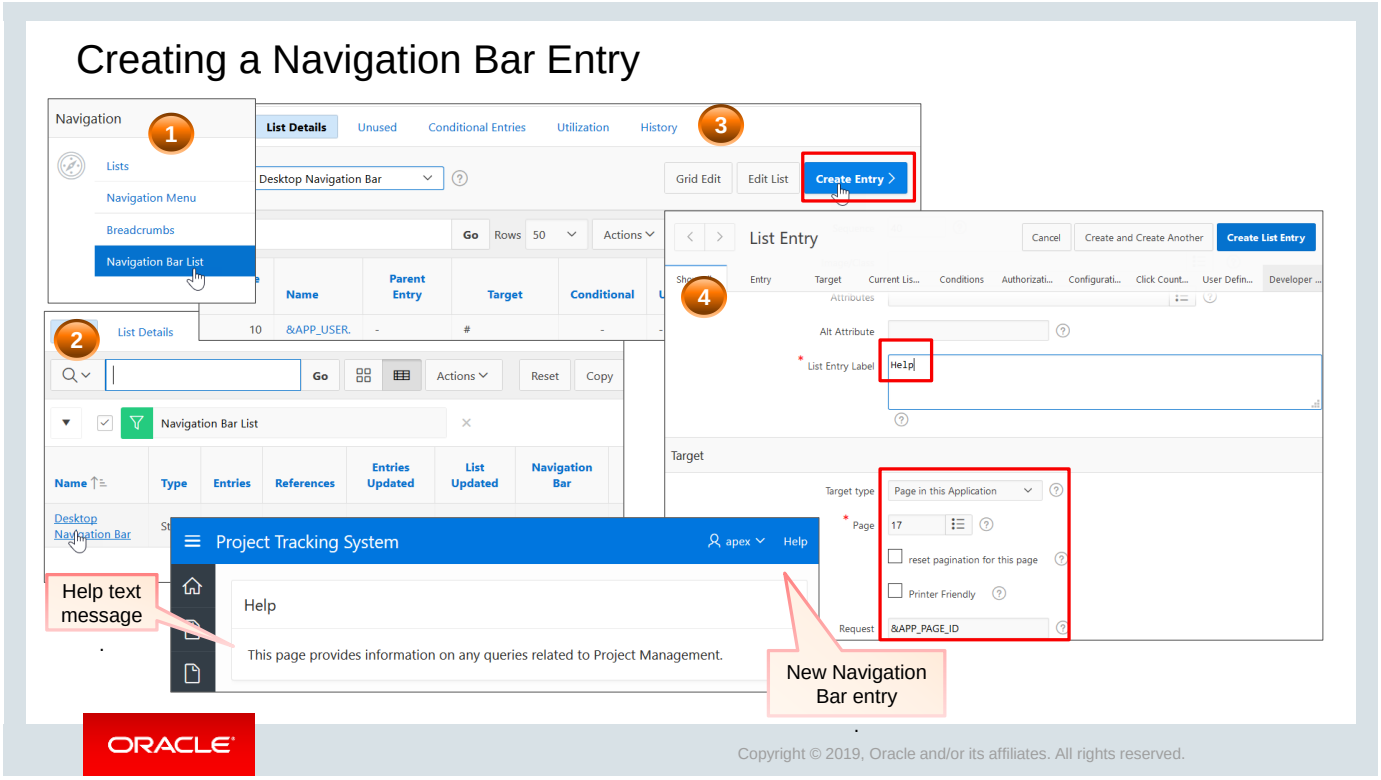
In the next couple of slides, you will create a *Help* page and add it as a navigation bar entry.



Now, Steve's requirement is that he wants to create *Help Text* for all the pages created in *PTS.* He then wants to add a *Help* navigation bar entry so that new users to *PTS* can get help on each page in the application. But, before he creates a *Help* navigation bar entry, let's first see how he creates a *Help* page in the application:

1. Create a blank page with the rest of the options as defaults and click **Finish**:
   - **Page Name**: Enter `Help`
2. In the page definition of the blank page (*Help*), right-click the **Regions** node and select **Create Region**.
3. Select **Help Text** as **Type** in its Property Editor.
4. Enter a **Title** (for example, *Help*) for the help region in the property editor.
5. Enter the following **Help Text**: `This page provides information on any queries related to Project Management.` The Help page with a **Help Text** region is created.
   - Note that to view the Help Text for any page, you must enter the Help Text on that page separately. For example, if any Page 3 has a value entered in "Help Text" under the Help tab in its Properties Pane on the right side, this value will be displayed when the Help Page link is clicked from this page.
6. Save the page and run the application. The page help is displayed.

In the next slide, you will create a **Help** navigation bar entry. Let's see how.



In the preceding slide, Steve has created *Help Text* for the *Help* page in *PTS.* (He has plans to later add help text for all his other pages.) He now adds a *Help* navigation bar entry to help his users. Let's see how.

To create a new navigation bar entry, navigate to the application's **Shared Components** and click **Navigation Bar List** in the Navigation pane. You then perform the following steps:

1. Click **Desktop Navigation Bar** and click **Create Entry**.
2. Enter the following values and click **Create List Entry** by retaining default values for other fields (screenshot 4):
   - **List Entry Label:** Enter `Help`.
   - **Target Type:** Select `Page in this Application`.
   - **Page:** Select `Page 17` from the popup LOV.
   - **Request:** Enter `&APP_PAGE_ID`.
3. Run the page and click the *Help* link in the navigation bar to read its Help Text.

Quiz

Which shared components would you use to create a shared collection of links on a page?

a. Breadcrumbs
b. Lists
c. Navigation bar entries
d. Tabs

Answer: b

Practice14 Overview: Adding Shared Components That Aid Navigation

This practice covers the following topics:

- Creating lists and list regions
- Creating and editing a navigation menu
- Creating a Help page and adding a navigation bar entry
- Adding breadcrumbs to an existing page

Summary

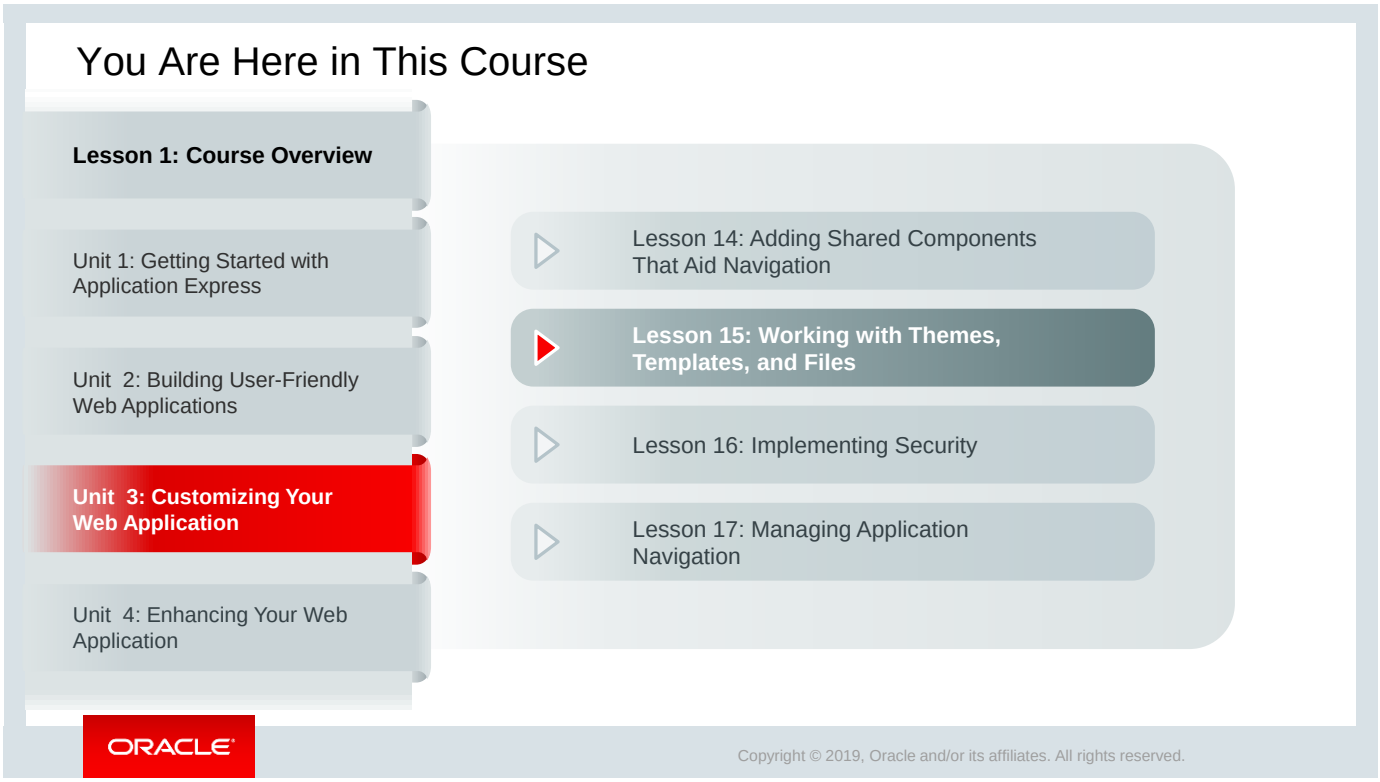
In this lesson, you should have learned how to:

- Provide an overview of shared components
- Include the following shared components in your application:
  - Navigation menu and its entries
  - Lists
  - Breadcrumbs
  - Navigation bar and its entries

In this lesson, you learned how to create, edit, and use navigational shared components in your application.

15

Working with Themes, Templates, and Files



This slide is a graphical depiction of the course, particularly highlighting Unit 3 - Lesson 15, which is dealt with in these slides.

Objectives

After completing this lesson, you should be able to:

- Define themes and their uses
- Create a new theme
- Copy a theme
- Edit a theme
- Switch to a different theme
- Explain Universal Theme and Theme Roller
- Use Theme Roller to change the theme style
- Define templates and their uses
- View existing templates
- Create, copy, edit, and replace a template
- Upload and use a cascading style sheet and an image

This lesson provides an overview of the themes and templates provided by Oracle Application Express.

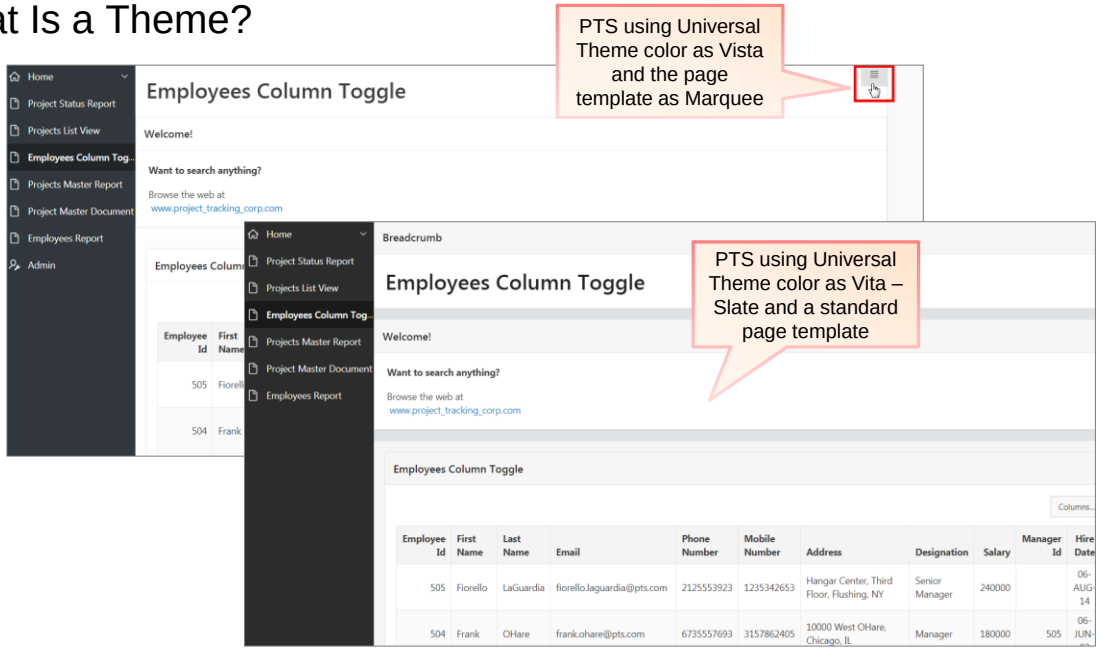
Steve Works with Application User Interface

Now that the functionality part of the PTS application is ready, Steve is looking to work on the look and feel of the application. He explores the various themes available in Oracle Application Express that he can use to enhance the user experience of the PTS application. He wants to customize the style of his PTS application using the Runtime Developer toolbar and Theme Roller. Let us see how he changes the theme style of his application from the Themes page.

Lesson Agenda

- Using Themes
  - What Is a Theme?
  - Accessing the Themes Page
  - Creating a New Theme
  - Creating a Copy of an Existing Theme
  - Editing a Theme
  - Switching Between Themes
  - Changing the Theme Style
  - Accessing the Theme Roller
  - Introducing Universal Theme and Theme Roller
  - Customizing Your Theme Style Using Theme Roller
  - Saving Theme Style in Theme Roller
- Using Templates
- Using Files

What Is a Theme?



The slide shows the *Employee Column Toggle* page of the PTS application using the universal theme, but the theme style is different in the two images in the slide. This theme style is provided by Oracle Application Express and defines an application's user interface, including the reports, buttons, and other controls.

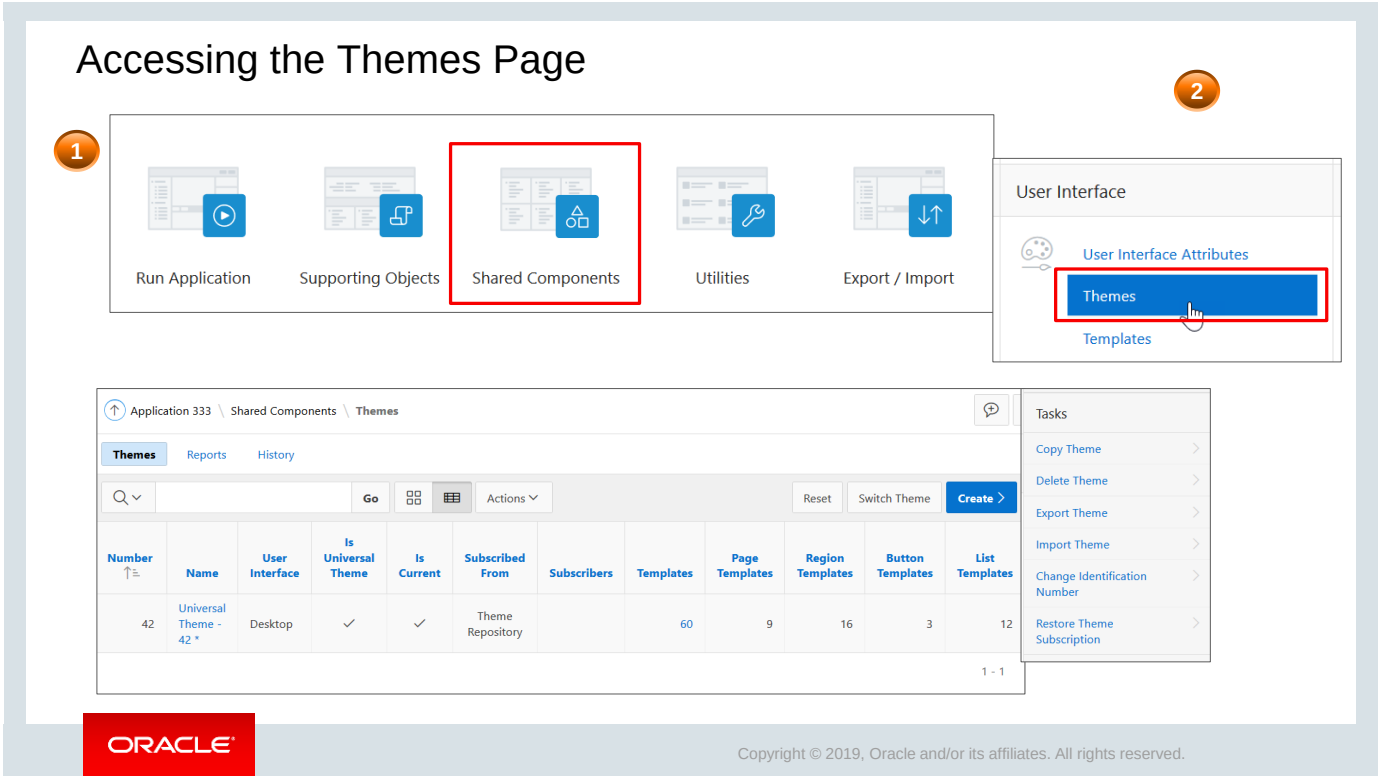
So what is a theme? A theme is a collection of templates that can be used to define the layout and style of an entire application. The purpose of a theme is to provide a complete set of templates that accommodate every user interface (UI) pattern that may be needed in an application. There are two categories of themes:

- **Standard Themes:** Themes supplied with Oracle Application Express
- **Custom Themes:** Additional themes available for use. They can be themes created by workspace administrators for use within a workspace or created by an Instance Administrator, making it available to all developers across all workspaces in that instance.

Oracle Application Express introduced a highly responsive theme called **Universal Theme (Theme 42**), which comes with one or more templates for application components, such as reports, forms, charts, and so on. You can also create a new theme from the beginning and define templates for an application.

Later in this lesson, you will learn how to create a theme from the beginning.

In this lesson, you also learn more about Universal Theme and how to use the themes and templates provided with Oracle Application Express.

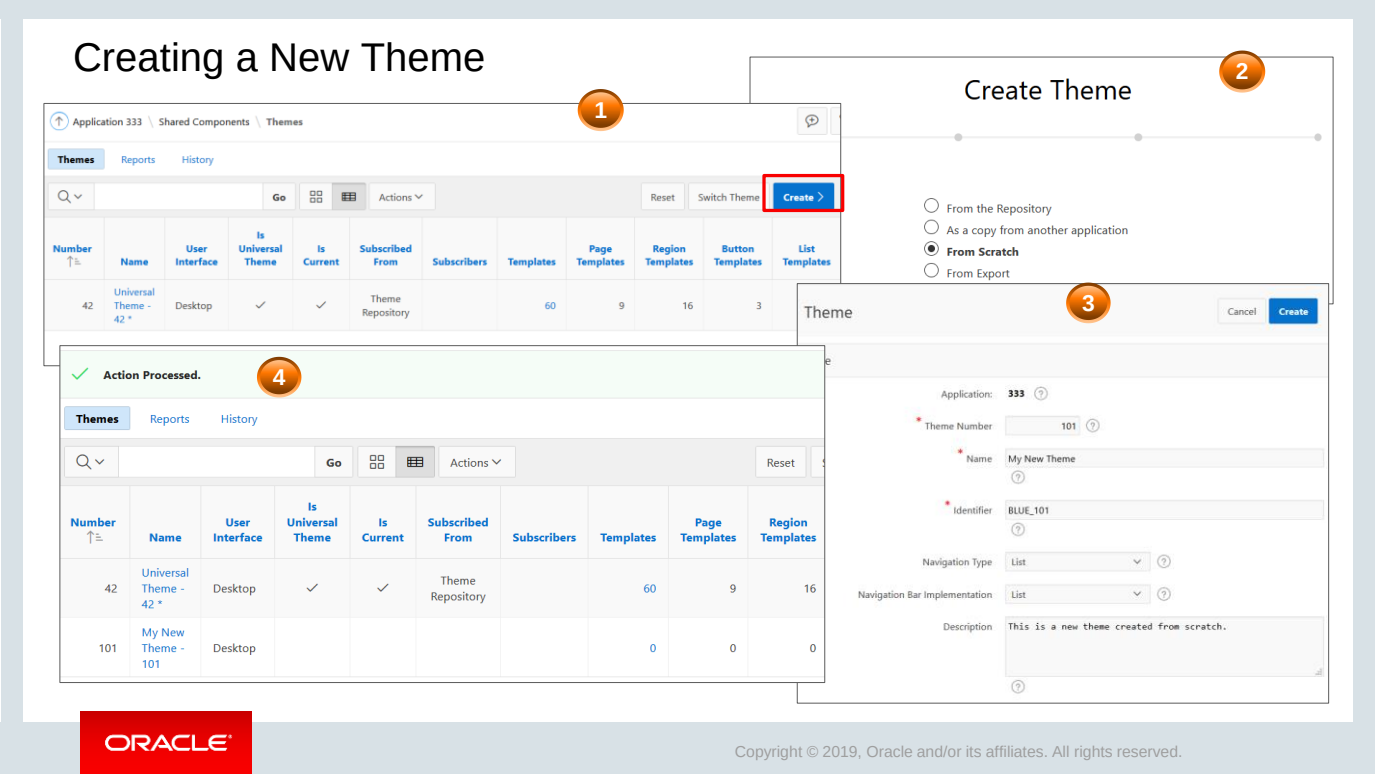


So, how do you access the Themes page?

1. To access the Themes page for an application, click **Shared Components** on the application's home page.
2. Under **User Interface**, click **Themes**.

The Themes page displays the themes available for the application. From the Themes page, you can create a new theme for your application and then switch between those themes. You can also edit a theme, copy a theme, import or export a theme, and so on by selecting the appropriate option from the **Tasks** section.

In the next few slides, you learn how to create, edit, and copy a theme.



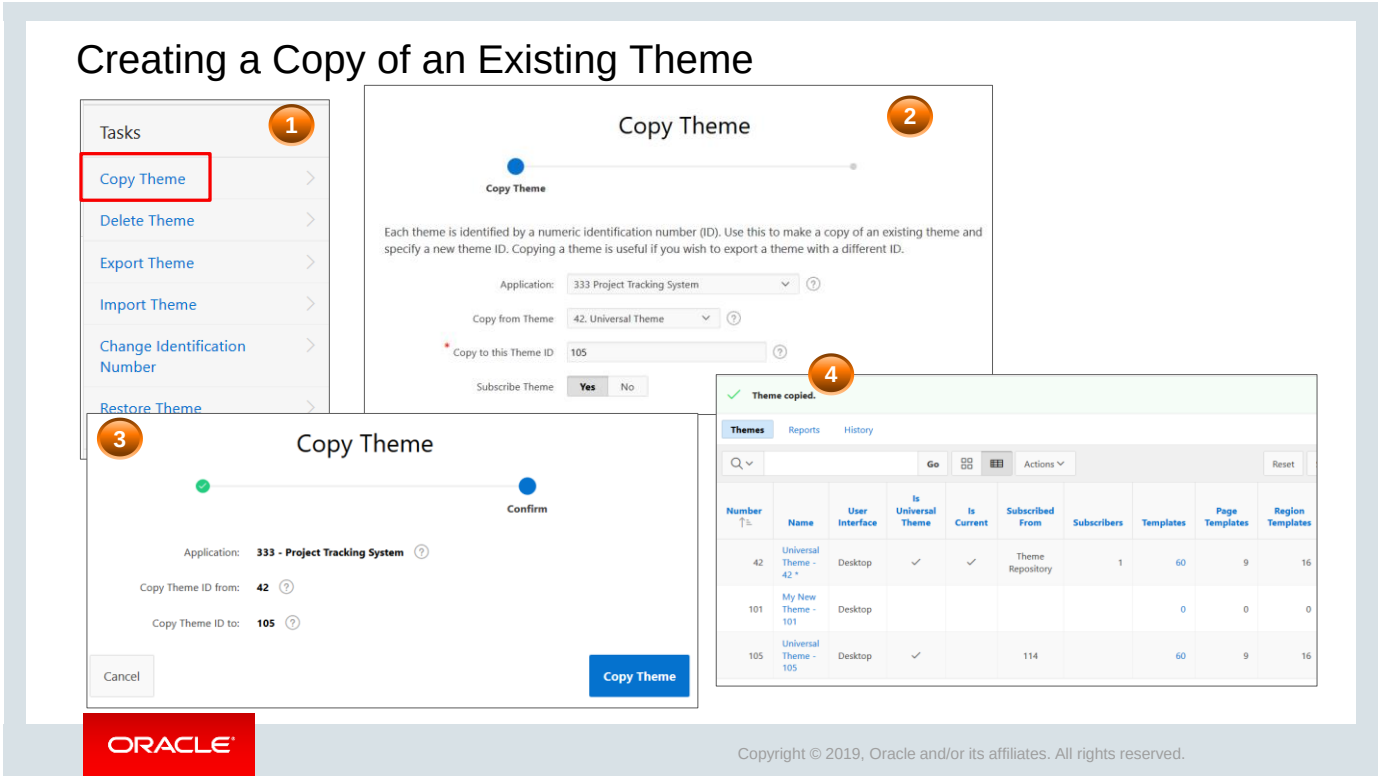
Steve wants to give a new look and feel to his application. He wants to first start creating a theme from the beginning. Let's see how he does it.

You must first open your application and click the **Shared Components** icon. On the Shared Components page, click **Themes** under **User Interface** and then perform the following steps:

1. To create a new theme, click **Create**.
2. Select **From Scratch** and click **Next**.
3. Enter the following values and retain the default values in the remaining fields. Then click **Create**.
   - **Theme Number:** `101` (The theme number is an arbitrary ID and must be unique within your application)
   - **Name**: Enter `My New Theme`.
   - **Identifier***:* Enter `BLUE_101` (the identifier is an arbitrary ID and must be unique within your application).
   - **Navigation Type**: Select **List.**
   - **Navigation Bar Implementation**: Select **List.**
   - **Description:** Enter `This is a new theme created from scratch.`
4. You can see the new theme `My New Theme` added to the Themes list.

Note that after you create a new theme from the beginning, you need to define templates for an application. See "Understanding Template Options" in *Oracle Application Express documentation* (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/understanding-template-options.html#GUID-531DC9F4-0707-45F4-8EA9-5188A7ED99CB) to learn more.

In this lesson, you learn more about Universal Theme and how to use the themes and templates provided with Oracle Application Express.



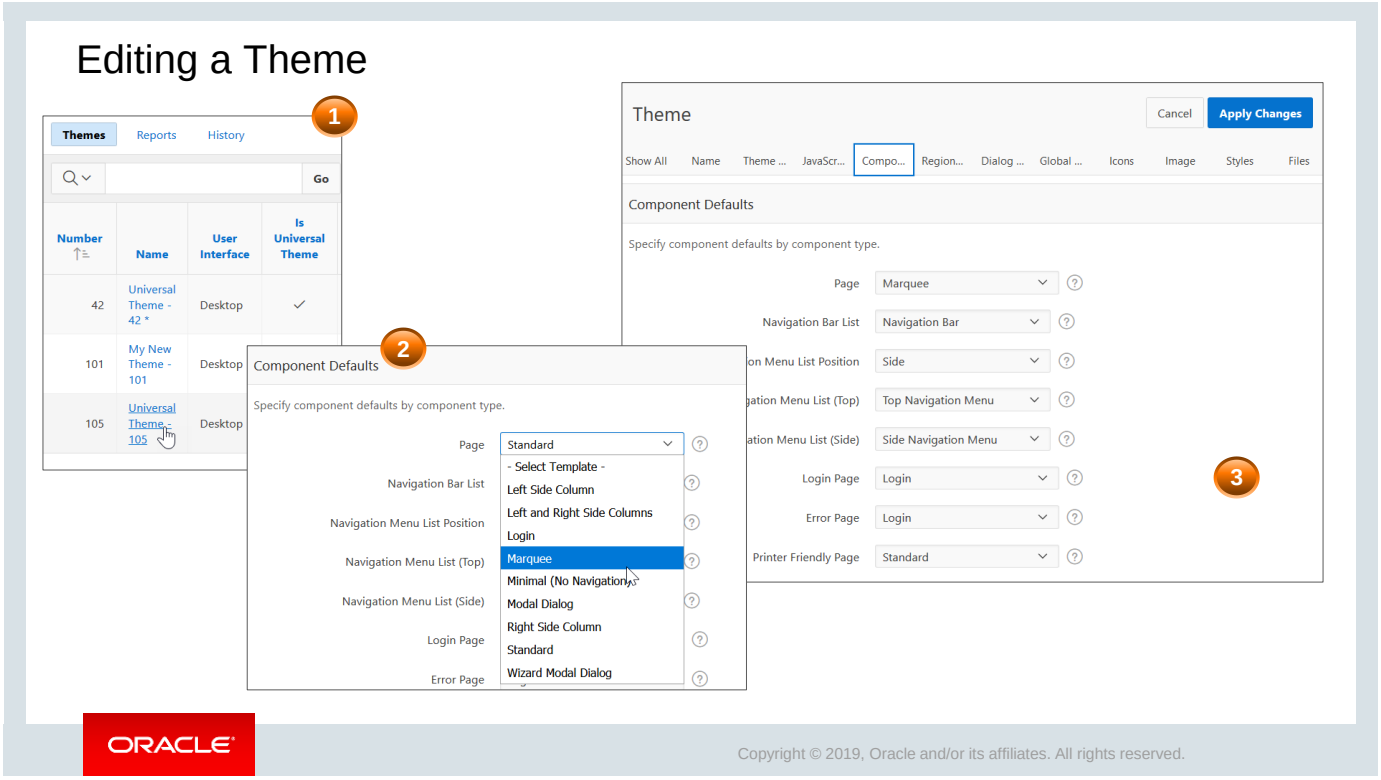
Now, Steve wonders that instead of creating a theme from the beginning, why not copy an existing theme and then make necessary changes to it?

Navigate to the Themes page and perform the following steps.

1. In the Tasks section, click **Copy Theme**.
2. Select the theme that you want to copy and enter a **Theme ID** for the theme. This number must be 100 or greater to indicate that it is a custom theme. Here Steve selects the following values:
   - **Copy from Theme**: Select **42. Universal Theme**.
   - **Copy to this Theme ID**: Enter `105`.

   Now, click **Next**.
3. Click **Copy Theme**.

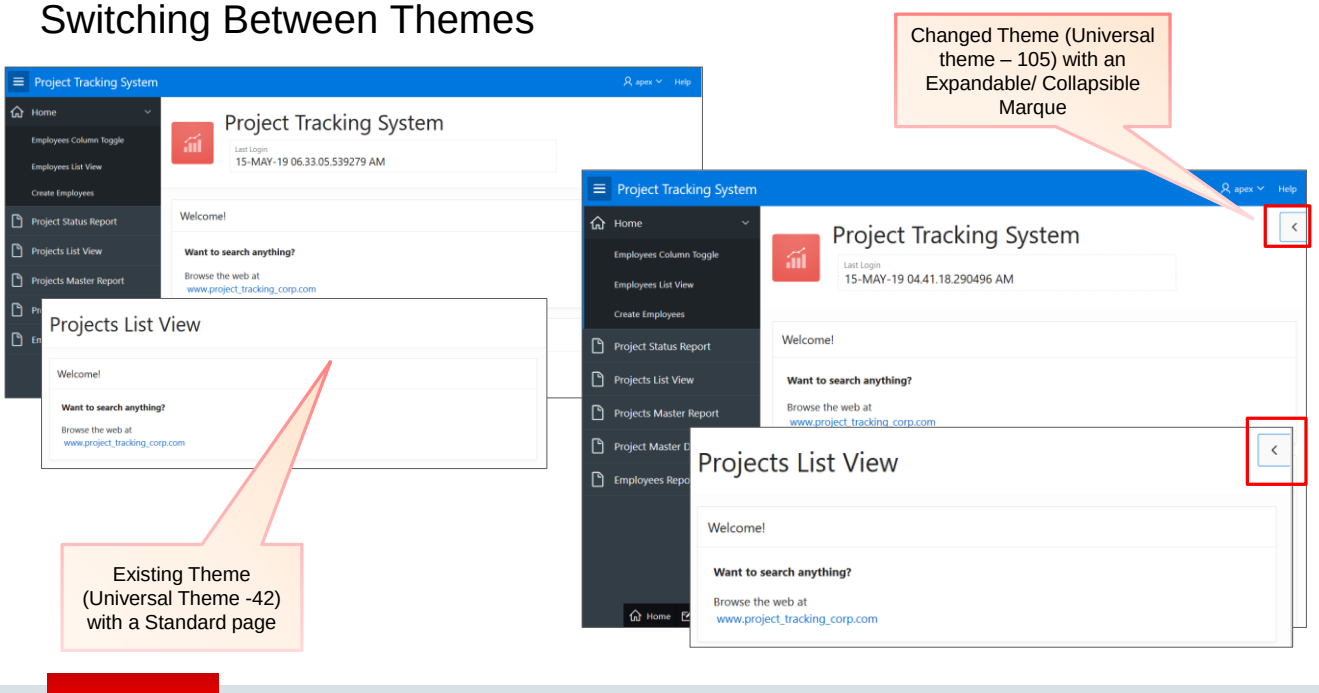
The theme is now copied successfully, and you can make changes to it. In the next slide, we learn how you can edit the theme.



Steve does not want to retain the same *Standard* page layout for his application page. Instead, he wants it to be *Marque*, which creates an expandable and collapsible side column. Let's see how he edits and changes the newly copied theme according to his preference.

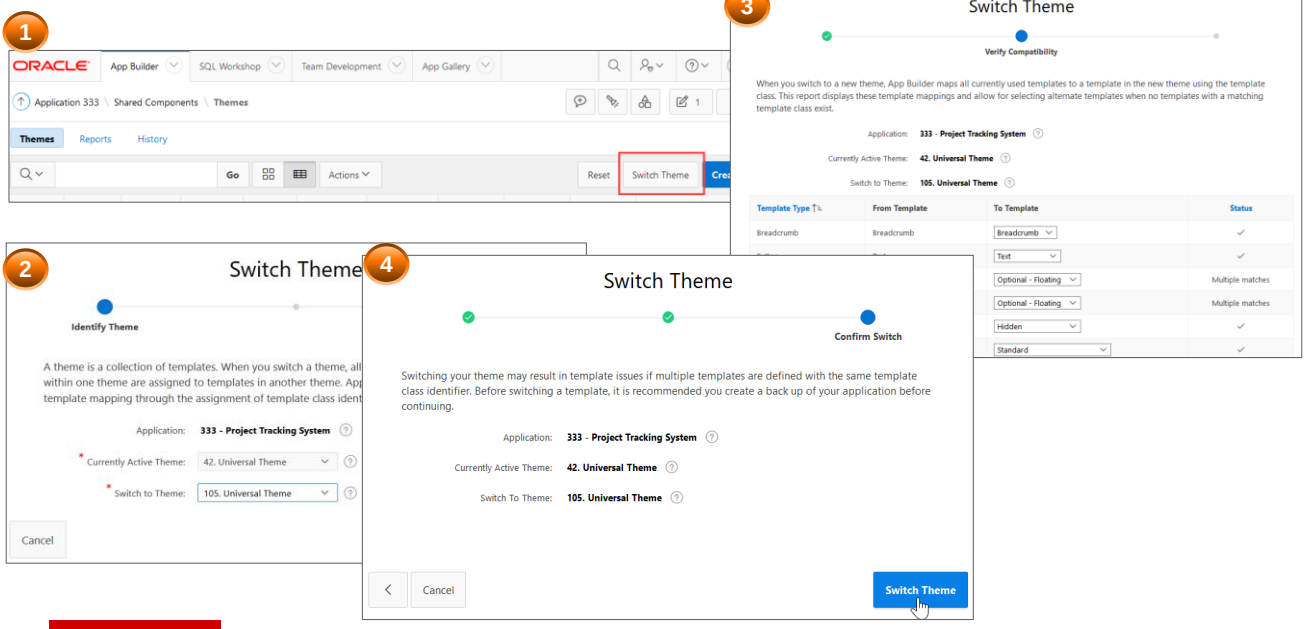
1. On the Themes page, click the theme that you want to edit.
2. The theme properties page opens. You can change the theme properties.
3. Click the appropriate tab and make changes. In this slide, Steve has changed the Page style from **Standard** (used in the Universal Theme – 42) to **Marquee**. Click **Apply Changes** to save your modifications.

Switching Between Themes



Now, Steve wants his application pages to reflect the new style. However, for that he has to switch the current theme to the newly modified theme. Note that you can switch between the themes available for an application (that is, those displayed on the Themes page of an application). When you switch to a new theme, all the components that are assigned a template are assigned to a corresponding template in the new theme. In the next slide, Steve switches his current theme from Universal Theme 42 to Universal Theme 105 (which he copied and edited in the previous slides).

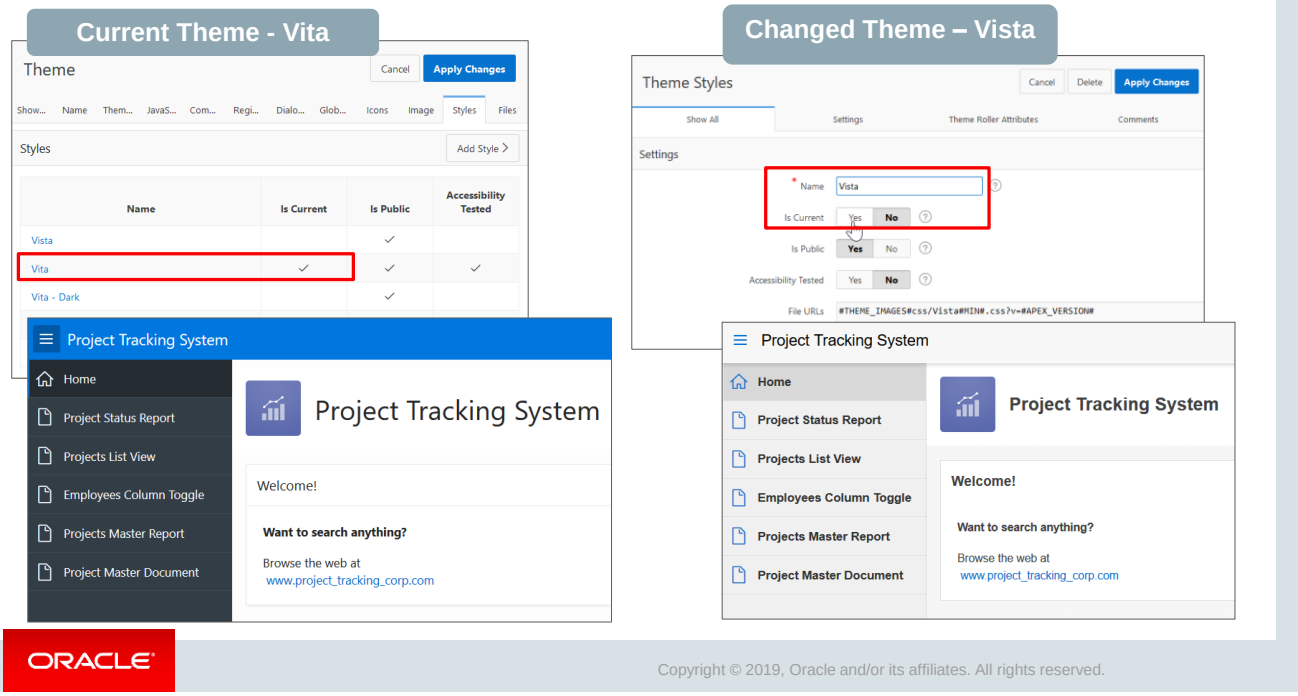
Example: Switching Between Themes



Click the **Switch Theme** button on the Themes page and perform the following steps:

1. On the Themes page, click the **Switch Theme** button.
2. Select the currently active theme and the theme to switch to from the select list and click **Next**.
3. Review the compatibility status report and click **Next**.
   - A check mark in the **Status** column indicates that the mapping was successful.
   - A warning indicates that there is more than one template in the theme you are switching to with the identified class. The warning provides a select list from which to choose the appropriate template.
   - An error indicates that Application Builder was unable to map the class between the themes. Ensure that a class is identified for the templates in both themes.
4. Click **Switch Theme**. The new theme becomes the current theme, and all the application pages reflect the changed current theme.

Changing the Theme Style



Can you see the difference in the theme style in the two images in the slide? You know that Steve has copied a theme from an existing theme, and now he wants to customize the theme style according to his preference. So, what is a *Theme Style*? A theme style defines a Cascading Style Sheet (CSS) that is added to the base CSS to alter the look and feel of an application. (You will learn more about how to create and apply CSS later in this lesson.)

Here Steve uses theme styles to switch to a different color scheme. Let's see how he changes the theme style of an application that is already available in Oracle Application Express, from **Vita** to **Vista**:

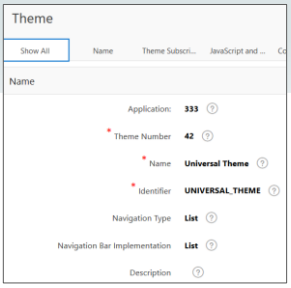
1. Open your application. (Here Steve selects *PTS.*)
2. On the Oracle Application Express home page, click **App Builder**.
3. Click **Shared Components**.
4. Under User Interface, click **Themes**.
5. Click **Universal Theme - 42 ***.
6. Click the Styles tab and select **Vista**.
7. Click the Settings tab and select **Yes** in the **Is Current** field.
8. Click **Apply Changes** to save the changes.
9. Click the Run Page icon in the upper-right corner on the Themes page. A rendered version of page appears. Your application now has a new theme style called **Vista**.

Note that you can also change the Theme style of your application at run time by using the **Theme Roller**. Let's learn more about **Universal Theme** and **Theme Roller** in the next slide.

Using Universal Theme and Theme Roller

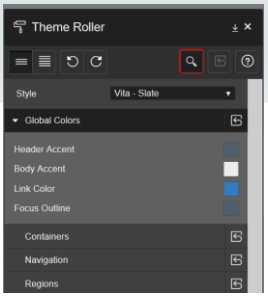
Universal Theme

- Builds a highly responsive user interface (UI)
- Is completely list based, does not support tabs
- Offers in-built navigation menu with option to add new entries
- Supports Theme Roller: A magic wand in a developer's hand
- Is inherently simple with lesser Template Options



Theme Roller

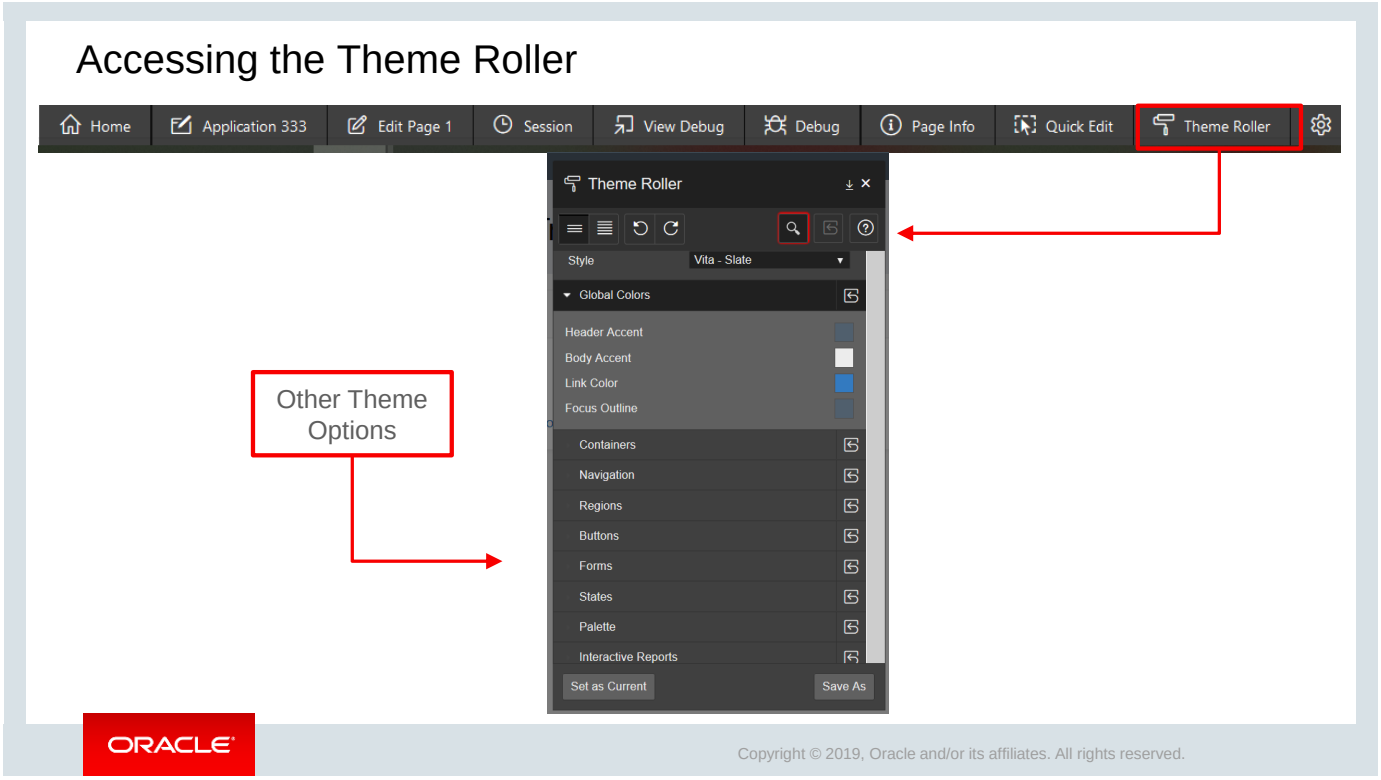
- Allows developers to explore theme colors, fonts, and theme layouts
- Offers easy customization of UI without getting into CSS, HTML, or JavaScript
- Provides scope to completely change the look and feel of UI
- Enables saving of private themes



Universal Theme is a responsive, versatile, and customizable user interface for your Application Express applications. It is designed uniquely for Oracle Application Express to make it easy for developers to build beautiful, modern applications. It supports **Theme Roller**, which is a live CSS editor. The developers can use this function at run time to quickly change the colors, rounded corners, and other attributes of their applications without touching a line of code.

Note that the Theme Roller option is displayed in the Runtime Developer toolbar only if theme styles have been defined.

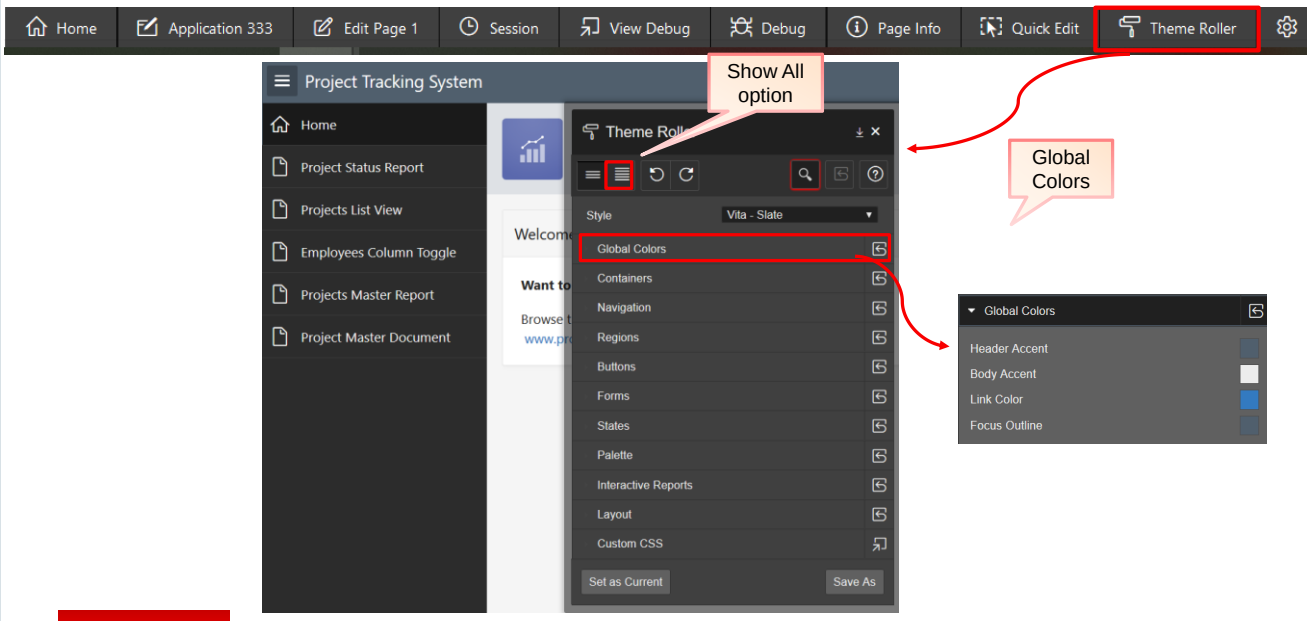
In the next few slides, you will learn how to access and use the **Theme Roller** to change the appearance of your application.



So, if Steve wants to edit the theme style of his application at run time, the **Theme Roller** is his obvious choice. To use Theme Roller:

1. Preview the page by running it. When a developer runs a desktop application, the Runtime Developer toolbar displays at the bottom of any editable running page.
2. Click **Theme Roller** on the Runtime Developer toolbar. Theme Roller fetches the styles for your application and loads them in the editor.

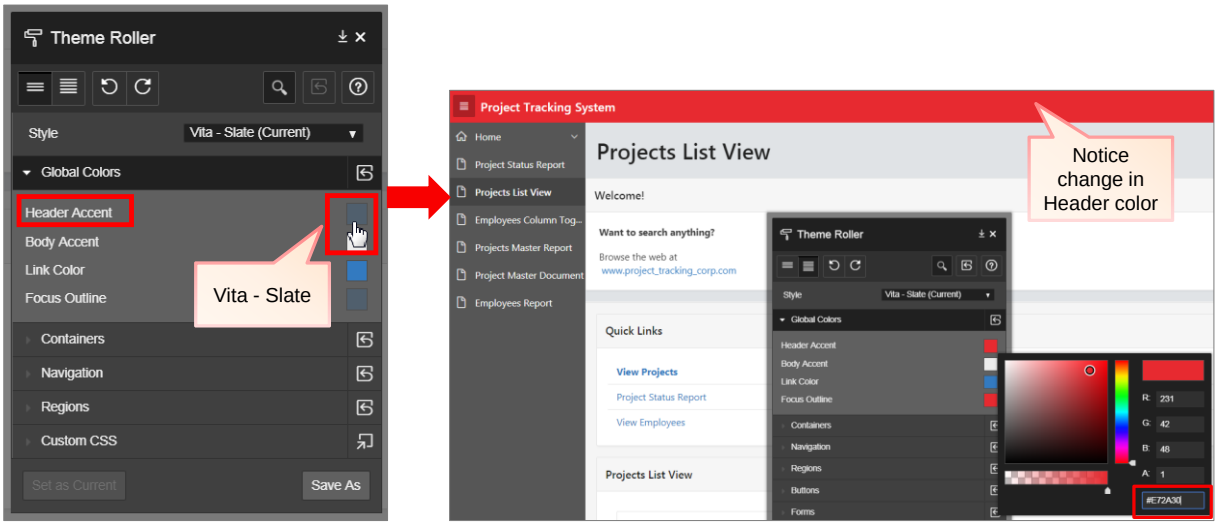
Customizing Your Theme Style Using Theme Roller



Steve now starts exploring the functionality of the Theme Roller to customize the theme style and change the appearance of the pages in his application. He starts with the *Projects List View* page. To customize your theme style:

1. Click **Theme Roller** in the Runtime Developer toolbar.
2. Click **Show All** to edit the attributes. A Theme Roller editor appears, showing the current theme style settings. Note that **Global Colors** region is expanded by default.

Customizing Your Theme Style Using Theme Roller – Changing Header

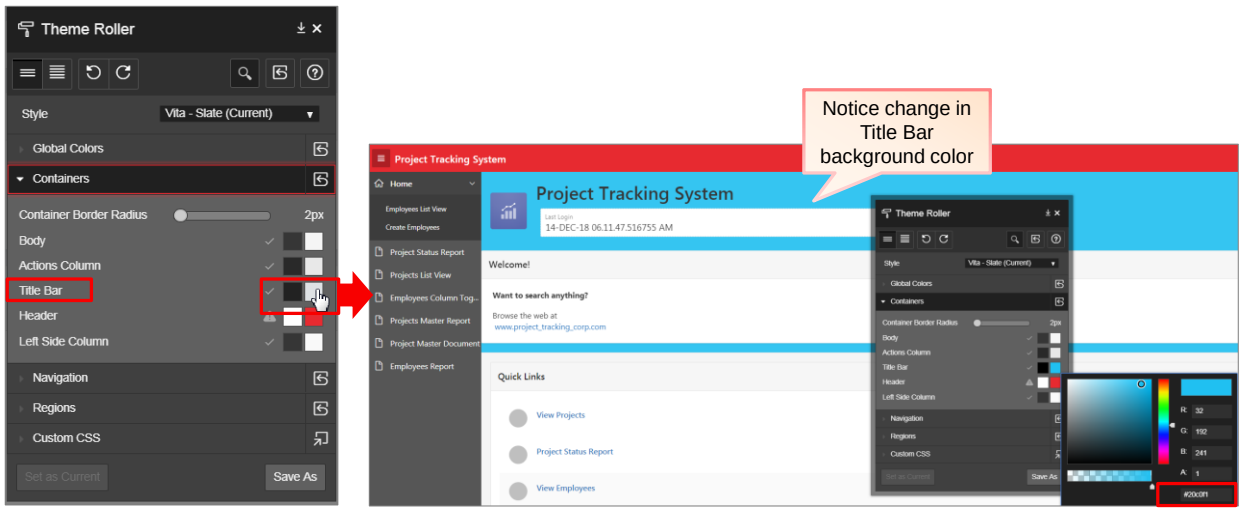


Remember Steve selected **Vita – Slate** as the Theme style color a couple of slides back? The **Header Accent** under **Global Colors** shows you the same theme color. Now, he tries changing the appearance of his application. Let's see how.

1. Click **Header Accent** color under Global Colors.
2. Change the value to **#E72A30**. You can also directly select red color by clicking the color chart.

In the next slide, he changes the background color.

Customizing Your Theme Style Using Theme Roller – Changing Title Bar Background

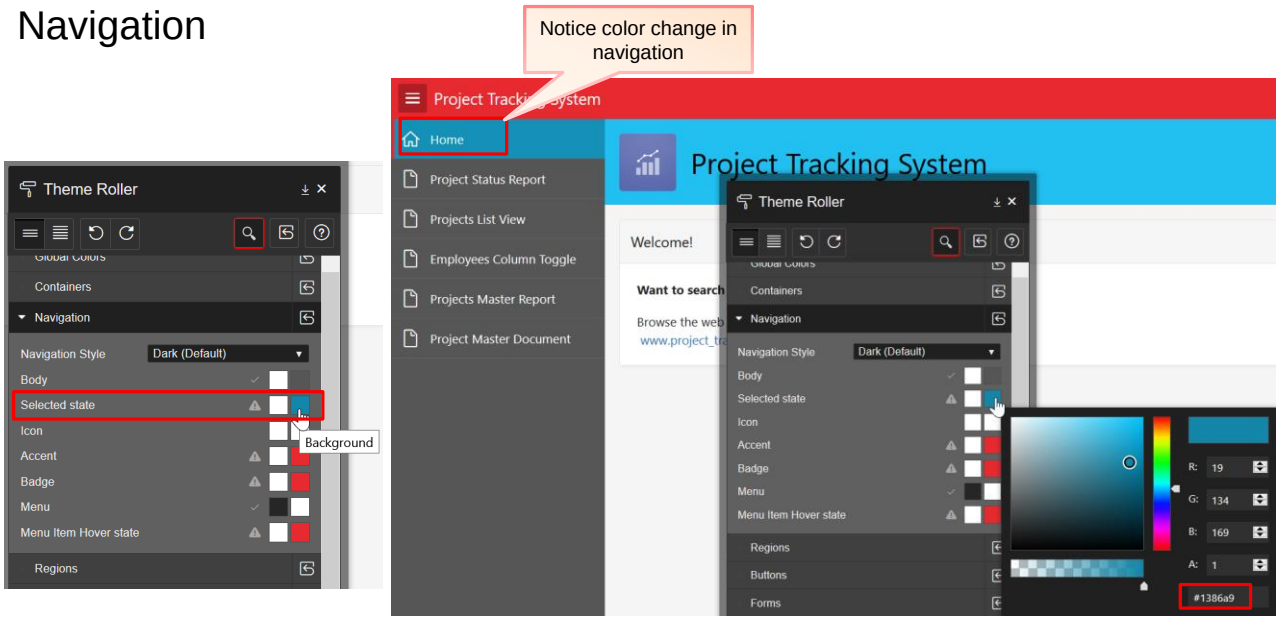


Now, let's update the background color of the **Title Bar**.

1. Expand **Containers**.
2. Click the Background option next to **Title Bar**.
3. Enter the value, **#20C0F1**. Notice the color change in the Title Bar background.

In the next slide, Steve updates the navigation colors. Let's see how.

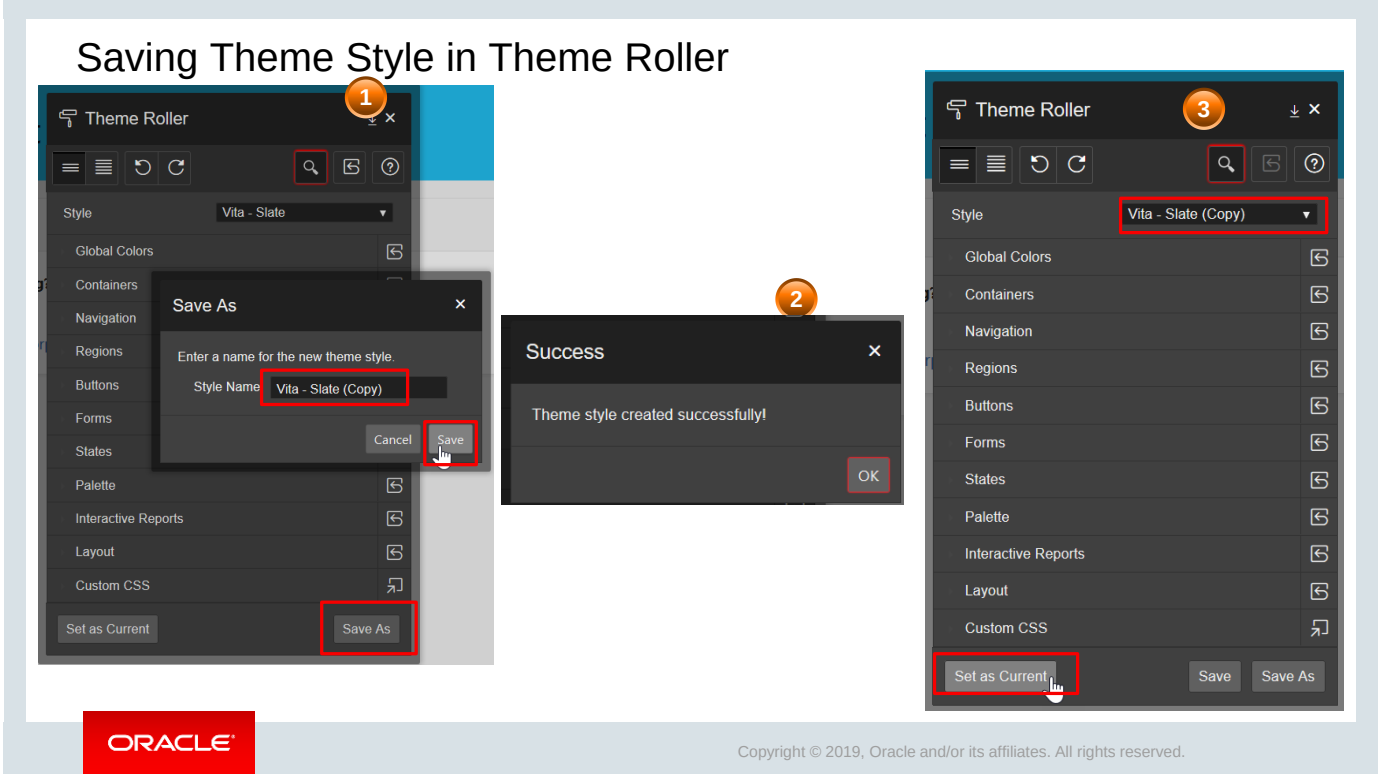
Customizing Your Theme Style Using Theme Roller – Updating Navigation



Let us now update the navigation.

1. Expand **Navigation**.
2. For Selected state, click **Background**.
3. Enter the value, **#1386A9**.

Notice the color change for the selected item in the navigation pane.



Now that Steve has changed his theme style, he wants to set the style as the current theme for his application. Let's see how.

1. Click **Save As** to save the changes.
2. In the **Save As** dialog box, for **Style Name**, enter `New Theme`.
3. Click **Save**. A success dialog box appears. Click **OK** to exit the dialog box.
4. A dialog box appears to set your current theme. Click **Set as Current** to apply the custom theme for your application. A success dialog box appears. Click **OK**.

Note that you can use the Theme Roller to change the current custom theme style, for example, the header, background, or navigation colors, back to what it was/or any other color any time you want.

Quiz

Which of the following statements are true about themes? (Choose all that apply.)

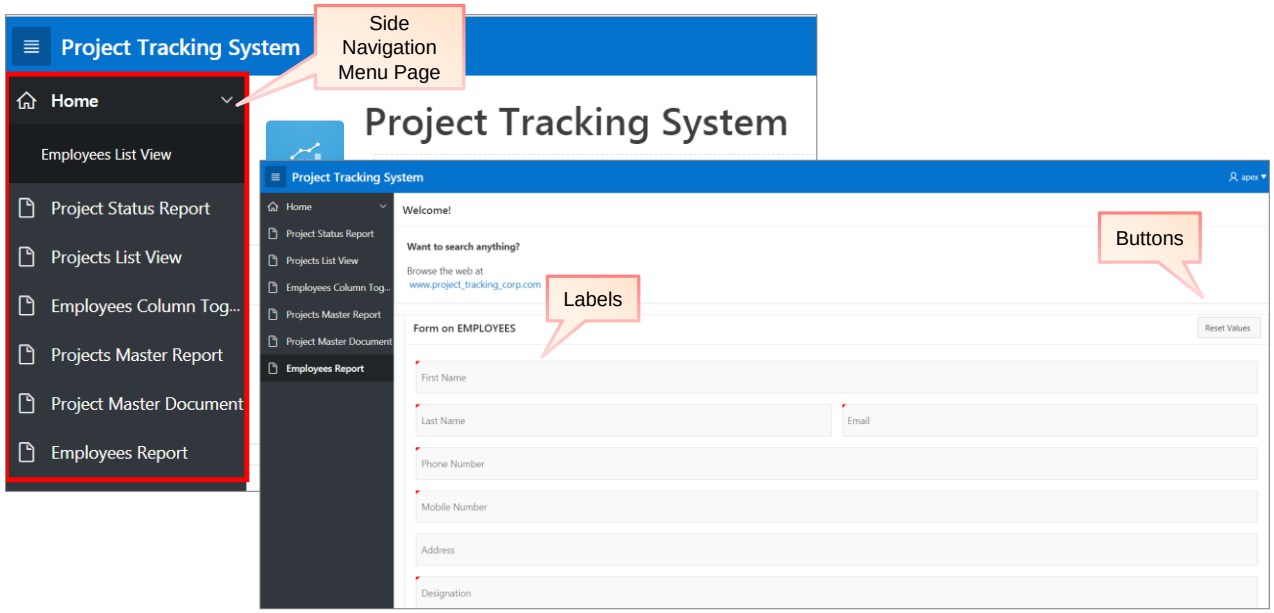
a. Workspace themes are available to all developers in the workspace.
b. You can switch from Universal Theme to a Simple Red theme.
c. When you switch to a new theme, all the components that are assigned to a template are assigned to a corresponding template in the new theme.
d. You can copy an existing theme and make changes to the copy.

Answer: a, c, d

Lesson Agenda

- Using Themes
- Using Templates
  - What Are Templates?
  - Types of Templates
  - Accessing the Templates Page
  - Replacing a Template
  - Creating a Copy of an Existing Template
  - Editing a Template
  - Applying a Template
  - Changing the Default Templates for a Theme
  - Overriding Application Defaults at the Page Level
  - Using Substitution Strings in Templates
- Working with Files

What Are Templates?



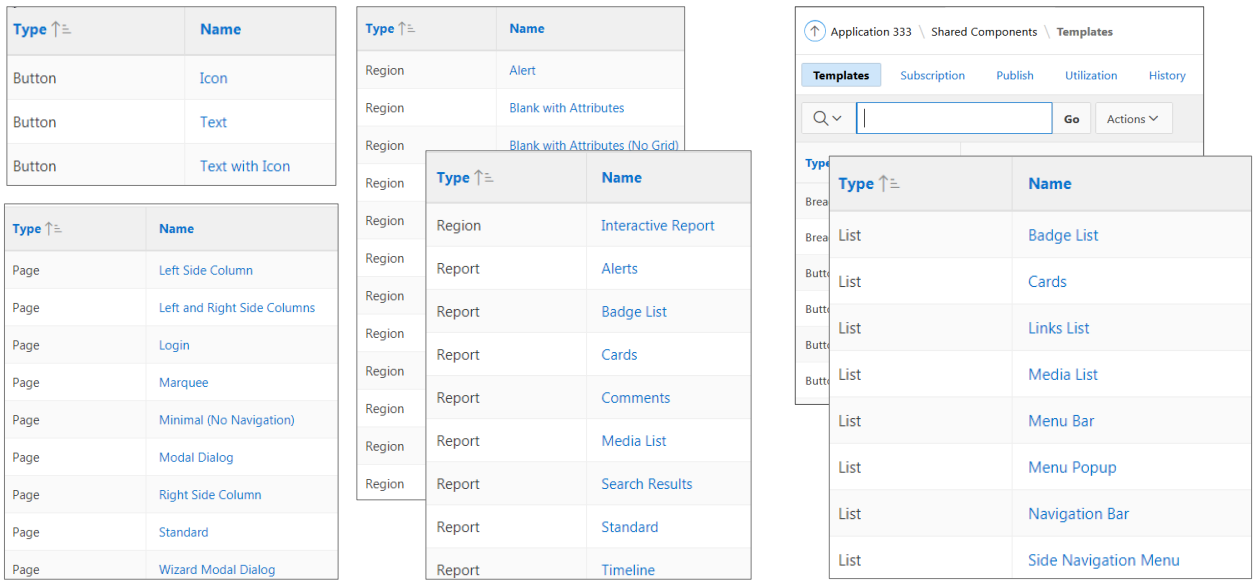
The slide shows an example of a page and the various types of templates associated with the page (for example, Button or Labels region templates used in the Form) and its components. The templates used on a page can be accessed from the Shared Components region of the page definition. Templates define how the pages or the page components of an application are displayed. You can select templates for your page or page components from the templates available in the application's theme. Alternatively, you can customize the look and feel of the application by modifying the existing templates or creating new templates using HTML and cascading style sheets (CSS).

Templates facilitate the separation of business logic from the user interface. You can focus on the code for the business logic, whereas the graphic artists can concentrate on the look and feel. The advantages of using templates are as follows:

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

Let's learn about the different types of templates in the next slide.

Types of Templates

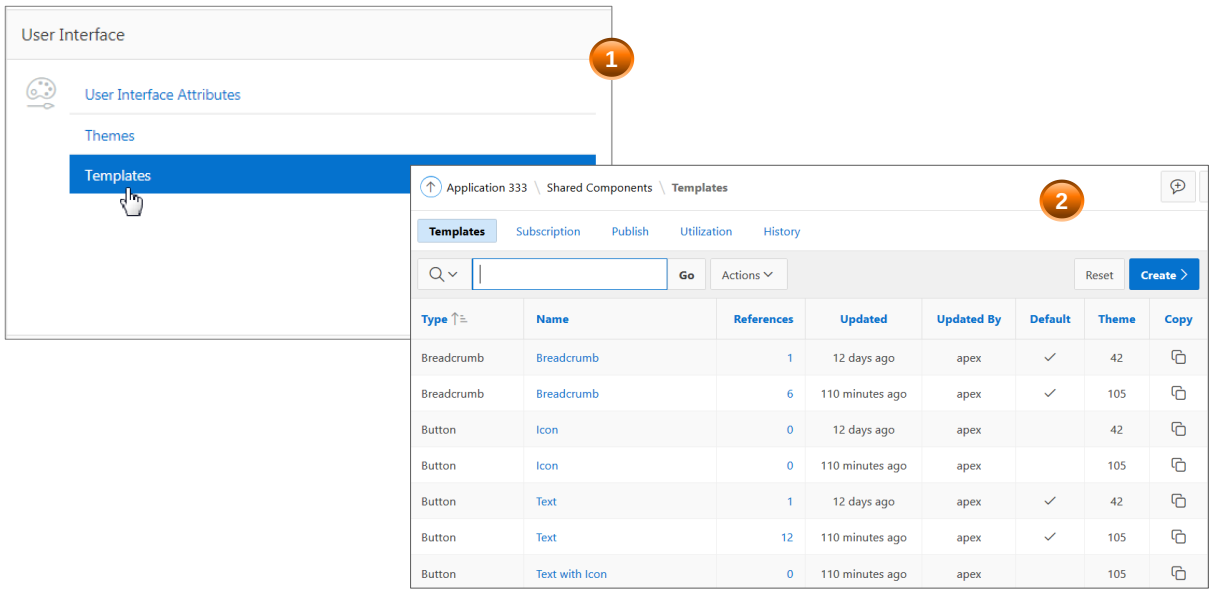


In the previous slide, you saw the label and button regions template associated with the form template class. Templates are first organized by template type. Oracle Application Express offers nine types of templates. Each theme comes with one or more template classes for each template type. For example, a region template can be classified as a form region template, a report region template, and so on. Here region template is the template type, whereas form region and report region are template classes.

This slide shows some of the templates available for the Page, Report, Region, and Label types. *Page* templates control the appearance of the navigation menu, master detail, modal dialog, and the page layout. *Region* templates control the display of region titles, buttons, and so on. *Report* templates control the format of the displayed report. The *Label*, *List*, *Popup*, *Calendar*, *Breadcrumb*, and *Button* templates specify how those respective components should be displayed.

You will learn more about how to access templates page and how to replace, copy, and edit templates in the next few slides.

Accessing the Templates Page



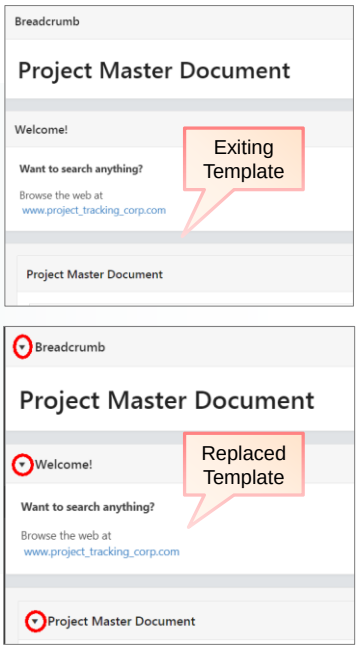
To view the Templates page, navigate to the **Shared Components** page of the application.

Under User Interface, select **Templates**. The Templates page appears. You can view the default templates and the referenced templates.

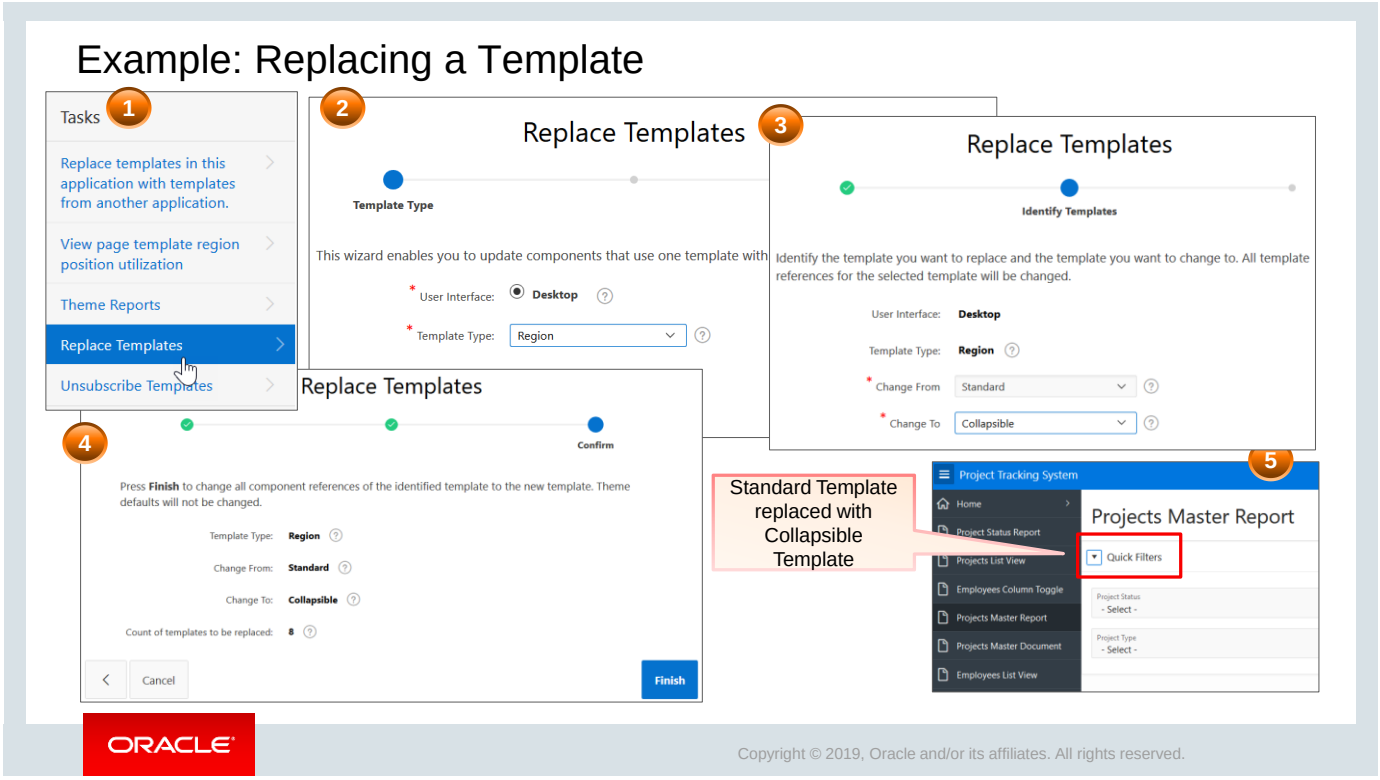
If you are not satisfied with the default template (for example, for a region), you can replace it with the template you want. Steve does exactly that in the next couple of slides. Let's see how.

Replacing a Template

1. Select the application and click **Shared Components**.
2. Under User Interface, select **Templates**.
3. For Template Type:
   a) User Interface - Select the **User Interface**. (Here you select **Desktop**.)
   b) Template Type - Identify the template type to be replaced.
   c) Click **Next**.
4. From the Task list, select **Replace Templates**.
   a) Change From - Select the template you want to change.
   b) Change To - Select the template you want to change to.
   c) Click **Next**.
5. Click **Finish**.



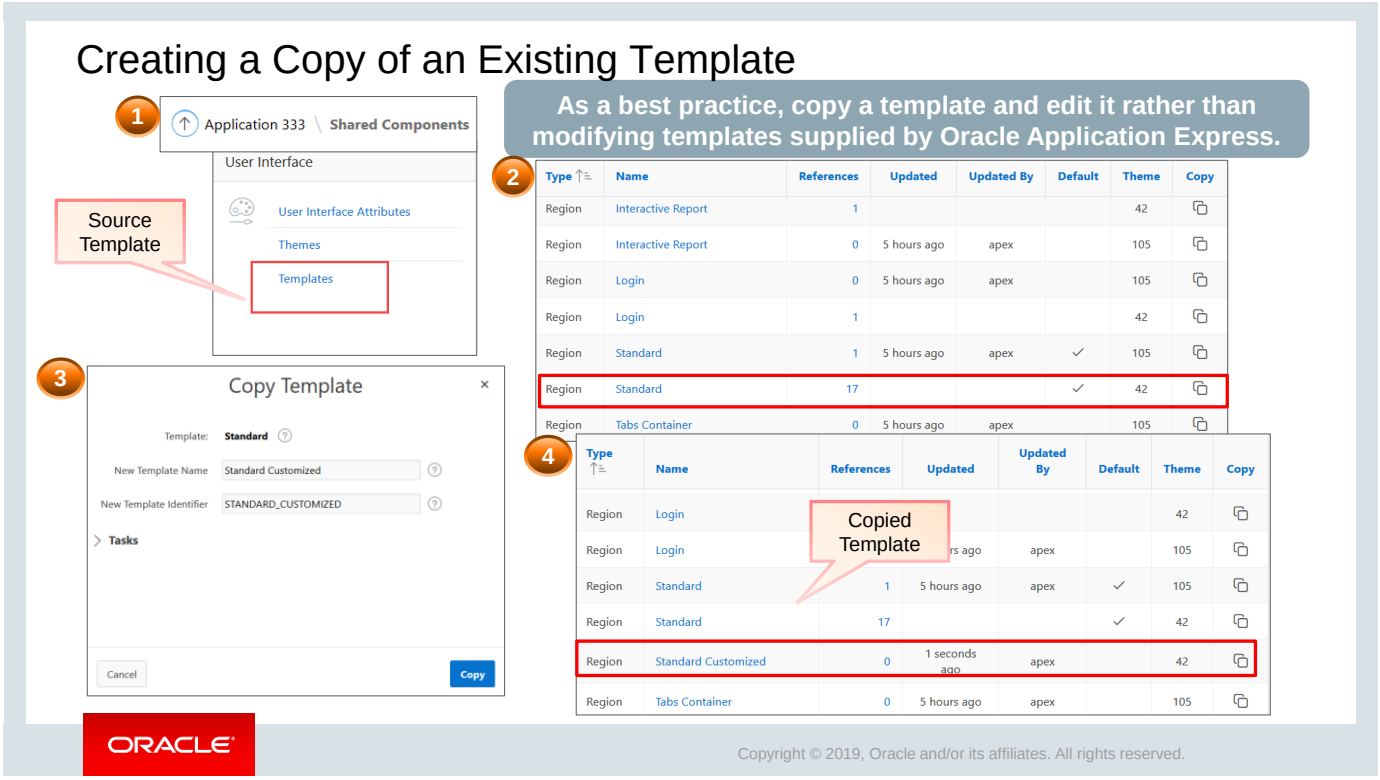
Now, Steve has a requirement that he does not want to use the standard region template that his PTS application is currently using. Instead, he wants the regions to be collapsible. This way the page will look less crowded, and he can expand and collapse the regions as and when he wants. You will learn how he does it in the following slide.



Let's see how Steve replaces a template for the **Region** Template Type.

You must first select the application and then click **Shared Components**. On the Shared Components page, select Templates under **User Interface** and then perform the following steps:

1. From the Task list, select **Replace Templates**.
2. For **Template Type**:
   - **User Interface** - Select the user interface. Here you select **Desktop**.
   - **Template Type** - Identify the template type to be replaced. Here you select **Region**.
   - Click **Next**.
3. For **Replace Templates**:
   - **Change From** - Select the template (here you select **Standard**) you want to change.
   - **Change To** - Select the template (here you select **Collapsible**) you want to change to.
   - Click **Next**.
4. Click **Finish**. Run the page to see the change in the region display of your application.



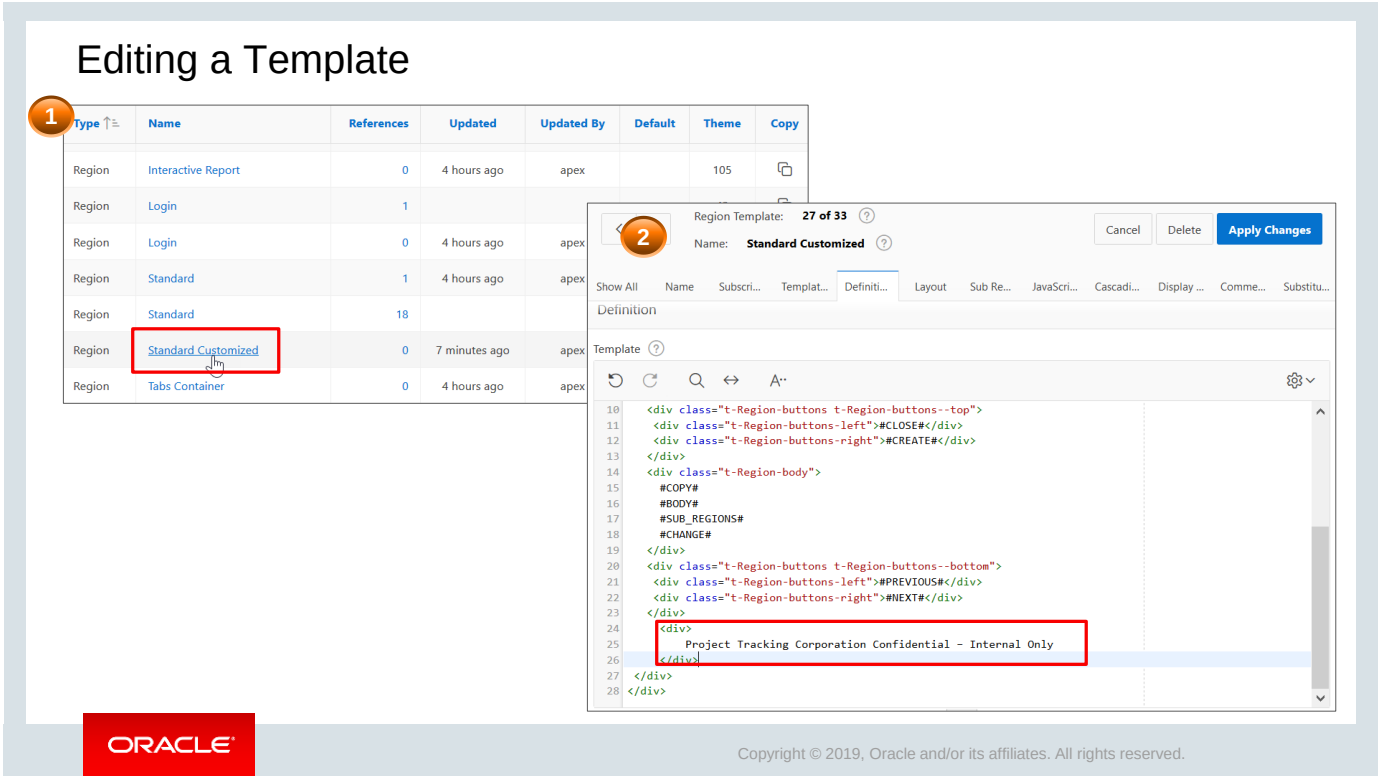
Now, if you want to change one or a few of the templates supplied by Oracle Application Express, it is best to copy the template to another name and then modify the copied template. You can then associate the copied template with the desired page. It is better to copy a template so that you always have the original template to go back to or use in a different application.

To copy a template, perform the following steps:

1. On the Shared Components page, select **Templates** under User Interface.
2. On the Templates page, click the **Copy** icon for the template that you want to copy. In this example, you select **Region** template.
2. Enter **New Template Name** (for example, *Standard Customized*). The **New Template Identifier** automatically populates the name that you entered for the new template.
3. Click **Copy**. The copied template appears in the template list. In the slide example, you created a copy of the **Region** template.

Note: If you want to create a new template for use in your application, click the Copy Template icon for any template that can be found from the current theme's template list.

Steve finds this feature very helpful; however, although he likes to use this template for his application, he also wants to add some company-specific text into it. Let's see how he edits this template to match his requirement.



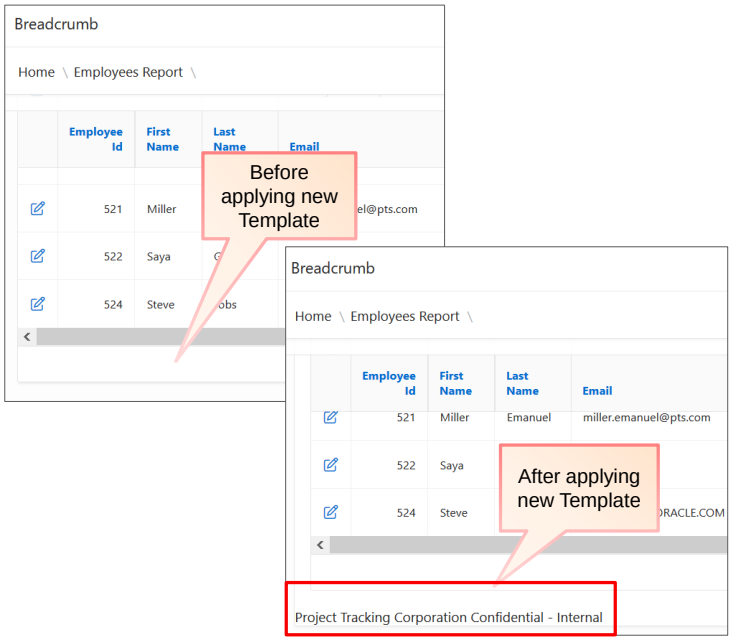
Steve now wants to add some company-specific text or style into the copied template by performing the following steps:

1. On the Templates page, click the name of the template to modify. Here he selects *Standard Customized*.
2. Modify the definition of the template and click **Apply Changes**.
   In this example, he adds the text `Project Tracking Corporation Confidential - Internal Only` at the bottom of the page.

Note: You cannot edit any templates provided with Application Express. However, you can modify the templates that are created by copying from an existing template.

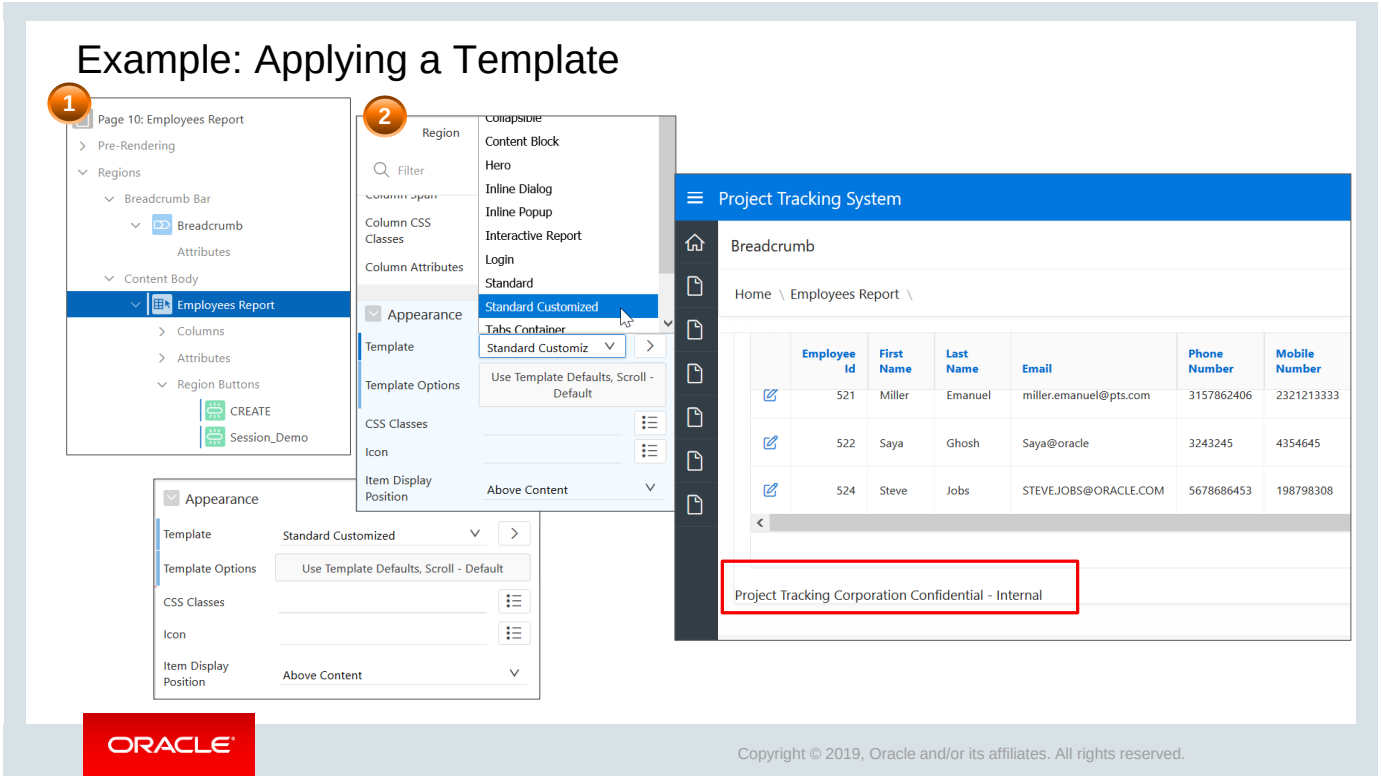
Applying a Template

1. Open the page where you want to apply the new template in Page Designer view.
2. For the region to which you want to apply a new template, click the Region node under page rendering tree.
3. In the property editor, locate Appearance.
4. Select the new template from the Template drop-down list and click the “Save and Run” icon.



Now that Steve has edited the template and added the company-specific text into the copied template definition, he has to apply it to the region where he wants it to reflect in his application. He chooses the *Employees Report* to display this text.

You will learn how he does it in the next slide.

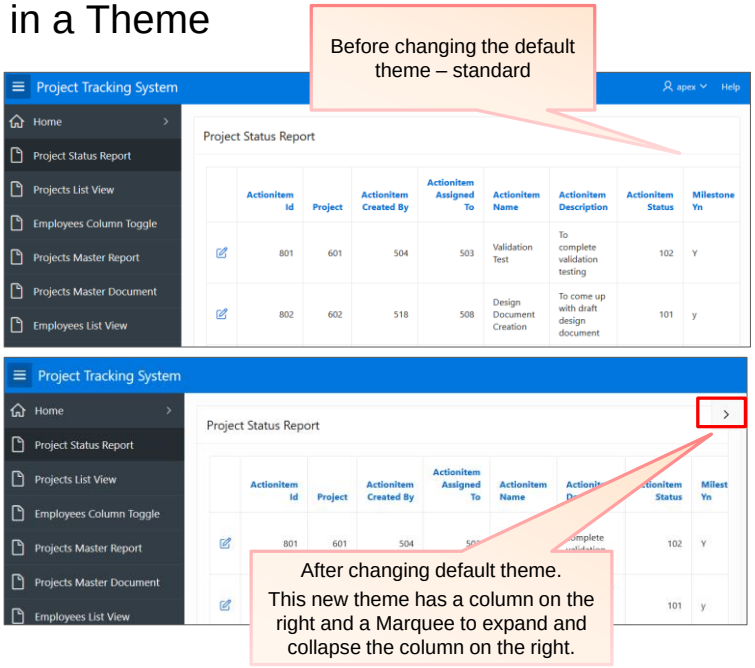


Let's see how Steve applies the edited template to the *Employees Report* region:

1. Open the *Employees Report* in the Page Designer view. Click the **Region** node under page rendering tree. Here, he selects **Report 1**.
2. Locate **Appearance** in the property editor. Select the new template from the Template drop-down list. Here he selects *Standard Customized*. He had copied and edited this template earlier.
3. Click the **Save and Run** icon. The *Employees Report* displays the changed template definition.

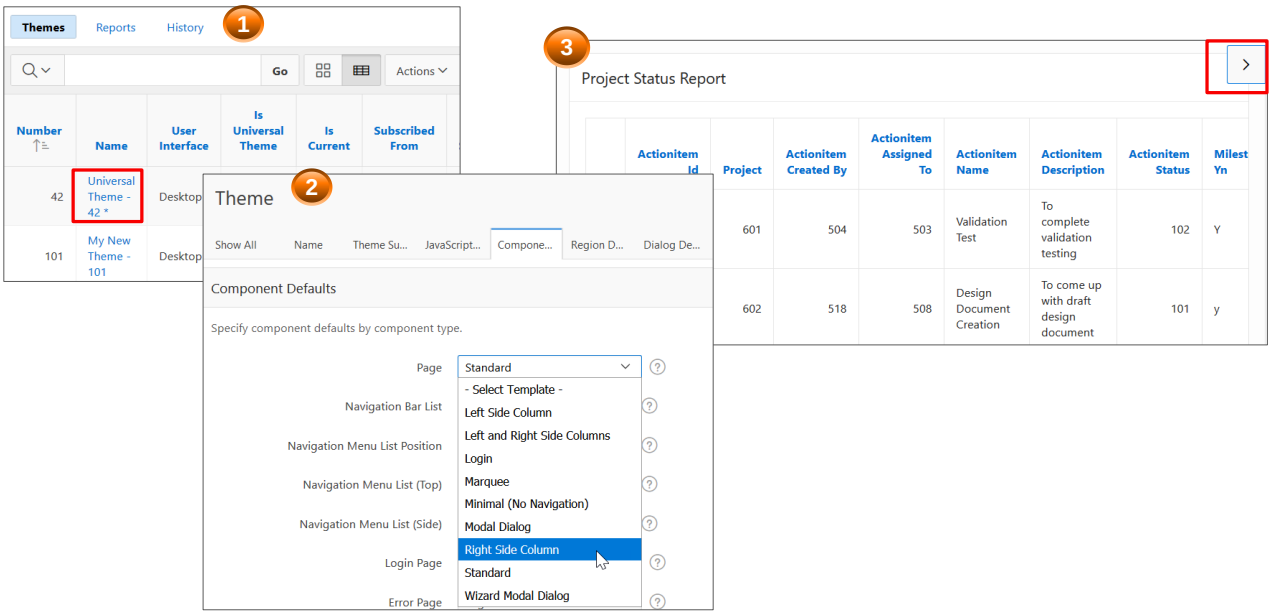
Changing Default Templates in a Theme

1. Navigate to application's **Shared Components.**
2. Click **Themes** under **User Interface**.
3. Click the name of the theme that you want to edit.
4. Click the **Component Defaults** tab and change the template defaults as required.



Steve is not too pleased with the standard default theme style and, therefore, wants to change his default theme to give his pages a new look and feel. Interestingly, he can change the default templates for each type of template in a theme, for example, page, navigation menu, navigation bar, and so on. He wants to start with changing the default standard theme of his page template. In the next slide, let's see how.

Example: Changing Default Templates in a Theme

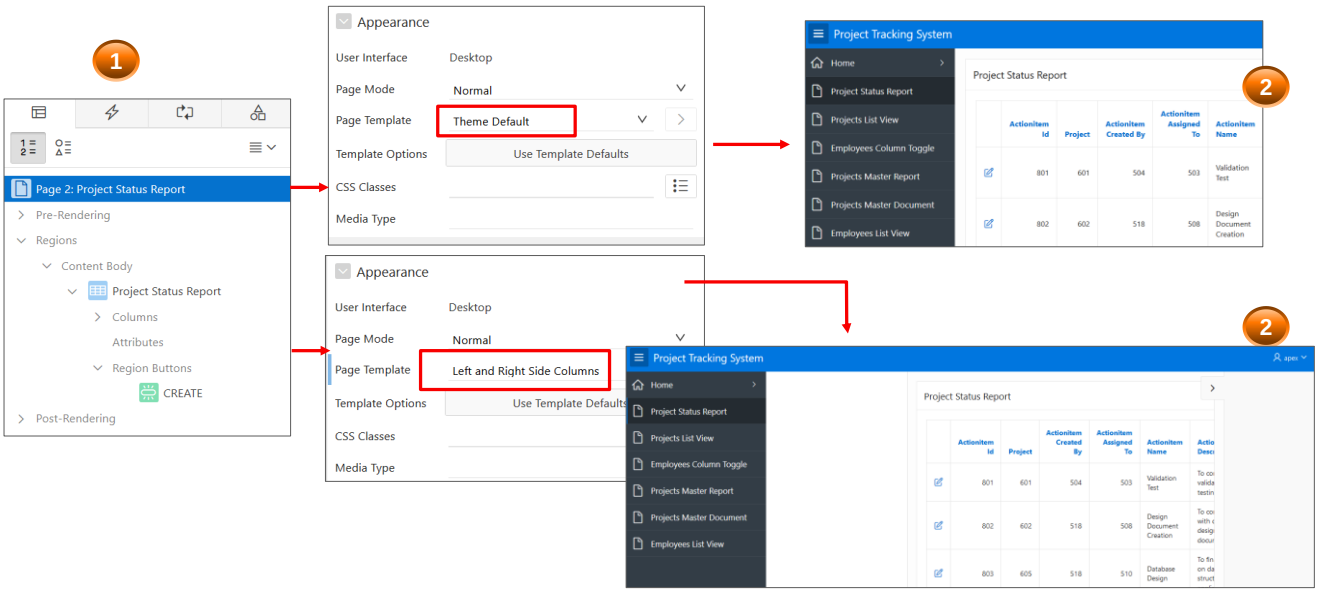


To change the default template in a theme, navigate to the application's shared components and click **Themes** under User Interface.

1. Click the name of the theme that you want to edit. Here Steve selects **Universal Theme – 42***.
2. Click the **Component Defaults** tab. Because he wants to change the default theme *Standard* for his Page component type, he selects *Right Side Column* (to give it a new look and feel, or if he wants, he can later create some quick links too in that area). Note that you can also change a region's defaults on the **Region Defaults** tab. Click **Apply Changes**.

The new default page template reflects in all the pages of your application. Here you see the change in appearance on the Home page and the Employees Report page.

Overriding Application Defaults at the Page Level



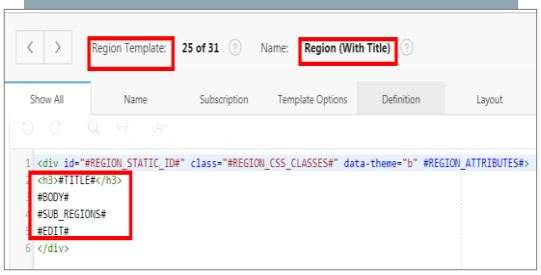
Now, there may be situations where you have defined an application-level default template; however, for a particular page, you want to use a different template. For example, you can specify a page template default to be *Left Side Column*, but for a specific page, you want to use *Left and Right Side Columns*. To specify the page-level template, perform the following steps:

1. Navigate to the page definition in Page Designer view and click the page node under the page rendering tree view. The page properties will open in its Property Editor.
2. Locate **Appearance** in the Property Editor and select *Left and Right Side Columns* for the **Page Template** from the drop-down list.

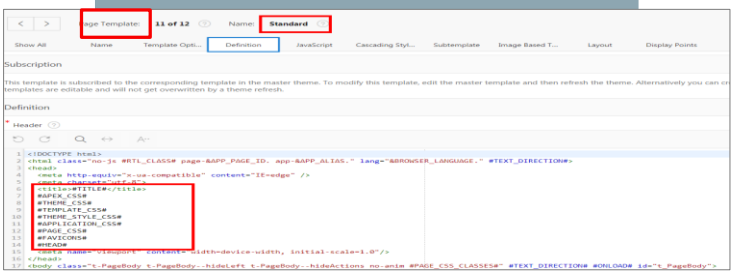
Using Substitution Strings in Templates

```
<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--showLeft t-PageBody--hideActions no-anim #PAGE_CSS_CLASSES#" #ONLOAD# id="t_PageBody">
#FORM_OPEN#
<header class="t-Header" id="t_Header">
```

Region Substitution String: Example



Page Substitution String: Example



ORACLE

Now, Steve wants to add the company logo into his application. He was contemplating the best way to reference an uploaded image in his application or for any particular application page. Application Express provides substitution strings that you can use to dynamically reference image, region, title, body, and so on in your application.

In the above example, #TITLE# is a substitution string that is replaced with the title text at run time.

Following are the properties of 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

Another 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 Application Express engine replaces these strings with values, other objects, or null values.

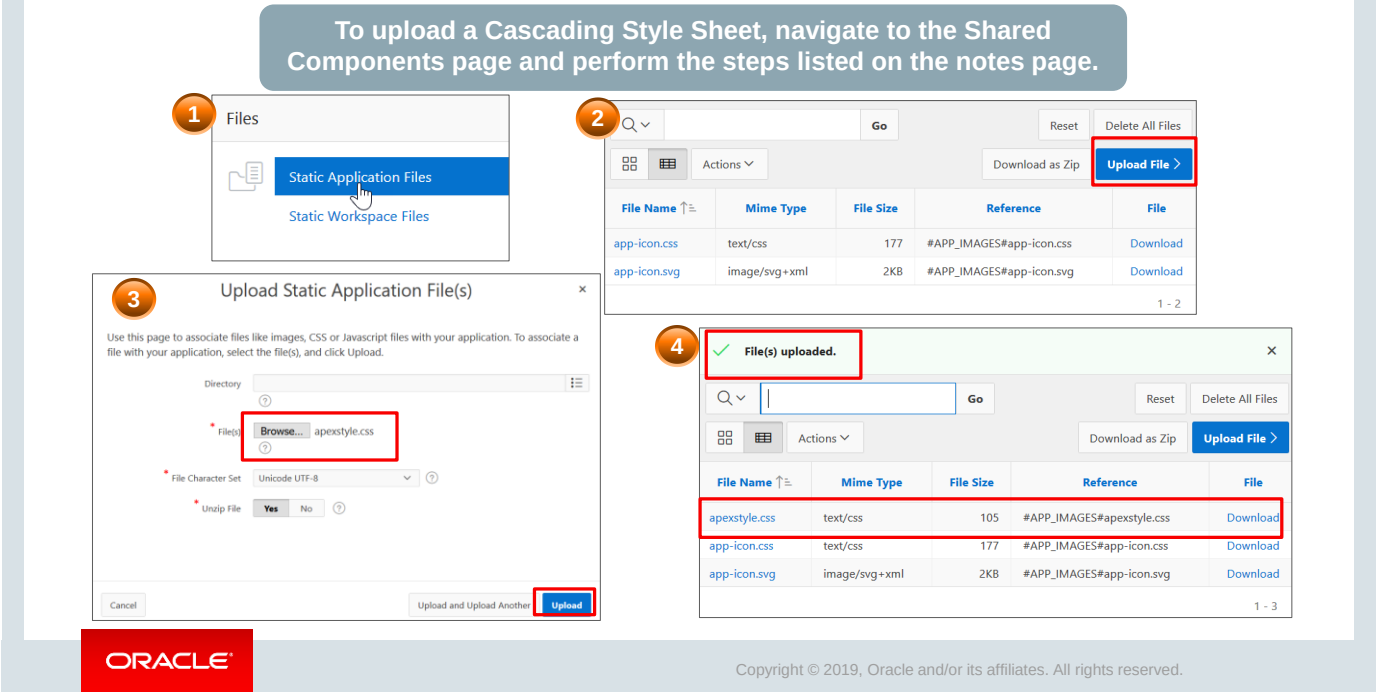
For more information, see Oracle Application Express documentation on Using Substitution Strings (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/understanding-substitution-strings.html#GUID-A896A94B-DF69-4D53-B422-3256C09AE464).

Note that, in later slides (while working on files), you will be using substitution strings to reference an image on your application pages.

Lesson Agenda

- Using Themes
- Using Templates
- Working with Files
  - Uploading a Cascading Style Sheet
  - Referencing Cascading Style Sheets
  - Uploading an Image
  - Using the Uploaded Image

Uploading a Cascading Style Sheet



In the previous slides, you learned that you can create a custom theme by modifying existing templates. After you have created one or more default templates, you can modify those templates to fit your specific needs. This is where a cascading style sheet (CSS) comes in. CSS provides a way to control the style of a web page without changing its structure. A CSS separates visual attributes, such as color, margins, and fonts, from the structure of the HTML document.

Oracle Application Express includes themes that contain templates that reference their own CSS. The style rules defined in each CSS for a particular theme also determine the way reports and regions are displayed.

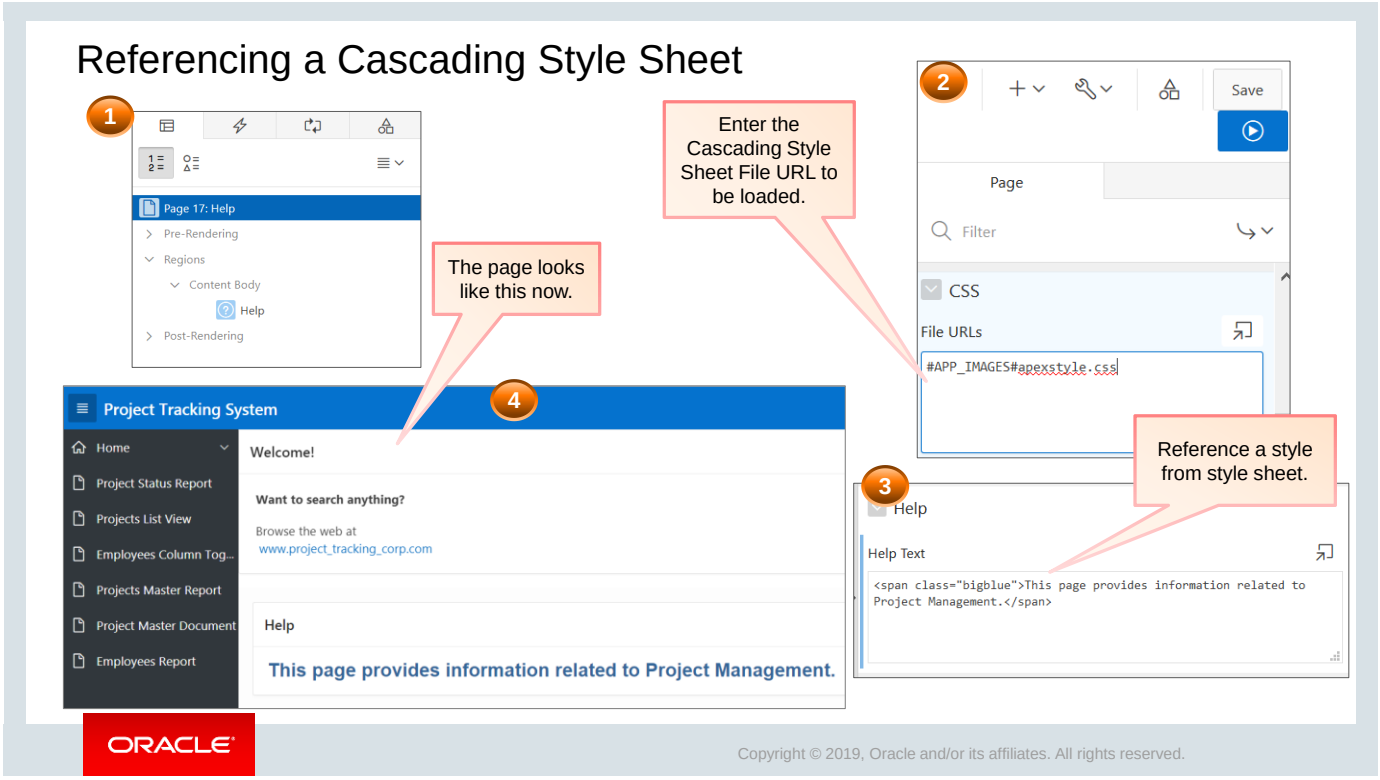
Steve has a cascading style sheet for his theme style, which he wants to reference while enhancing the look of his application's *Help* region. But before he references the CSS in his application, he has to upload the file. Let's see how:

To upload a CSS, navigate to the Shared Components page of the application and perform the following steps:

1. Under Files, click **Static Application Files**.
2. Click **Upload File**.
3. Choose the file, which has to be uploaded. Browse for the `.css` file and click **Upload**.

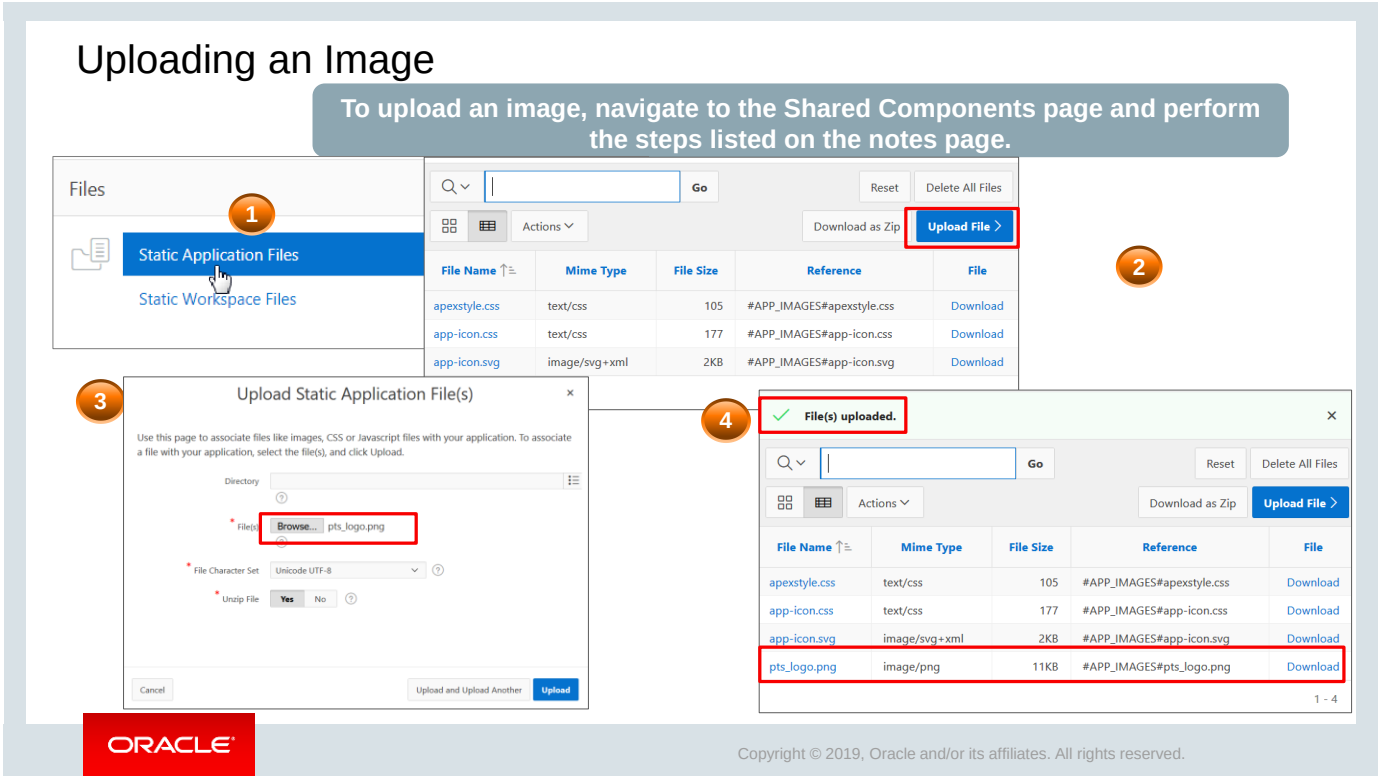
   The file is uploaded successfully.

In the next slide, you will learn how he references this file on his application page.



You can reference an uploaded CSS by modifying the page attributes. Perform the following steps:

1. In the page definition of the page, select the page from the Rendering pane. In this example, Steve selects `Page 17: Help` (screenshot 1).
2. In the Property Editor, scroll down to the CSS subsection. In the **File URLs** field, enter the reference `#APP_IMAGES#apexstyle.css` (screenshot 2). This is the reference of the uploaded file.
3. Enclose the text where you would like to apply the CSS with the `<span>` tag. In this example, the CSS “bigblue” is applied to the Help text (screenshot 3).
4. Save and run the page. You will notice the change.

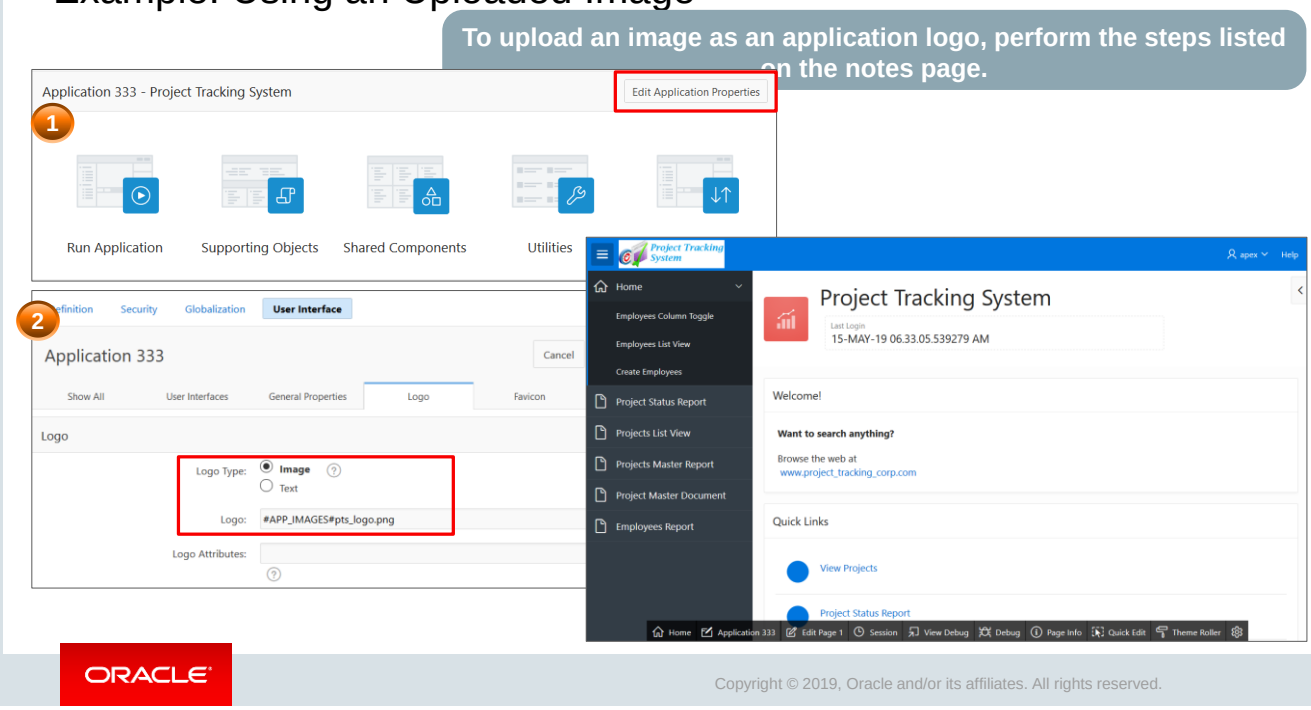


Steve now wants to upload an image (*PTS* logo) that he wants to reference in his application. To upload an image, navigate to the Shared Components page of the application and perform the following steps:

1. Under Files, click Static **Application Files**.
2. Click **Upload File**.
3. Browse for the image file and click **Upload**. The file is uploaded successfully.

You can copy the reference of the file `#APP_IMAGES#pts_logo.png`. In the next slide, Steve is going to reference to this image (this is where the substitution string that you learned in an earlier slide come in), because he wants the company logo to appear on all the pages of his application. Let's see how in the next slide.

Example: Using an Uploaded Image



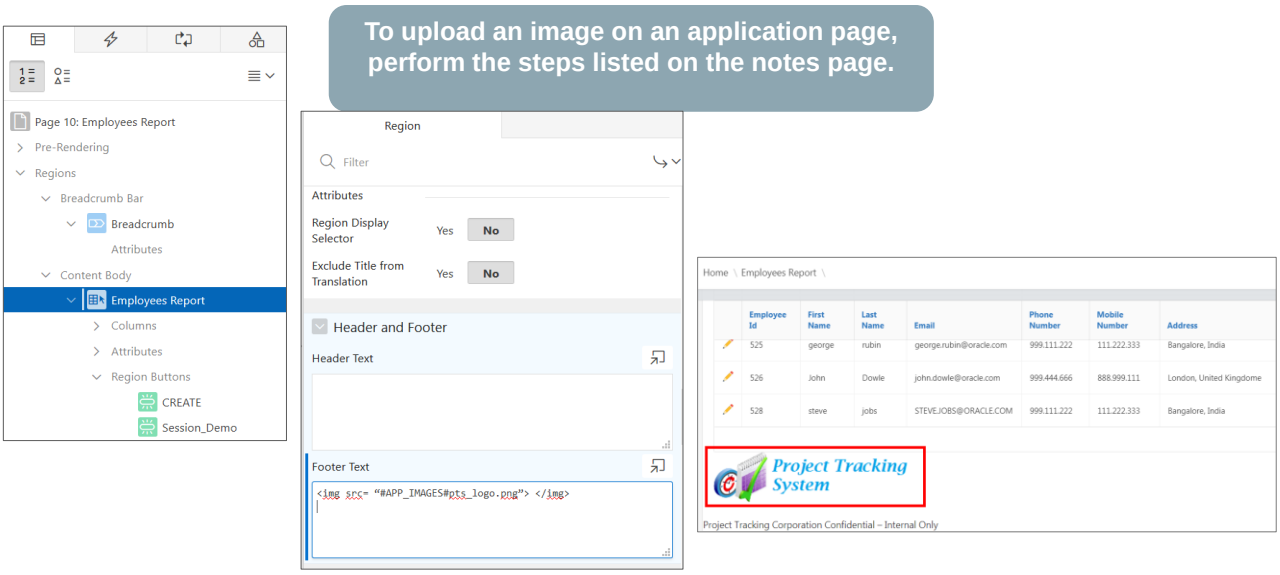
You can use the images uploaded to a workspace on application pages or as a logo for the application. To specify the uploaded image as a logo for the application:

1. Click the **Edit Application Properties** button on the application home page.
2. Select **User Interface** and click the **Logo** tab. Specify the image name in the **Logo** field. Remember you copied the reference of the file that you uploaded in the previous slide? Steve enters that here: `#APP_IMAGES#pts_logo.png`

Note: You can get the image file details from the Reference column on the Files report under **Shared Components > Files > Static Application Files** or **Workspace Application Files** depending on whether the image is uploaded as an application file or as a workspace file.

In the next slide, you learn how to use an image on an application page. (Here Steve chooses the *Employees Report* page.)

Example: Using an Uploaded Image



To reference an image on application pages, you can use one of the following substitution strings:

- `#APP_IMAGES#` is used when the uploaded image is specific to the given application.
- `#WORKSPACE_IMAGES#` is used when the uploaded image is shared among various applications in the given workspace.
- `#IMAGE_PREFIX#` is used when you want to point to the images directory distributed with Oracle Application Express.

In the example in the slide, you see that Steve has added the company logo into the *Employees Report* Footer. You know that Steve has already uploaded the PTS logo; he is now going to reference to this image. Let's see how.

1. Open *Employees Report* in Page Designer view. Select **Report 1**.
2. In the Properties Editor, go to the **Header and Footer** attribute. Enter the following for the **Footer Text**: `<img src= "#APP_IMAGES#pts_logo.png"> </img>`
3. Save and run the page. You can see the PTS logo appearing as a Footer on *the Employees Report* page.

Quiz

Which substitution string would you use to upload a CSS that is associated with a specific workspace?

a. `#IMAGE_PREFIX#`

b. `#APP_IMAGES#`

c. `#WORKSPACE_IMAGES#`

Answer: c

Practice 15 Overview: Working with Themes, Templates, and Files

This practice covers the following topics:

- Creating a theme from the scratch
- Copying, editing, and switching theme
- Editing templates
- Uploading a CSS and applying to a template

Summary

In this lesson, you should have learned how to:

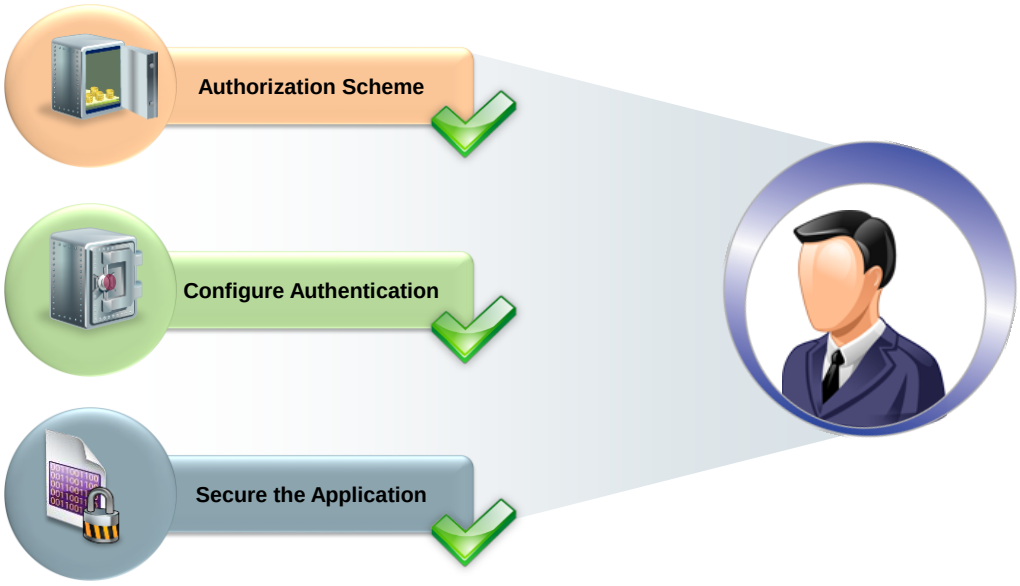
- Define themes and their uses
- Create a new theme
- Copy a theme
- Edit a theme
- Switch to a different theme
- Explain Universal Theme and Theme Roller
- Use Theme Roller to change the theme style
- Define templates and their uses
- View existing templates
- Create, copy, edit, and replace a template
- Upload and use a cascading style sheet and an image

This lesson provided an overview of the themes and the page, region, report, and other templates in Oracle Application Express.

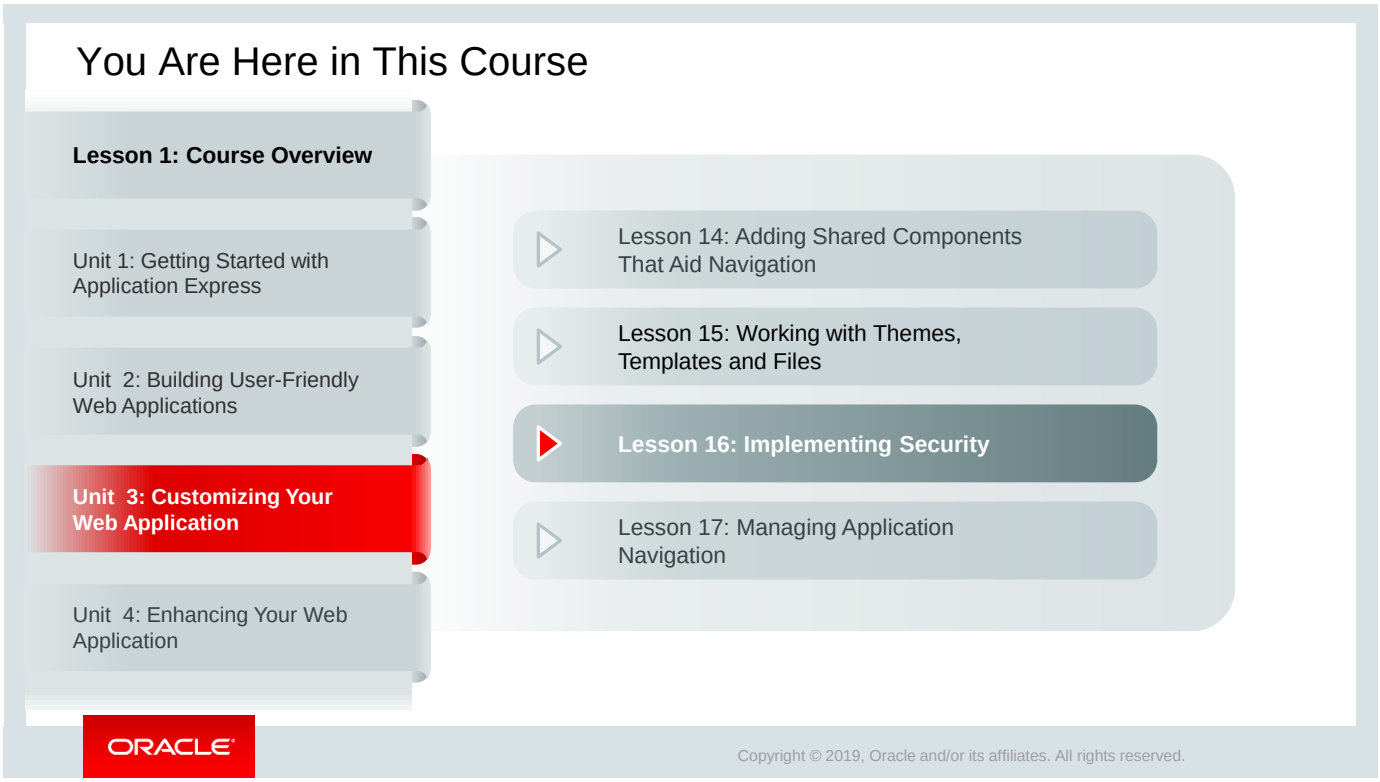
16

Implementing Security

Steve Implements Security in the Application



Steve is now concerned about the security of the PTS application. He is exploring various features available in Oracle Application Express that can help him in securing the application. He knows that once the PTS application is available for use, there will be multiple users like project managers, developers, and team members who would be using the varied features on a regular basis. Therefore, he wants to be careful about giving access to different pages of the application for different users. And with that in mind, he plans to add security features, such as authentication and authorization for various levels of users, to make the application secure, before it is made available in the production server.



This slide is a graphical depiction of the course, particularly highlighting Unit 3 - Lesson 16, which is dealt with in these slides.

Objectives

After completing this lesson, you should be able to:

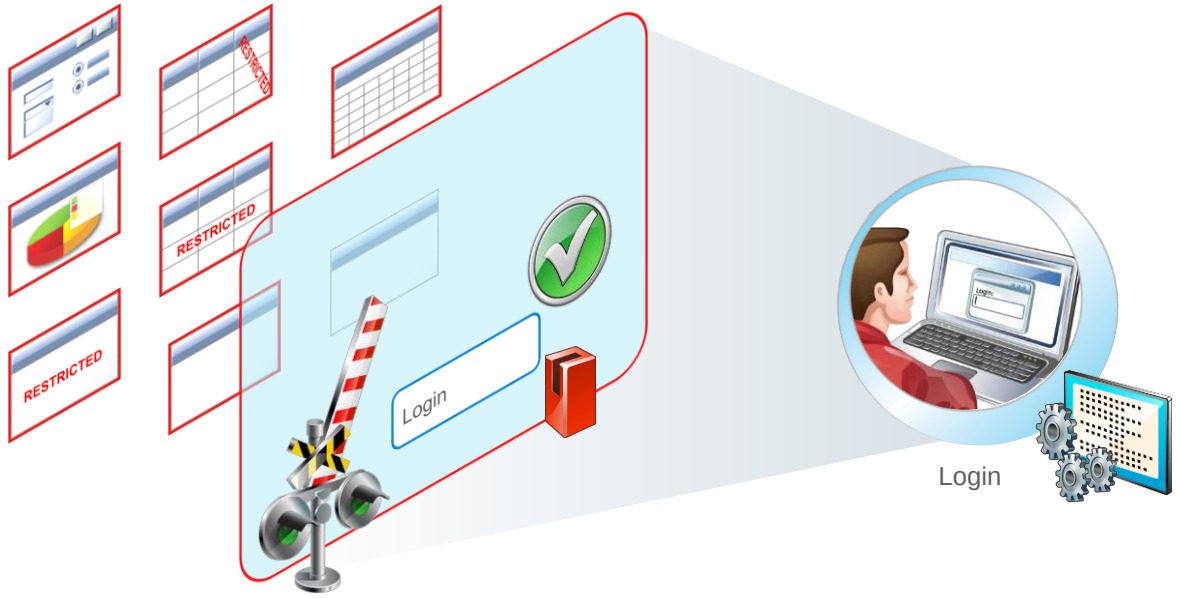
- List the different ways to secure your application
- Differentiate between authentication and authorization
- Create an authentication scheme for your application
- Create an authorization scheme by using Access Control
- Enable and configure Session State Protection

This lesson shows you how to implement security for an application by using the security features of Oracle Application Express. You learn the difference between authentication and authorization. You also learn how to enable Session State Protection in your application to prevent hackers tampering with the URLs within the application.

Lesson Agenda

- Securing an Application
  - Overview
  - Accessing the Security Tasks
- Using Authentication Schemes
- Using Authorization Schemes
- Using Session State Protection

Securing an Application: Overview



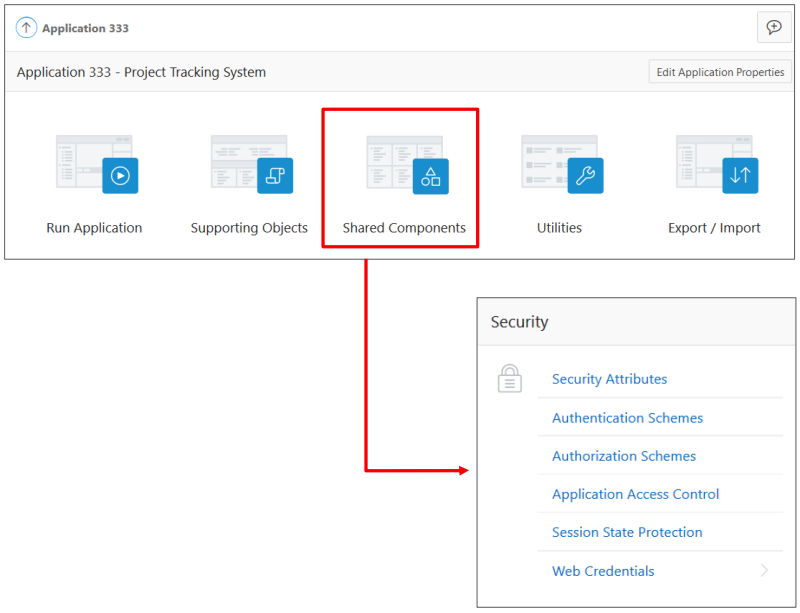
As you know, Steve wants to ensure that only authorized users can access the PTS application and, most importantly, have rights to access certain pages in his application. For example, he wants to give a Reader access (only read privilege) to the *Project Status Report* page but wants to give Contributor access (with edit and read privileges) on the *Project Members* form page.

Oracle Application Express enables you to secure your application using the following methods:

- **Authentication:** Use this method to confirm user credentials before allowing access to the application. This is done through a login page. The user can view any component of the application only if the login succeeds.
- **Authorization:** Use this method to restrict access to specific pages, components (for example, forms, reports, or items), or to a particular column in a report. Only privileged users can access these components.
- **Session State Protection:** Use this method to prevent users from tampering with the URLs.

In later slides, you will learn how to use each of these security methods. But let's first start with how to access these security tasks from your application's home page.

Accessing Security Tasks



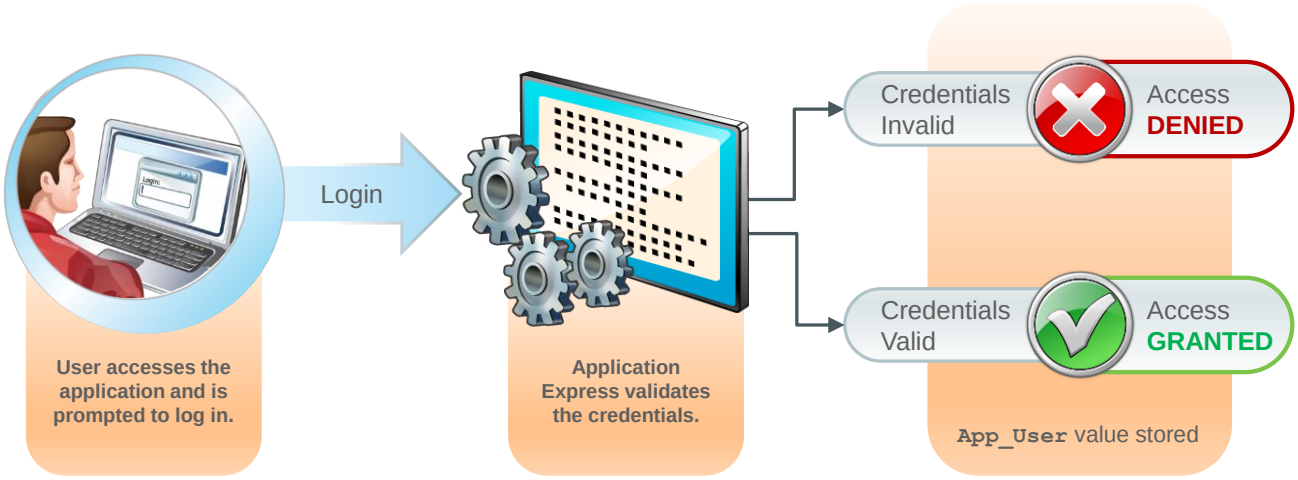
To create security mechanisms for an application, navigate to **the Shared Components** page and click the appropriate link in the Security list.

In the next few slides, you will learn how to use these methods to secure your application.

Lesson Agenda

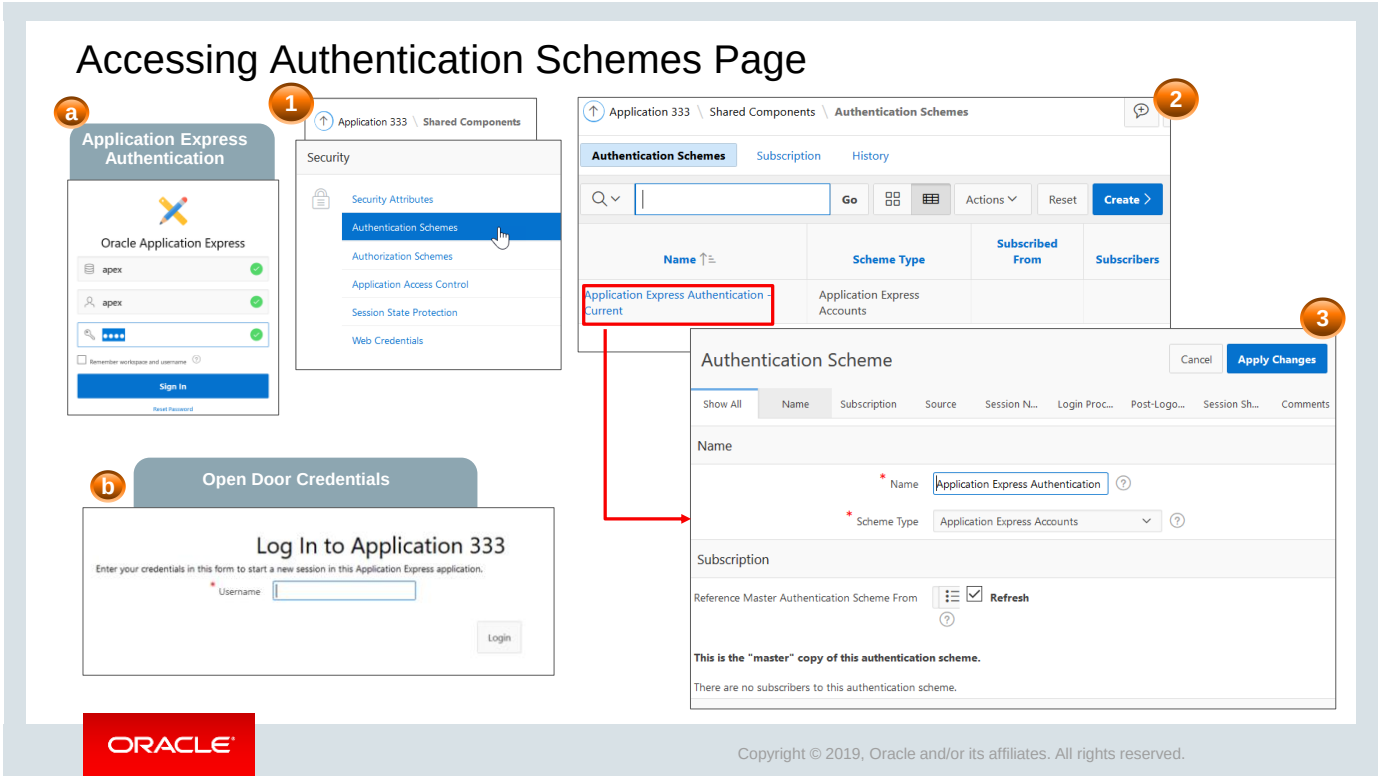
- Securing an Application
- Using Authentication Schemes
  - Using Authentication
  - Accessing Authentication Schemes Page
  - Using Preconfigured Schemes
    - Examples: Preconfigured Authentication Schemes
  - Creating Authentication Based on Preconfigured Schemes
  - Copying an Authentication Scheme
- Using Authorization Schemes
- Using Session State Protection

Using Authentication



See the *Access Denied* and *Access Granted* options in the slide. When your application uses an authentication scheme, Oracle Application Express prompts each user for a username and password when the user tries to log in. The credentials are evaluated, and accordingly, the user is allowed or denied access to the application. After a user is identified, the Oracle Application Express engine keeps track of the user by setting the value of `APP_USER`. `APP_USER` is a built-in variable representing the current user running the application. The Oracle Application Express engine uses `APP_USER` to track each user's session state.

In the next slide, let's see how to access the authentication page for your application.



The slide shows two different login screens on the left. The one on top displays a login page prompting you for a username and password. You must enter the user credentials created by using Oracle Application Express for this application. And the one below shows another authentication scheme where the user just needs to enter the username to log in to the application.

The Application Express Authentication scheme enables access to users created in Application Express. The Authentication Schemes page displays the authentication schemes available for an application.

To access the Authentication Schemes page, click the **Authentication Schemes** link under Security on the Shared Components page of the application (screenshot 1). The scheme that is current for the application is appended with the word “Current” (screenshot 2). You can create more than one authentication scheme for an application, but only one scheme can be current. Click the link on the row to view details about the current authentication scheme for an application (screenshot 3).

Note that if you choose not to authenticate your application, Oracle Application Express does not check user credentials. In that case, all the pages of your application are accessible to all users.

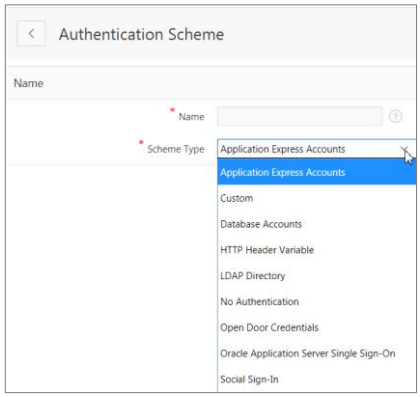
In Oracle Application Express, you can create authentication by:

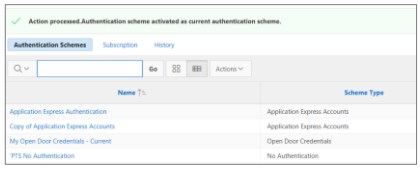
- Using one of the preconfigured schemes
- Copying an authentication scheme from the same application or from a different application and then modifying the settings as needed

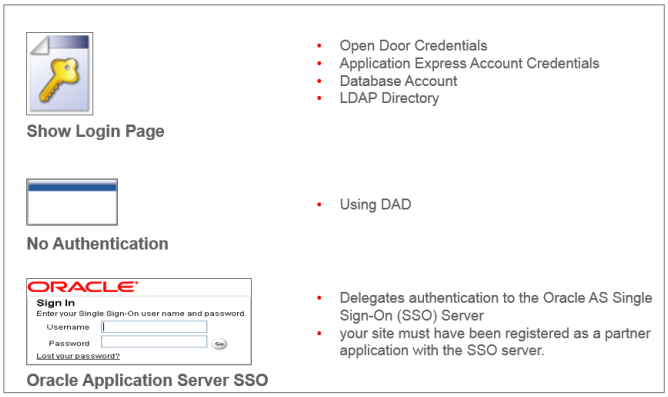
In this lesson, you learn to create authentication by using these two methods.

Let’s now start learning about preconfigured schemes provided by Oracle Application Express.

Using Preconfigured Authentication Schemes







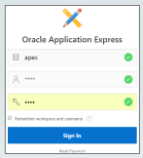
Oracle Application Express provides some common, pretested authentication schemes (see the slide) that you can choose while creating an authentication scheme. When you create an authentication scheme from the Oracle Application Express gallery, you can select a preconfigured authentication scheme, which follows a standard behavior for authentication and session management. Note that if you create a new authentication scheme, it automatically becomes the current authentication scheme for the selected application.

In the next slide, you see some examples of the preconfigured authentication schemes provided by Oracle Application Express.

Examples: Preconfigured Authentication Schemes

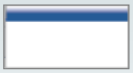
Application Express Accounts

- To log in to an application by using this scheme, you must provide the user credentials created by using Oracle Application Express for this application.
- These user accounts are created and managed by an Oracle Application Express Workspace administrator.
- When you create this scheme, you have the option to specify whether to use a built-in login page or a custom login page.



No Authentication

- Provides no authentication for the application
- No login page shown and all the pages of an application accessible to all users
- Uses Database Access Descriptor (DAD) configuration, which defines how Application Express will automatically log in to the database



Oracle Application Server SSO

- Delegates authentication to the Oracle AS Single Sign-On (SSO) Server
- To use this authentication scheme, your site must have been registered as a partner application with the SSO server.

Open Door Credentials

- Enables anyone to access your application using a built-in login page that captures a username

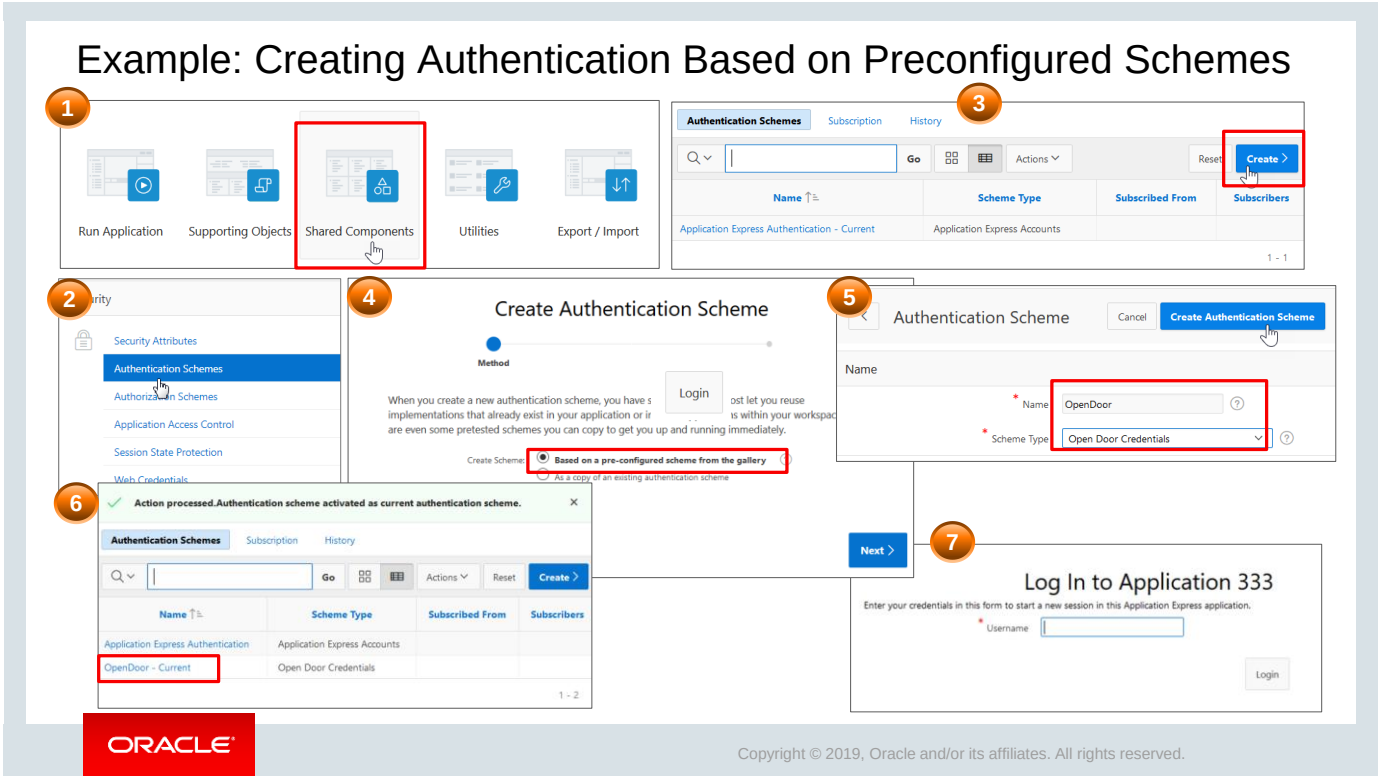
Social Sign-in

- Supports authentication with Google, Facebook, and other social network

For information about all preconfigured authentication schemes that are provided by Oracle Application Express, see "*Understanding Preconfigured Authentication Schemes*" (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/establishing-user-identity-through-authentication.html#GUID-CD382D4A-AC00-4185-B37F-9A5BC9417A7B).

Note that you will learn more about Social Sign-In Authentication scheme in *Advanced Workshop of Oracle Application Express (WS II)*.

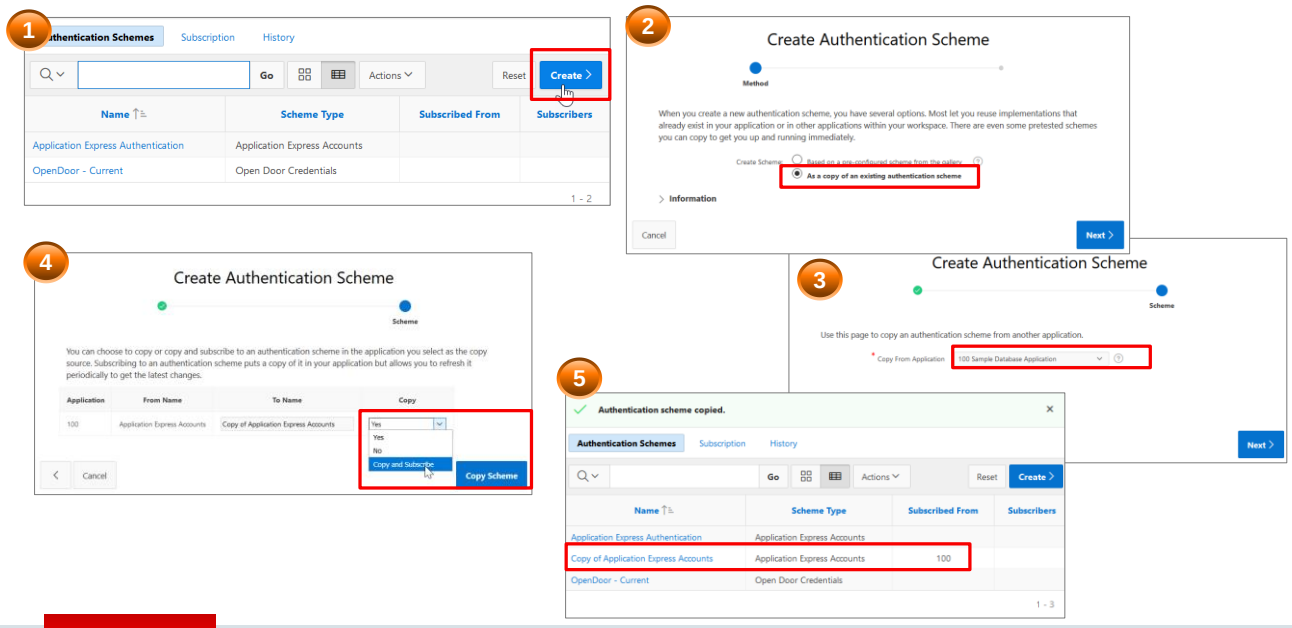
In the next slide, Steve creates a new authentication scheme for his application, using one of the above preconfigured schemes provided by Oracle Application Express. Let's see how.



Now, Steve wants to create an authentication scheme for his application, and he chooses Open Door Credentials as the preconfigured scheme. The Open Door Credentials enable anyone to access your application using a built-in login page that captures a username. This authentication scheme is mostly useful during application development. Steve feels that an authorized username would be enough, and the authenticated user does not need to enter the password every time he or she logs in to the application. Let's see how he does it.

1. Select Shared Components from your application's home page (screenshot 1).
2. Under Security, select Authentication Schemes (screenshot 2).
3. On the Authentication Schemes page, click the **Create** button (screenshot 3).
4. On the Create Authentication Scheme page, select **Based on a pre-configured scheme from the gallery** and click **Next** (screenshot 4).
5. Select the **Open Door Credentials** scheme. Enter a name for the new authentication scheme and click **Create Authentication Scheme** (screenshot 5).
6. The authentication scheme is created successfully and is appended as **Current**. The new scheme becomes the current authentication scheme for your application.
7. Run the application, and it prompts you with the Open Door Credentials to log in. You will see that you no longer get the Oracle Application Express login page but a login page asking just for your username.

Copying an Authentication Scheme



Steve is wondering what if instead of creating a new authentication scheme, he copies an authentication scheme from his application or any other application in his workspace and uses it to authenticate his application? He can then edit the copied scheme and change the name and other settings to meet his application requirements. Let's see how he does it.

1. On the Authentication Schemes page, click the **Create** button (screenshot 1).
2. Select **As a copy of an existing authentication scheme** and click **Next** (screenshot 2).
3. Select the application from which you want to copy the scheme and click **Next** (screenshot 3).
4. The schemes existing in the selected application are listed. Select **Yes** for the scheme that you want to copy. The **Copy and Subscribe** option copies the authentication scheme to your application, and you can refresh it periodically to retrieve the latest changes. Click **Copy Scheme** to copy the scheme (screenshot 4).

   The copied authentication scheme is now displayed in the list of authentication schemes (screenshot 5).

Quiz

Which authentication scheme uses the built-in users created by a workspace administrator within the workspace where the application is installed?

a. Open Door Credentials
b. Database Account Credentials
c. Oracle Application Express Credentials
d. LDAP Credentials

Answer: c

Practice 16-1 Overview: Creating an Authentication Scheme

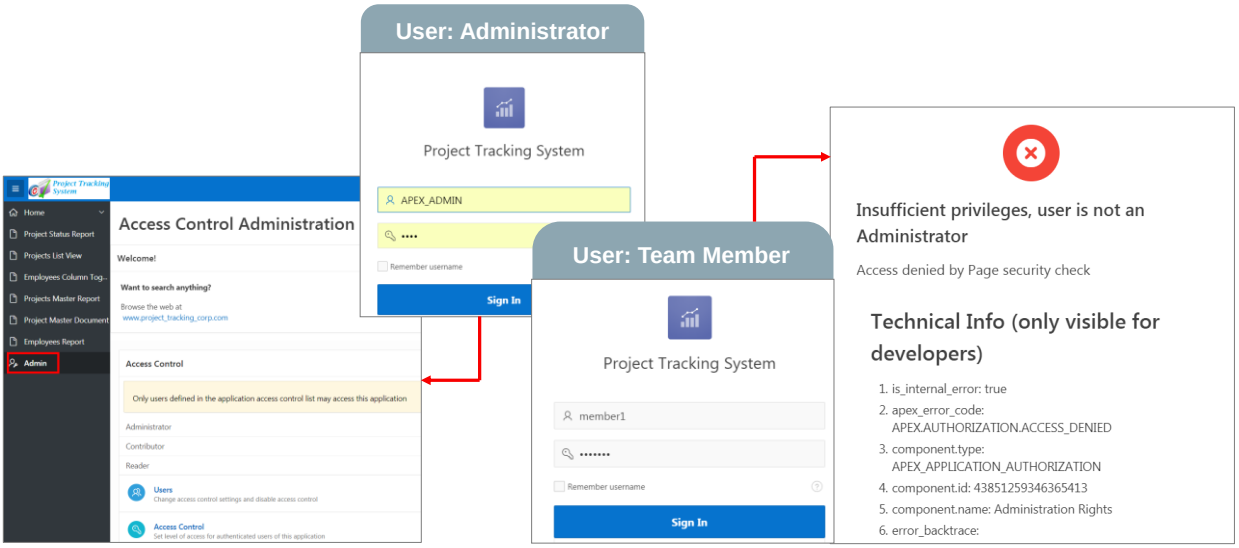
This practice covers the following topics:

- Creating an authentication scheme
- Switching the current authentication scheme to Application Express

Lesson Agenda

- Securing an Application
- Using Authentication Schemes
- Using Authorization Schemes
  - Where Can You Implement Authorization?
  - Methods to Implement Authorization
  - Creating an Authorization Scheme from the Scratch
  - Creating Users and Roles
  - Creating an Access Control Page
  - Setting Access Control Mode
  - Creating Access Control List
  - Applying an Authorization Scheme
- Using Session State Protection

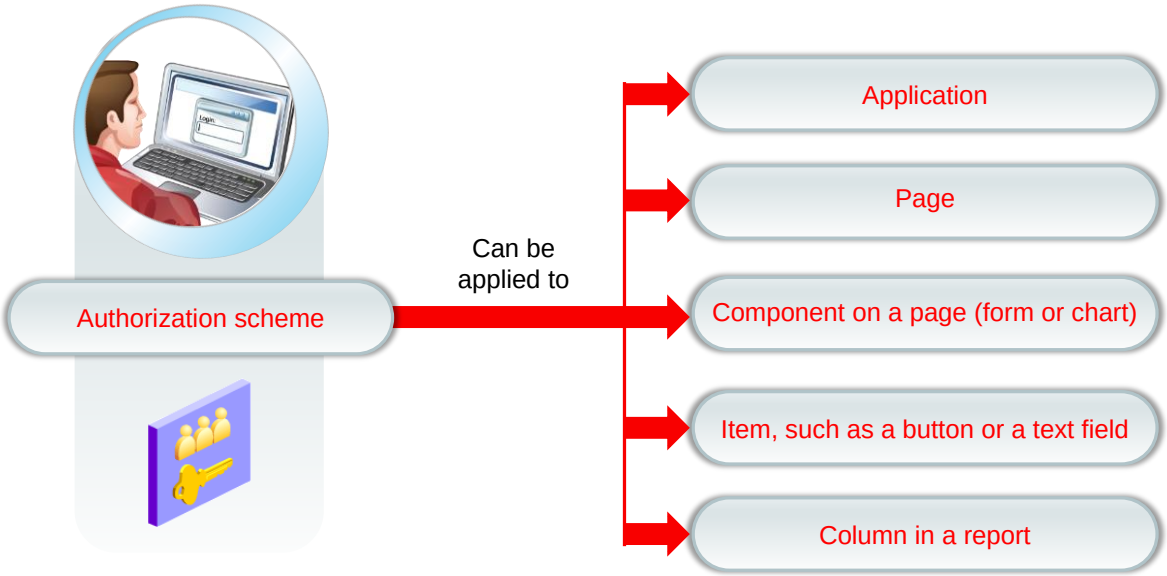
Using Authorization



In the slide, you see two users: the Administrator user and Team Member. The Administrator user has Administrator rights with access to the Admin page of the application. That is, he or she has the authorization to be an administrator, contributor, as well as a reader. However, on the other hand, the Team Member is authorized to have only reader rights. That is, he or she can just view the application but cannot contribute or edit the application.

Authorization, therefore, is a broad term for controlling access to resources based on user privileges. Let's learn more about this in the next few slides.

Where Can You Implement Authorization?



Authorization controls access to resources within the application. Authorizations are implemented by using authorization schemes. You can specify an authorization scheme for an entire application, a page, or specific components such as a region, an item, a button, or a column of a report. If the component-level authorization succeeds, the user can view the component. If the application-level or page-level authorization fails, Oracle Application Express displays a predefined message “insufficient privileges.”

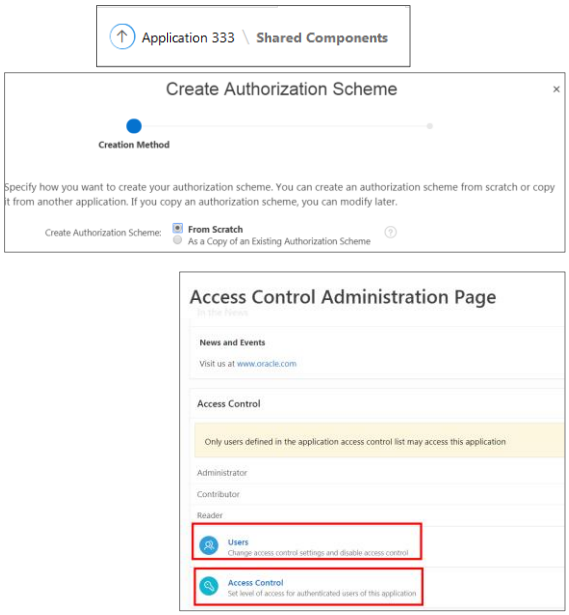
You first define the authorization scheme and then associate it with any component in your application.

You can view and modify the authorization schemes associated with a page from the **Security** node in the **Shared Components** column on the Page Definition page. You will learn how to in the next few slides.

Methods to Implement Authorization

Two ways to create and implement an authorization scheme:

- Shared Components
  - Create an authorization scheme from the beginning.
  - Copy an authorization scheme from an existing scheme.
- Access Control Administration page:
  - Create an Access Control page.
  - Set the Access Control mode.
  - Add users to the Access Control List.
  - Apply the authorization scheme to application components.

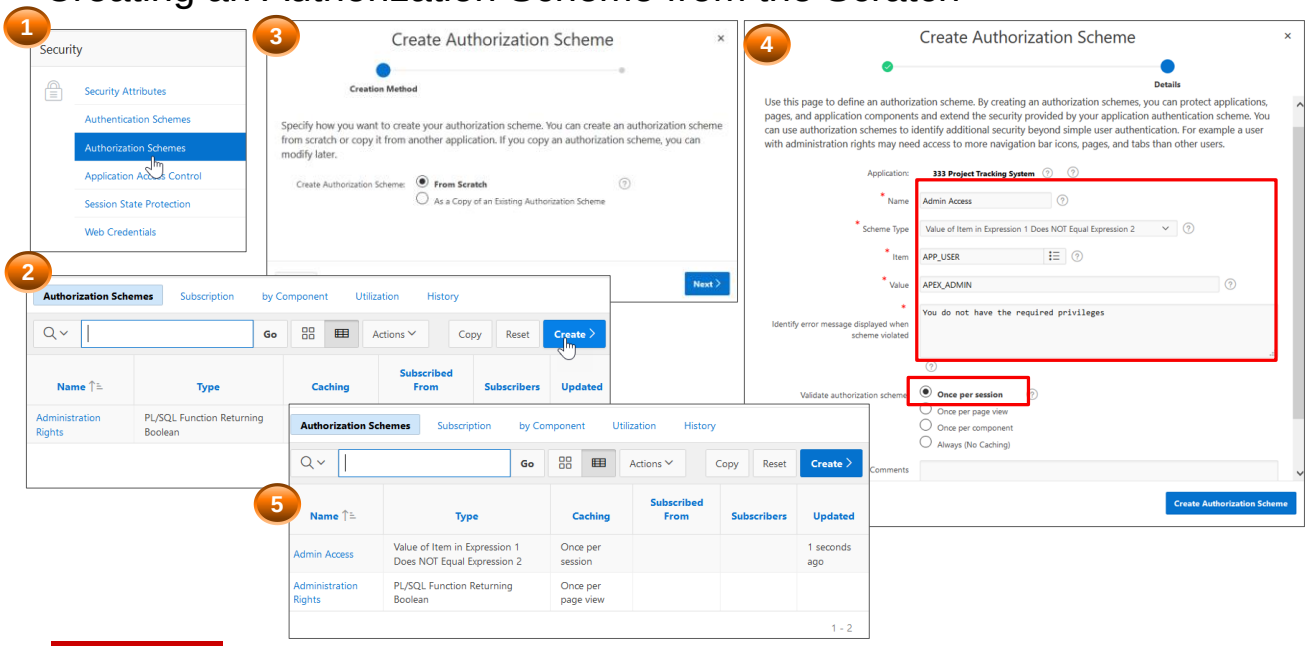


There are two ways to create and apply an authorization scheme to an application and its components:

- You can create an authorization scheme from the beginning or from an existing scheme from the **Shared Components** page of an application.
- You can also create an authorization scheme through an **Access Control** page, which automates the step of creating the authorization schemes. The **Access Control** page enables you to set the Access Control mode to restricted access, if any, that the application should have. The **Access Control** page also enables you to define each user and the access that the user should have. You can also apply the authorization scheme to various application components.

You will learn about these in detail in the next few slides.

Creating an Authorization Scheme from the Scratch

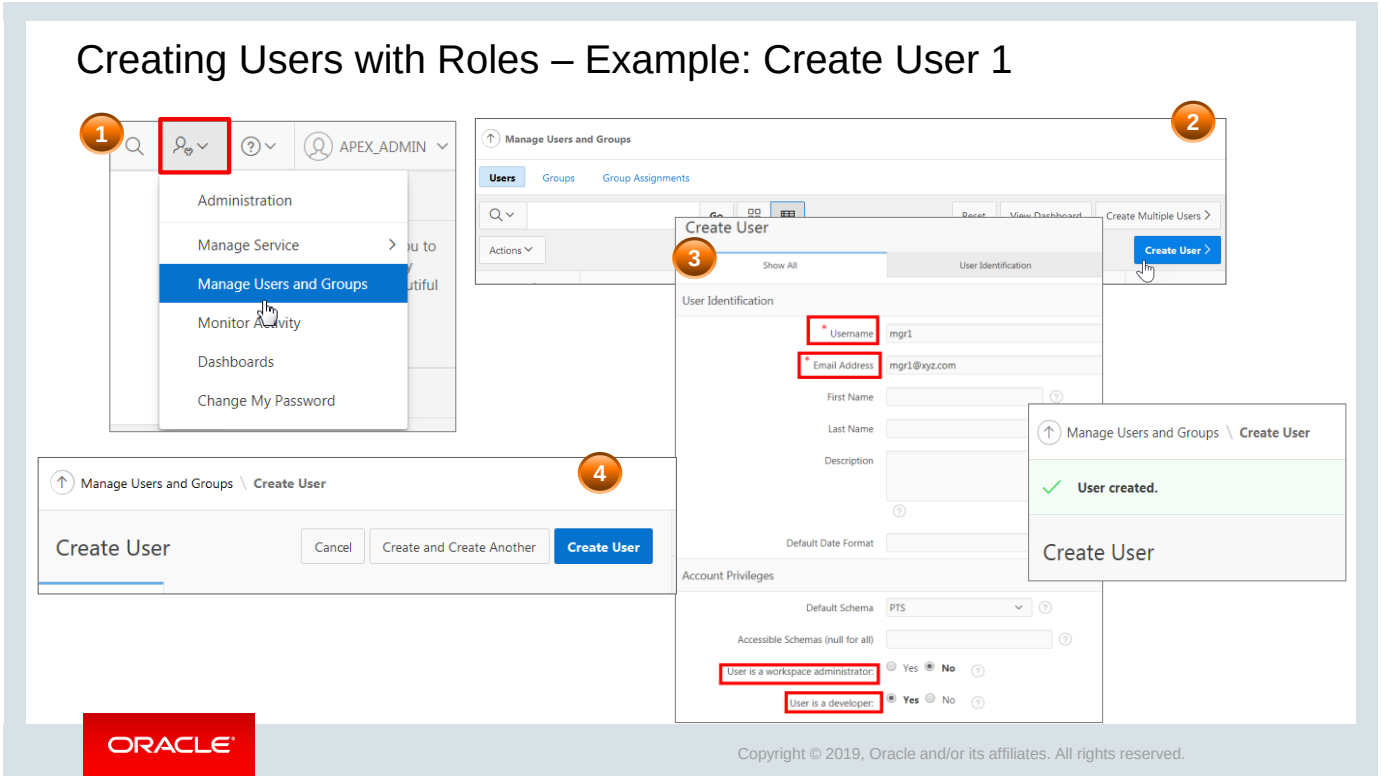


You learned in an earlier slide that you can create an authorization scheme from scratch or copy an existing authorization scheme and then customize it. Steve now wants to create a new authorization scheme, which he wants to apply in his application. Let's see how. To create a new authorization scheme from scratch, navigate to the **Shared Components** page and perform the following steps:

1. In the **Security** section, click **Authorization Schemes** (screenshot 1).
2. On the Authorization Schemes page, click the **Create** button (screenshot 2).
3. Select **From Scratch** and click **Next** (screenshot 3).
4. Specify the following details and click **Create Authorization Scheme** (screenshot 4).
   - Enter a name for the scheme (for example, *Admin Access*)
   - Select a scheme type that defines how the scheme will be applied. In this example, Steve *selects Value of Item in Expression 1 Equals Expression 2*. This means whatever value you enter in the Item (that is, in Expression 1) is compared to the value specified in Expression 2. The authorization succeeds if the item's value equals the authorization value.
   - Enter **Item** as `APP_USER`. Here `APP_USER` works as session variables and holds the value of user ID with which you log in to the application.
   - Enter the **Value** as `APEX_ADMIN`.
   - Enter the error text to be displayed when the authorization scheme fails.
   - Specify whether the authorization scheme must be evaluated once per session or once per page view. Authorization schemes are evaluated on first use in a session. Here Steve selects **Once per session**, which means the evaluation will happen only once and always use the memorized result afterwards.

For more information, see *Creating and Editing an Authorization Scheme* (https://docs.oracle.com/en/database/oracle/application-express/19.1/htmdb/providing-security-through-authorization.html#GUID-439E713B-7238-48D6-BE5A-A6D38137F694).

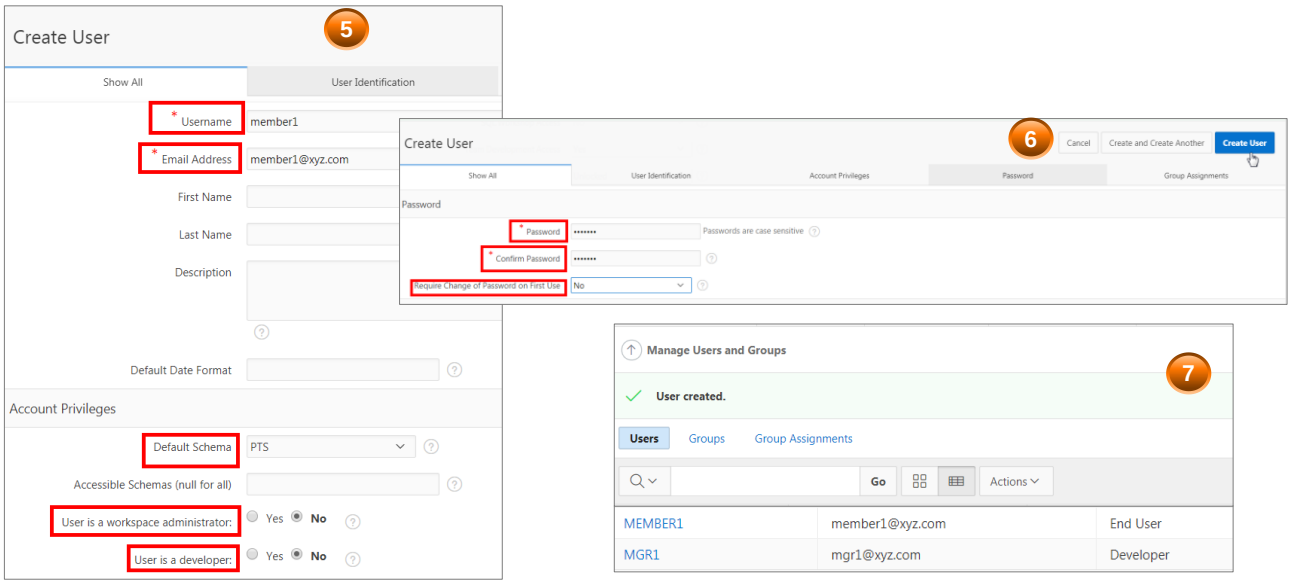
In the next few slides, you will learn how to create an **Access Control** page and add users to the Control List. However, let's start with creating users, and then we will assign their respective privileges.



Steve now starts with his requirement of creating users and assigning them their respective privileges. He first creates a user for the manager, and then he creates another one for the team member. Let's see how:

1. Click the **Administration** icon on the Navigation Bar (top right of Home Page) and click **Manage Users and Groups** (screenshot 1).
2. On the Manage Users and Groups page, click **Create User** (screenshot 2).
3. Enter the following values (screenshot 3):
   - **Username**: Enter `mgr1`.
   - **Email Address**: Enter `mgr1@xyz.com`.
   - **Default Schema**: Select `PTS`.
   - **User is a workspace administrator**: Select *No.*
   - **User is a developer**: Select **Yes.**
   - **Password & confirm password**: Enter ****.
   - **Require change of password on first use**: Select **No**.
4. Click **Create and Create Another** to create another new user (screenshot 4). This is because Steve wants to create another user for the team member. Let's see how in the next slide.

Creating Users with Roles – Example: Create User 2



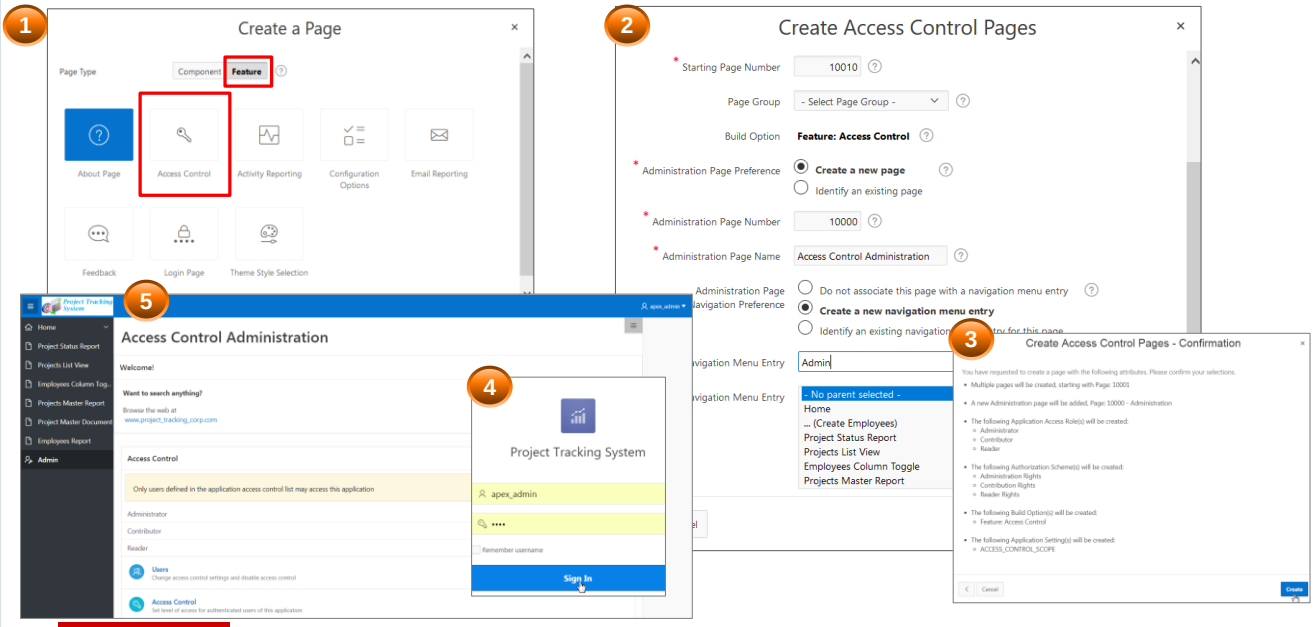
5. Enter the following values (screenshot 5) and click **Create User** (screenshot 6):
    - **Username**: Enter `member1`.
    - **Email Address**: Enter `member1@xyz.com`.
    - **Default Schema**: Select `PTS`.
    - **User is a workspace administrator**: Select *No.*
    - **User is a developer**: Select *No.*
    - **Password & confirm password**: Enter *******.
    - **Require change of password on first use**: Select *No.*

You can see that the two users are now available for the current workspace (screenshot 7):

- mgr1 as Developer
- member1 as End User (that is, the Team Member)

In the next slide, you will be creating an **Access Control** page and later assign users (that you just created) to the Access Control List. Let's see how.

Creating an Access Control Page



Steve first creates an **Access Control** page. He will later use this page to set the access control mode to restricted access, define the users who can access the application, and specify privileges for each user. Let's see how he creates an Access Control page for his application.

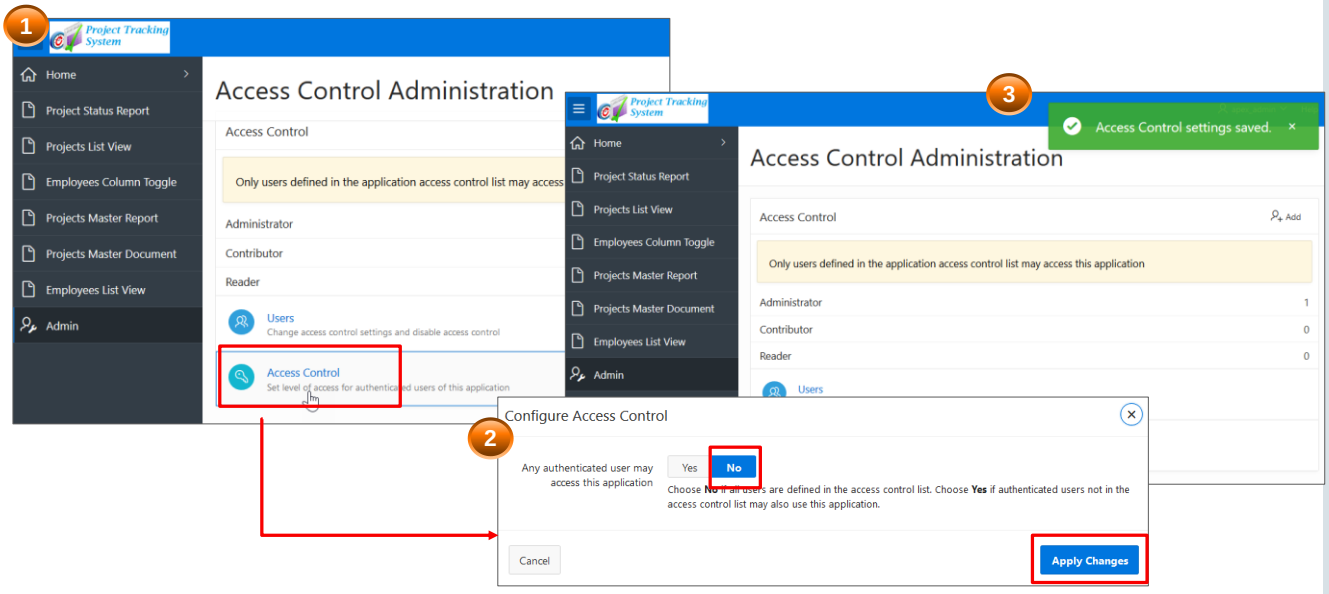
Navigate to the application home page and click **Create Page** and then perform the following steps:

1. Click the **Feature** tab and then select **Access Control** (screenshot 1).
2. On the **Create Access Control Pages** page, enter the following values and click **Next** (screenshot 2):
   - **Starting Page Number**: Enter `10010`.
   - **Administration Page Reference**: Select **Create a new page**.
   - **Administration Page Number**: Enter `10000`.
   - **Administration Page Name**: Enter `Access Control Administration`.
   - **Administration Page Navigation Preference**: Select **Create a new navigation menu entry**.
   - **New Navigation Menu Entry Name**: Enter `Admin`.
   - **Parent Navigation Menu Entry**: Select: **-No parent selected -**.

- The **Create Access Control Pages - Confirmation** page appears. Review the details and click **Create** (screenshot 3). A new Access Control administration page is created.
- Click the Run icon to load the page. Enter login details and click **Login**.
- The Access Control Administration page is now successfully created for the PTS application and is displayed with the message "*Only users defined in the application access control list may access this application*."

Because Steve is also the application administrator, he now wants to set Access Control mode to restricted users such that only the users defined in the Access Control List have access to the application. Let's see how in the next slide.

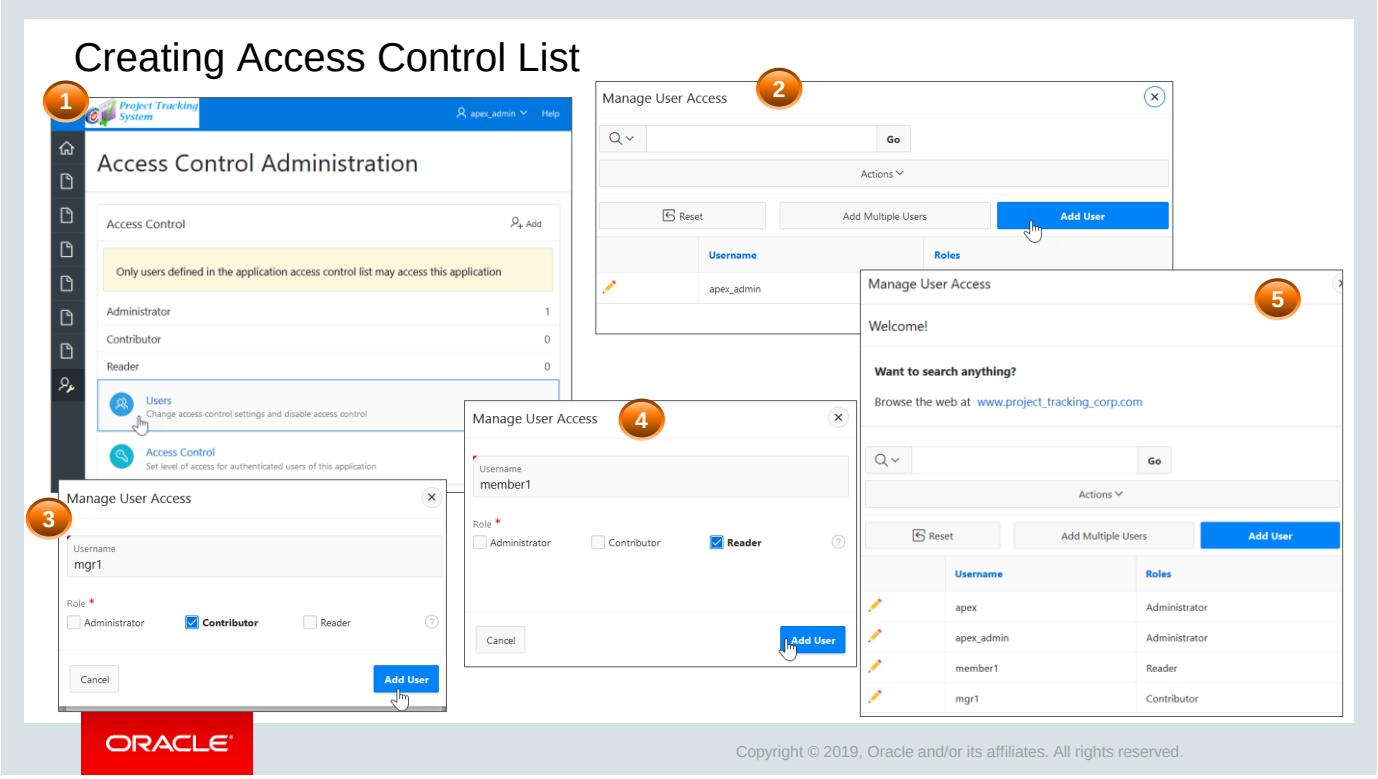
Setting Access Control Mode



To set Access Control mode to restricted users:

1. Click the **Access Control** link on the Admin page.
2. On the **Configure Access Control** modal page, for **Any authenticated user may access this application**, select **No**. This is because Steve wants to grant access only to the users defined in the Access Control list and not to all authenticated users. Click **Apply Changes**.
3. A message will be displayed as *Access Control settings saved*.

The Access Control mode is now set to restricted users (defined in the Access Control List). Steve will now start adding users (that he created in the previous slides) to the Access Control List. Let's see how in the next slide.



As an application administrator, Steve now adds the users to the Access Control List. He can define what type of access he wants the users to have. The options are:

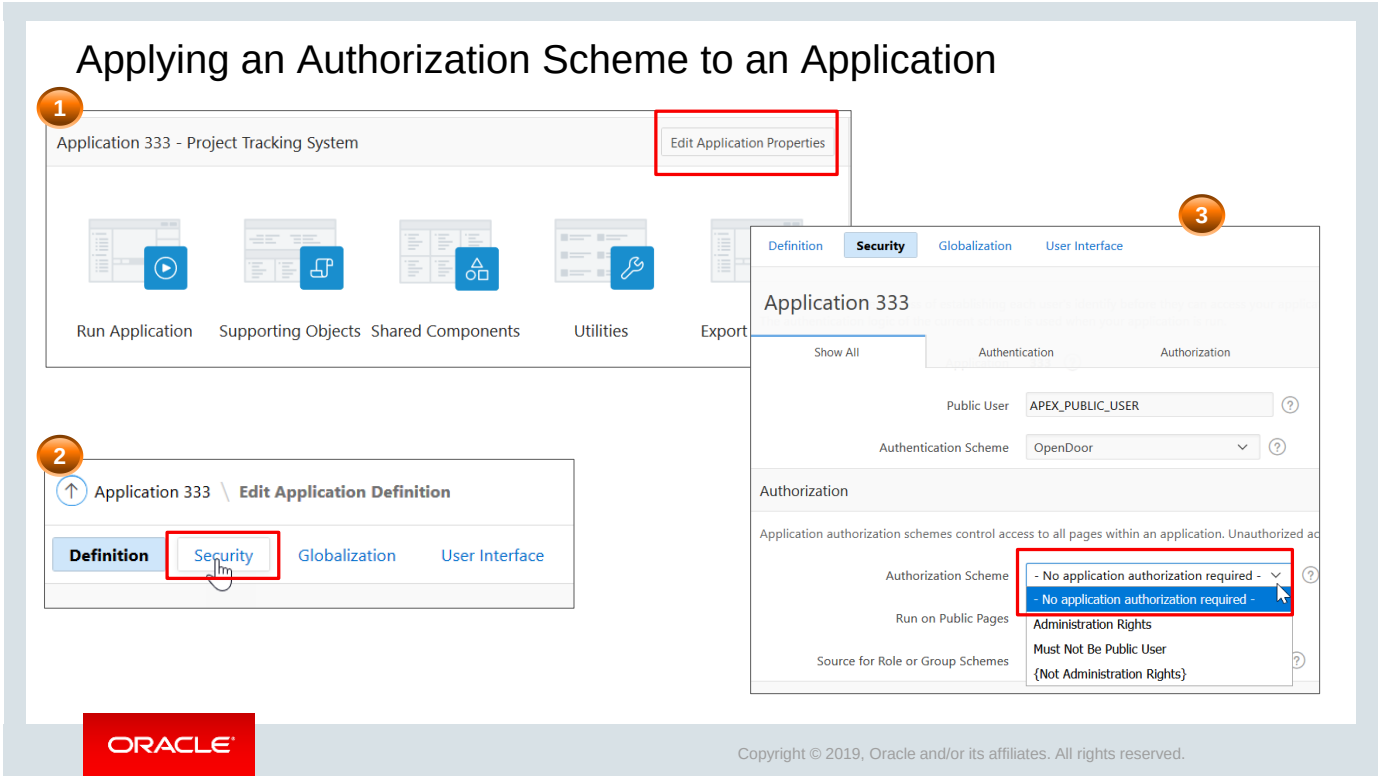
- **Reader:** User can view the application, but cannot edit the content.
- **Contributor:** User can view and edit content.
- **Administrator:** User can view and edit content, and also edit and manage the Access Control List.

To add users and assign privileges to those users in Access Control List, perform the following steps:

1. Click **Users** on the **Access Control Administration Page**.
2. On the **Manage User Access page**, click **Add User** for each user you want to create. Enter the following values and click **Add User**:
    - **Username**: Enter `mgr1`.
    - **Roles**: Select **Contributor**.
    - **Username**: Enter `member1`.
    - **Roles**: Select **Reader**.
- You can also add another user for *apex* with the *Administrator* privilege.
- See three new users with their corresponding roles (privileges) listed as shown below:

Username	Roles
apex	Administrator
mgr1	Contributor
member1	Reader

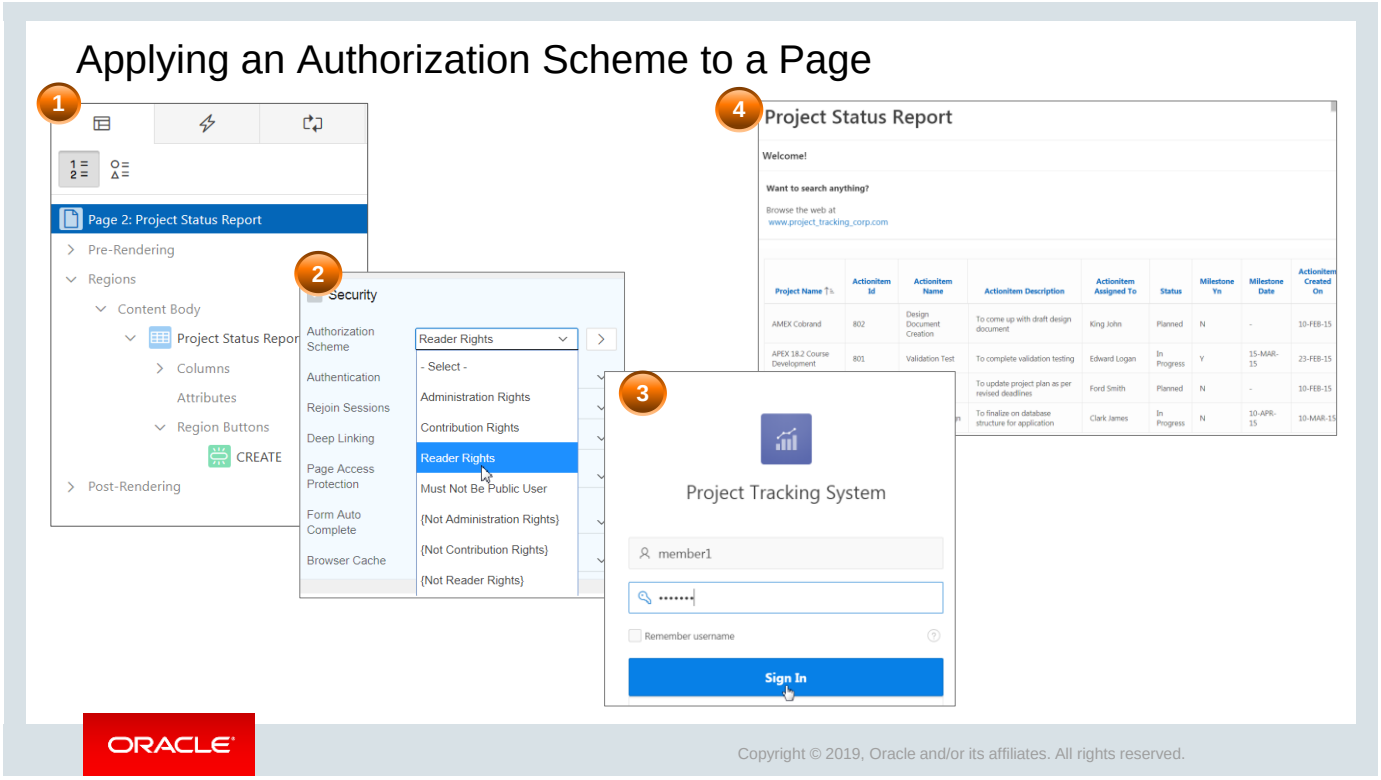
In the next couple of slides, you will define authorization schemes to an application, to a page in the application, and also to a column in a report and see how each user defined in Access Control List has different access privileges.



To apply an authorization scheme to an entire application, navigate to the application home page and perform the following steps:

1. Click the **Edit Application Properties** button.
2. Click the **Security** tab.
3. Click the **Authorization** tab. Select an authorization scheme from the Authorization Scheme drop-down list and click **Apply Changes**. The authorization scheme is applied to your application.

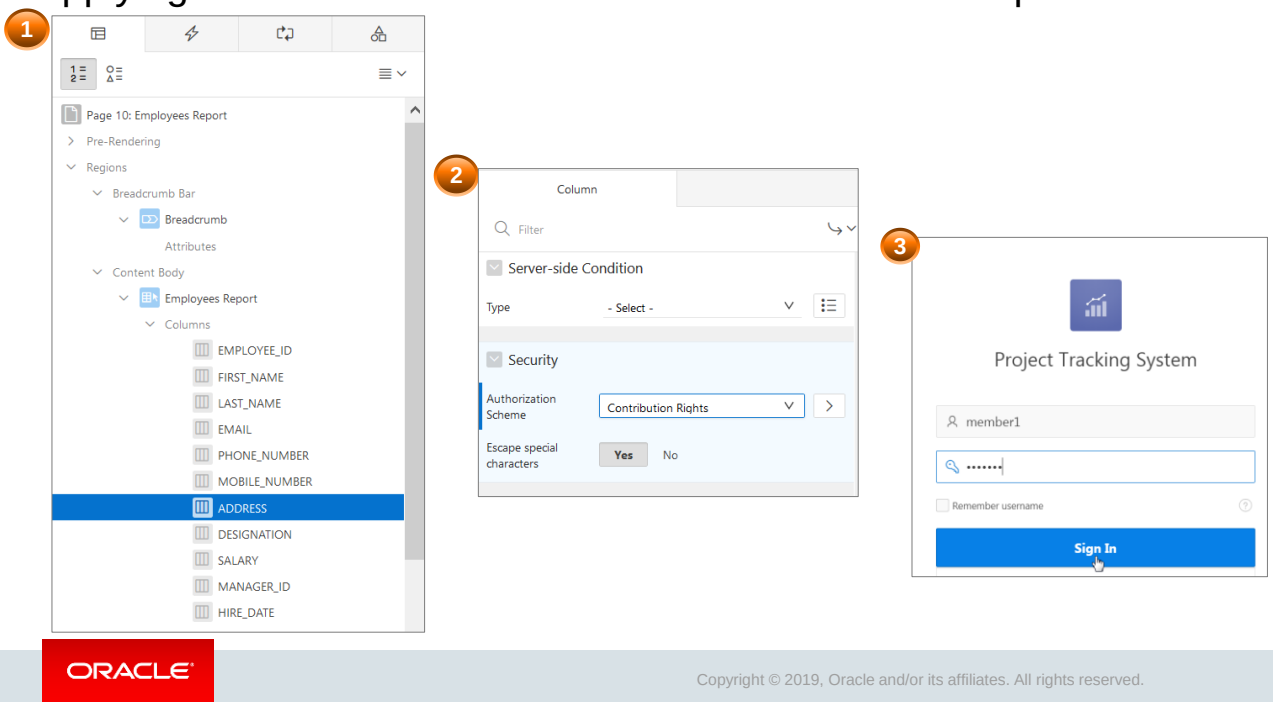
In the next slide, you learn how to apply an authorization scheme to a page in the application.



Now, instead of applying the same authorization scheme to all the pages of the application, Steve wants to apply different authorization schemes to different pages in his application. For example, he wants all the users defined in the Access Control List to have access to view the *Project Status Report*, but not everyone can have the edit rights on this page. Let's see how he does it.

1. Navigate to the page definition of the page to which the authorization scheme must be attached. In the Rendering pane, select the page name. Here, Steve selects *Project Status Report* (screenshot 1).
2. In the Property Editor, scroll down to the **Security** tab and select a scheme from the **Authorization Scheme** drop-down list. Steve selects **Reader Rights** (screenshot 2).
3. Save and run the page.
4. Now, log in with user details of the **member1** user (screenshot 3). You can display the *Projects Status Report* because Steve has selected **Reader Rights** as the authorization scheme for this page (screenshot 4).

Applying an Authorization Scheme to a Column in a Report

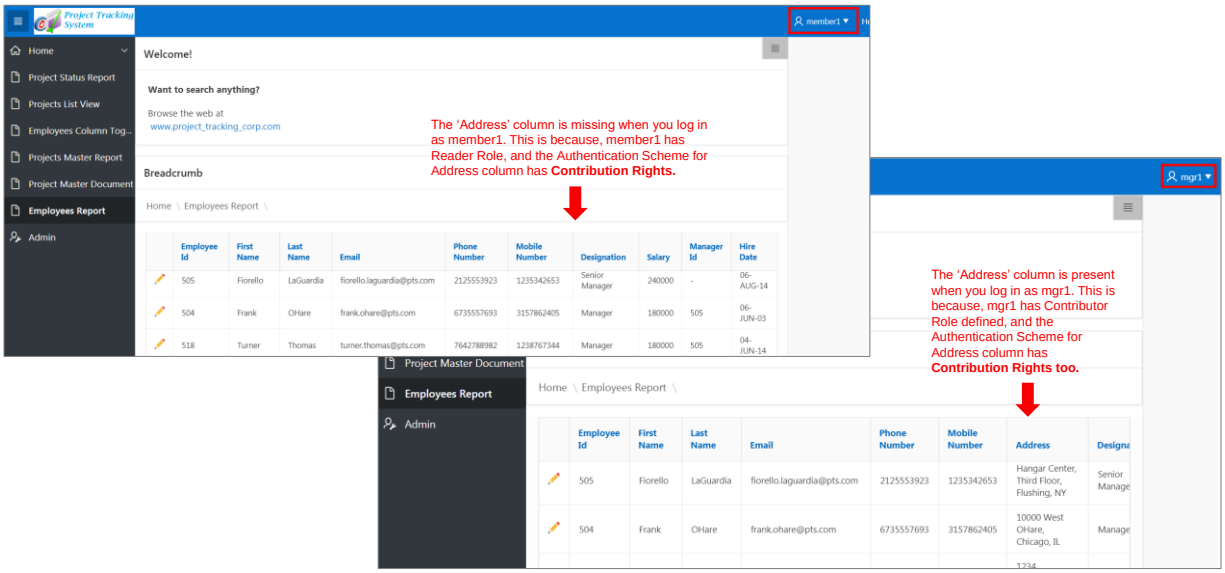


Now Steve wants to go a step ahead and decides to give restricted access to the *Address* column of his *Employees Report*. This is mainly because he wants only the managers to get the edit rights for creating or updating an employee's address details. Let's see how he applies an authorization scheme to a column in a report:

1. Navigate to the page definition of the page that contains the report. In the Rendering pane, select the column in the report where you want to apply the authorization scheme. In this slide example, Steve selects the **ADDRESS** column. This is because Steve wants the address to be a private column and not accessible to all the users.
2. In the Property Editor pane, scroll down to the **Security** tab and select a scheme from the **Authorization Scheme** drop-down list. Because Steve wants to give edit rights for this column only to the managers, he selects **Contribution Rights**. Click **Save**.
3. Now, log in with the Team Member's user details (**member1)** and try accessing the *Employees Report*.

   See the next slide to view the report.

Example: Applying an Authorization Scheme to a Column in a Report



You see that member 1 does not get to see the **Address** column in *the Employees Report*. This is because Steve has:

- Assigned **Reader Rights** to **member1** because he wanted the team members to have limited privileges to the application pages.
- Selected **Contribution Rights** in the **Authorization Scheme** for this column.

Now, if you log out and log in again with Manager's user details (**mgr1**), and run the *Employees Report* page, you get to see the **Address** column in the report. This is because Steve had assigned the contributors role to the Manager (**mgr1**) user.

Quiz

Which of the following statements are true about an authorization scheme?

a. You can attach an authorization scheme to any component or control in an application.
b. After associating an authorization scheme with a page, you cannot modify it.
c. You can create an authorization scheme through an Access Control page.
d. If a page-level authorization scheme fails, Oracle Application Express displays a previously defined message.

Answer: a, c, d

Practice 16-2 Overview: Restricting Users by Using Access Control

This practice covers the following topics:

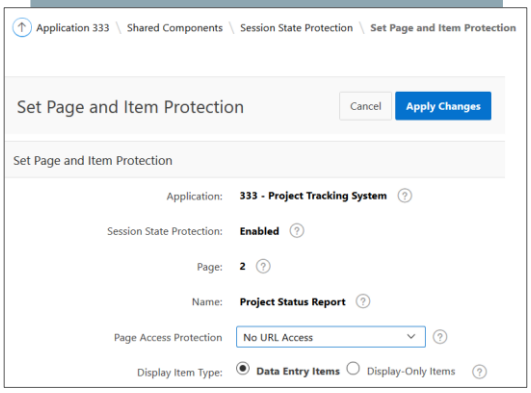
- Creating users to add to the Access Control list
- Creating an Access Control page
- Adding users to the Access Control List
- Defining and applying the authorization schemes to each application component

Lesson Agenda

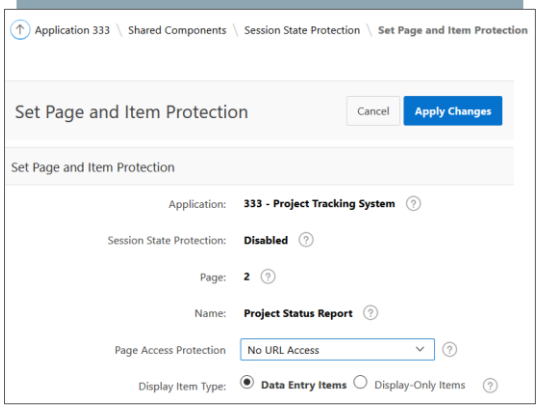
- Securing an Application
- Using Authentication Schemes
- Using Authorization Schemes
- Using Session State Protection
  - What Is Session State Protection?
  - Enabling Session State Protection
  - Configuring Session State Protection

Session State Protection

Session State Protection - Enabled

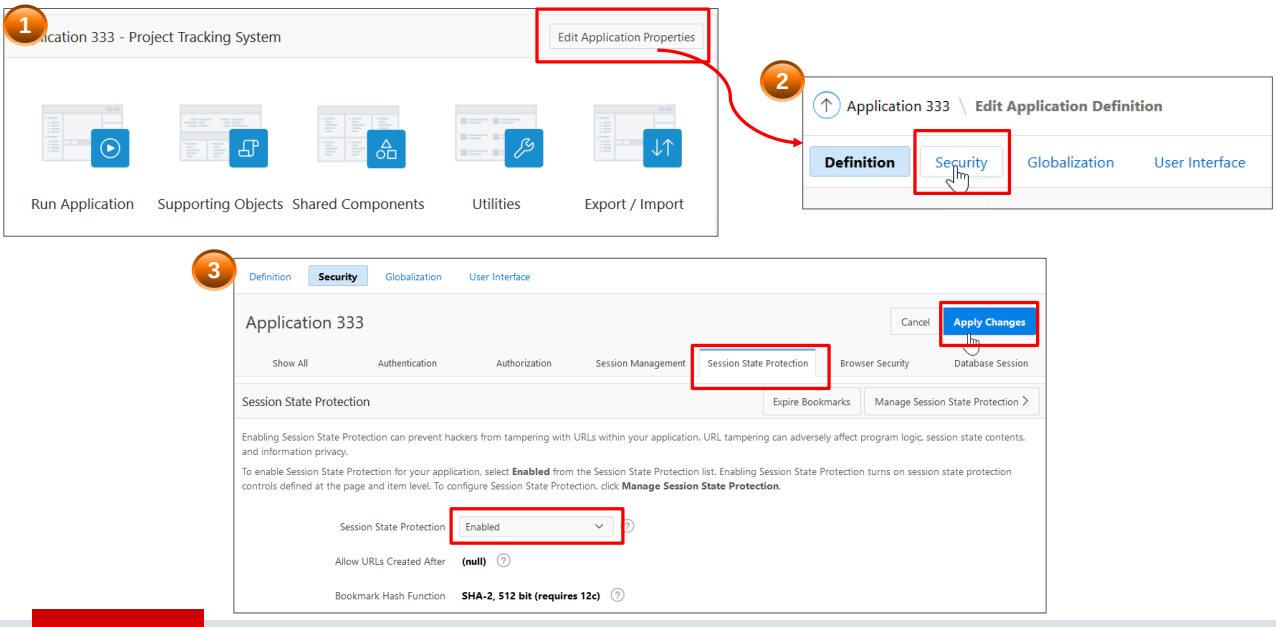


Session State Protection - Disabled



- Now Steve goes to his last requirement. He wants to secure his application by preventing hackers from tampering with the URLs in the application. This is because URL tampering can adversely affect program logic, session state contents, and information privacy of his application. And for that he uses the Session State Protection functionality of Oracle Application Express.
- When you enable Session State Protection for your application, it uses Page Access Protection attributes and Session State Protection item attributes together with checksums (an error detection method) positioned in f?p= URLs. (f?p= is a prefix used by Oracle Application Express to route the request to the correct engine process.) This prevents hackers from tampering with the URL and does not allow any unauthorized access and alteration of the session state.
- Note that when you disable Session State Protection for your application, the page and item attributes related to Session State Protection are ignored and checksums are not included in the generated `f?p= URLs`.
- For example, in the slide, you see that Steve has enabled Session State Protection for his PTS application and has set *No URL Access* for the Project Status Report page. Therefore, when he tries to run the application and then navigates to the *Project Status Report* page, he gets an error message stating, “This page cannot be invoked…”. However, when Steve disables the Session State Protection for his application, and navigates to the *Project Status Report*, he can still view the report (although the page protection remains as *No URL Access*).
- You can enable Session State Protection for your application both from:
  - The Edit Application Properties page and
  - The Session State Protection page
- In the next couple of slides, you will learn how.

Enabling Session State Protection from the Edit Application Page

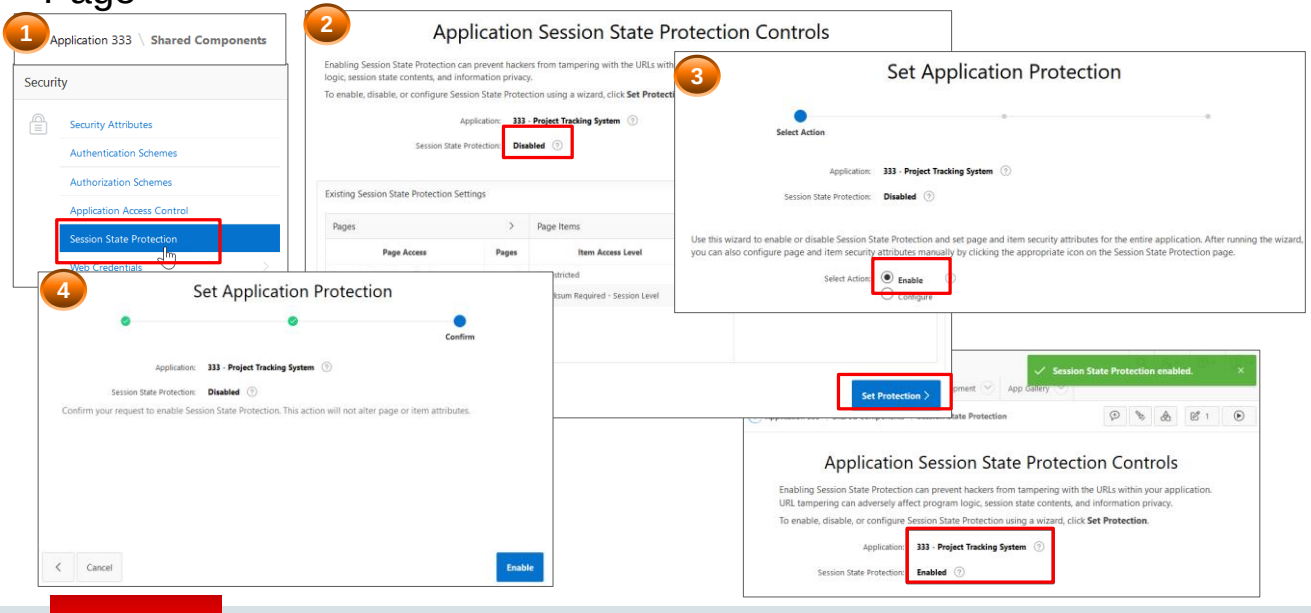


To enable Session State Protection for an application, perform the following steps:

1. Navigate to the application home page and click the **Edit Application Properties** button.
2. Click the **Security** tab and then the **Session State Protection** tab.
3. Select **Enabled** for Session State Protection and click **Apply Changes**.

Note: The Session State Protection is enabled by default. To disable Session State Protection, use the same procedure, but select **Disabled** instead of **Enabled**. Disabling Session State Protection will not change the existing security attribute settings, but those attributes will be ignored at run time.

Enabling Session State Protection from the Session State Protection Page



You can also access the **Session State Protection** page and then enable Session State Protection for the application. Perform the following steps:

1. Click the **Shared Components** icon on the application home page.
2. Click the **Session State Protection** link in the Security list.
3. The Session State Protection page appears. Click the **Set Protection** button.
4. Select **Enable** and click **Next**.
5. Click the **Enable** button.

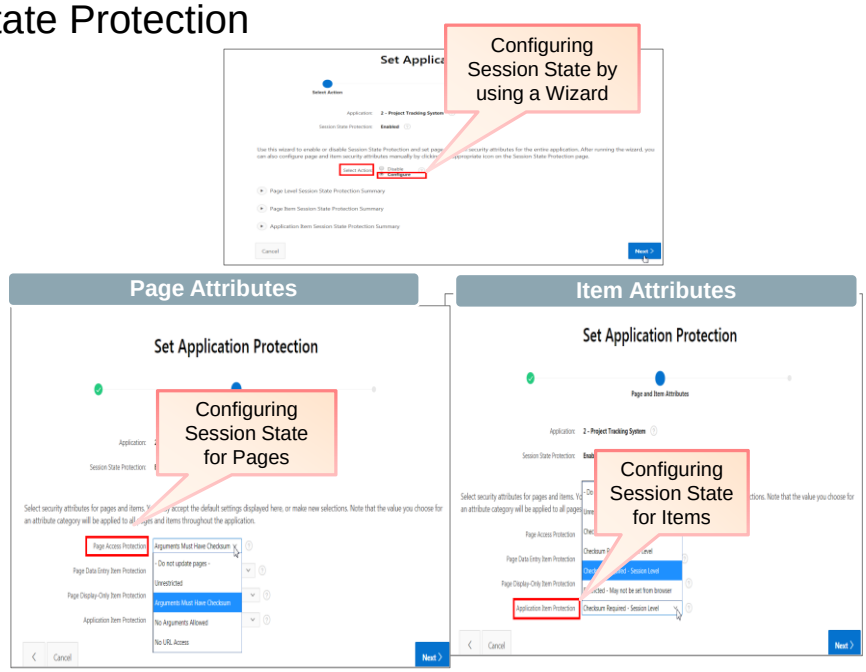
Session State Protection for your application is now **Enabled**.

In the next few slides, you will learn how to configure Session State Protection for your application pages and items.

Configuring Session State Protection

You can configure security attributes in two ways:

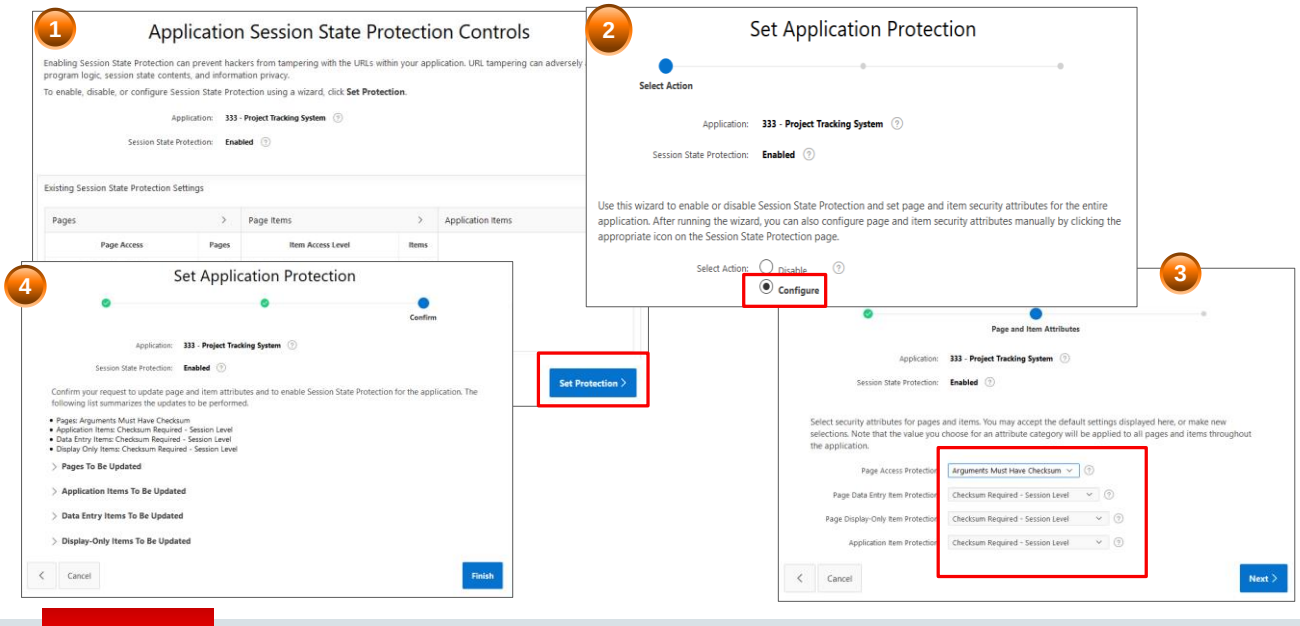
- Use a wizard and select a value for specific attribute categories. Those selections are then applied to all pages and items within the application.
- Configure values for individual pages, items, or application items.



After enabling Session State Protection, Steve now goes to the next step, that is, to configure security attributes. You can configure security attributes in two ways as mentioned in the slide.

In the next slide, Steve configures Session State Protection for his application by using a wizard. Let's see how.

Configuring Session State Protection by Using a Wizard



To configure Session State Protection, perform the following steps:

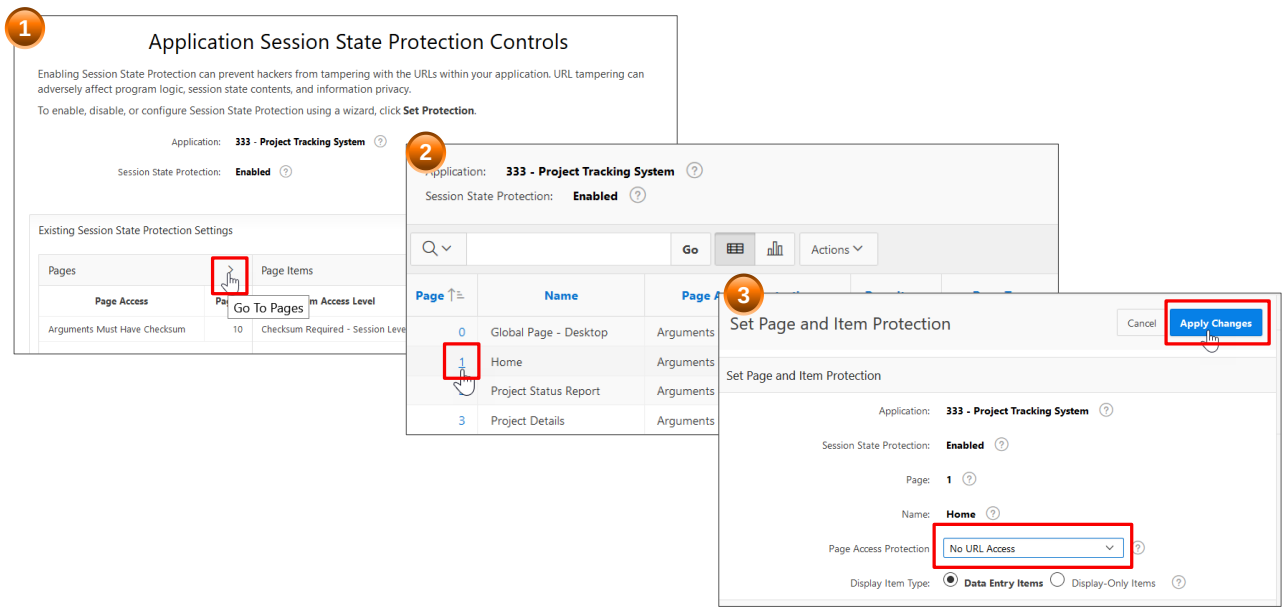
1. Navigate to the Session State Protection page and click the **Set Protection** button.
2. The Session State Protection Wizard appears. Select **Configure** and click **Next**.
3. Select the security attributes for application pages, application items, and page items. Click **Next**. To specify the way a page or an application item's session state value can be set, you have the following options:
   - **Checksum Required – Application Level**: Use this option when you want to allow the item to be set only by URLs having checksums that were generated by any user running the same application in the current workspace but in a different session.
   - **Checksum Required – User Level**: Use this option when you want to allow the item to be set only by URLs having checksums that were generated by the same named user, running the same application in the current workspace, but in a different session.
   - **Checksum Required – Session Level**: Use this option when you want to allow this item to be set only by URLs having checksums that were generated in the current session.

   These selections are then applied to all pages and items within the application.
4. Review the attributes and click **Finish**.

The security attributes are applied to all pages and items within the application.

In the next slide, you learn how to configure Session State Protection for the pages and items in your application by selecting each individual page.

Configuring Session State Protection for Pages and Items



To configure Session State Protection for pages, perform the following steps:

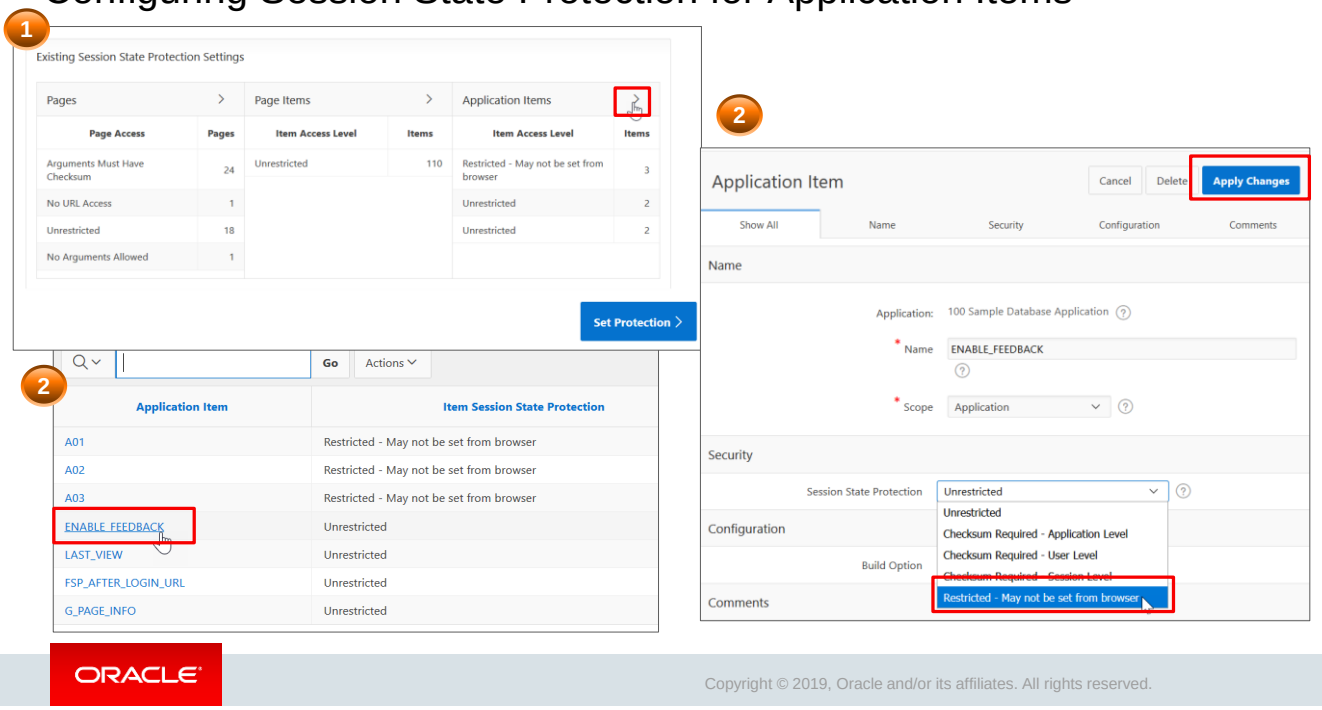
1. Navigate to the Session State Protection page and click the arrow next to **Pages**.
2. A report displays all the pages in the application and the security attribute set for the page. To set the security attribute for a page, click the page number link for the page.
3. You can now set the security attribute for the page. The page items for the page are also listed, and you can set the attributes for each item. The following **Page Access Protection** attributes are available for pages:
   - **Unrestricted:** The URL to request the page may or may not have session state arguments.
   - **Arguments Must Have Checksum:** If the session state arguments appear in the URL, a checksum must also be provided.
   - **No Arguments Allowed:** The URL used to request the page must not contain session state arguments.
   - **No URL Access:** The page may not be accessed by using a URL. However, the page may be the target of a branch to Page branch type, which does not redirect the user to a URL. (You will be doing this in one of your practices.)

   In this example, Steve selects **No URL Access**. This is because in this example he is working on the Home Page of his application and he does not want this page to be easily accessible to everybody. Click **Apply Changes** to save the settings.

If you click the Page Item icon on the Session State Protection page, a report displays all the page items in the application. You can click a particular item and set the attributes for that item.

In the next slide, let's see how to configure Session State Protection for the application Items.

Configuring Session State Protection for Application Items



You have learned in an earlier lesson in Unit 2 that application items are not associated with a page and, therefore, have no user interface properties. You can use an application item as a global variable. Application items are named session state variables that are not specific to a particular page. To configure Session State Protection for application items:

1. On the Session State Protection page, click the **Application Items** icon. A report is displayed listing all the application items for the application.
2. Click the Application Item link that you want to configure.
3. On the **Security** tab, select the **Session State Protection** for your application item.
4. Click **Apply Changes**.

Practice 16-3 Overview: Enabling Session State Protection

This practice covers the following topics:

- Setting Page Access Protection to No URL Access
- Creating a branch without passing the URL

Summary

In this lesson, you should have learned how to:

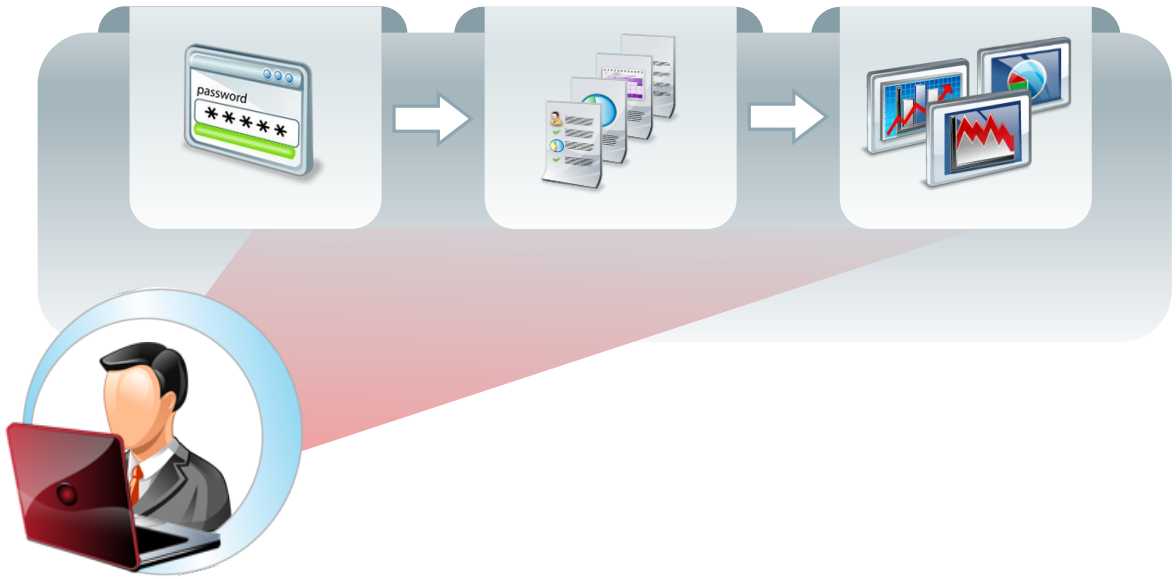
- List the different ways to secure your application
- Differentiate between authentication and authorization
- Create an authentication scheme for your application
- Create an authorization scheme by using Access Control
- Enable and configure Session State Protection

In this lesson, you learned how to implement security for your application. You learned how to associate an authentication scheme with your application and also how to create and attach an authorization scheme to your application. You also learned how to enable Session State Protection and configure security attributes.

17

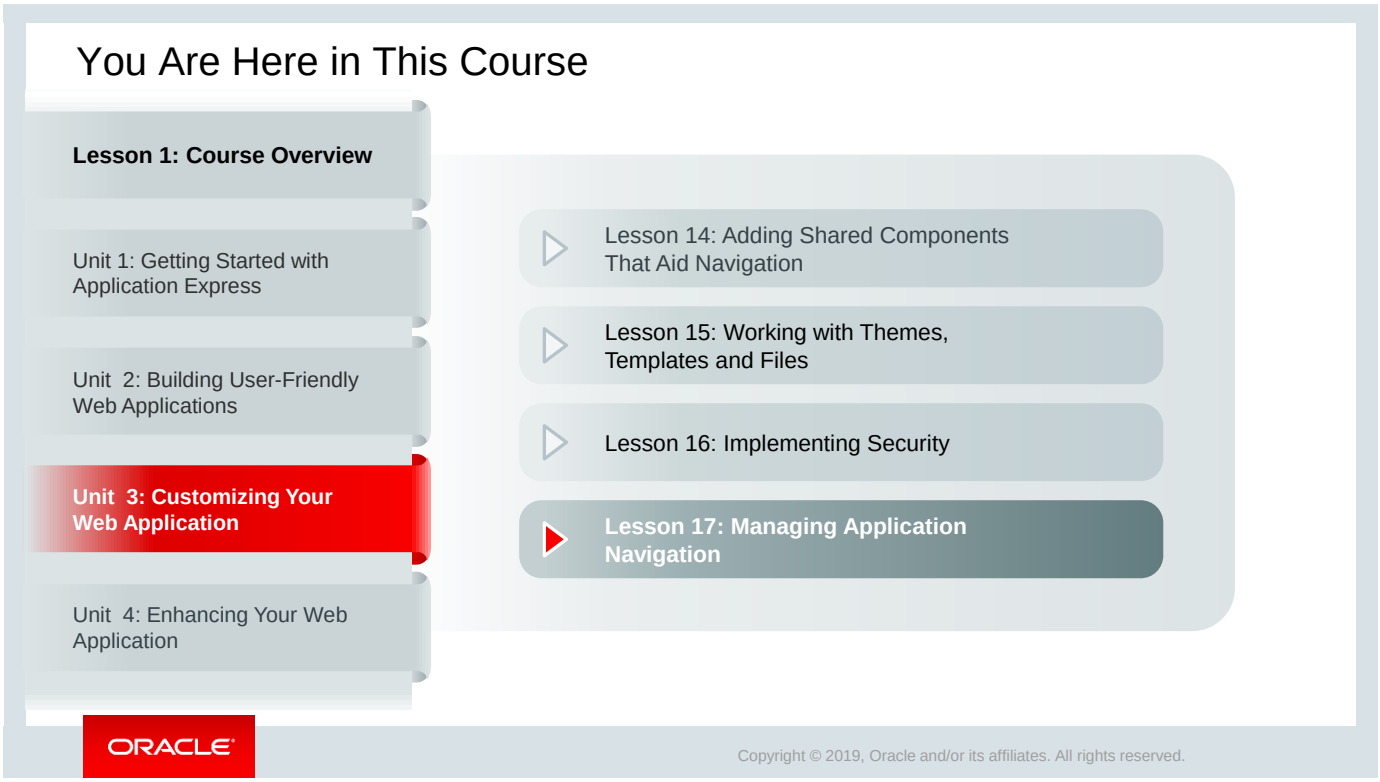
Managing Application Navigation

Steve Tries to Improve Application Navigation



Steve successfully added navigation to PTS application so that any page can be run easily. However, while working with different options in Oracle Application Express, he understood that Oracle Application Express allows developers to create visual and attractive navigation lists with images.

Along with improving the user experience, Steve also wants to give restricted access to his navigation lists so that only the users who have the assigned authorization get to view the list of pages.



This slide is a graphical depiction of the course, particularly highlighting Unit 3 - Lesson 17, which is dealt with in these slides.

Objectives

After completing this lesson, you should be able to:

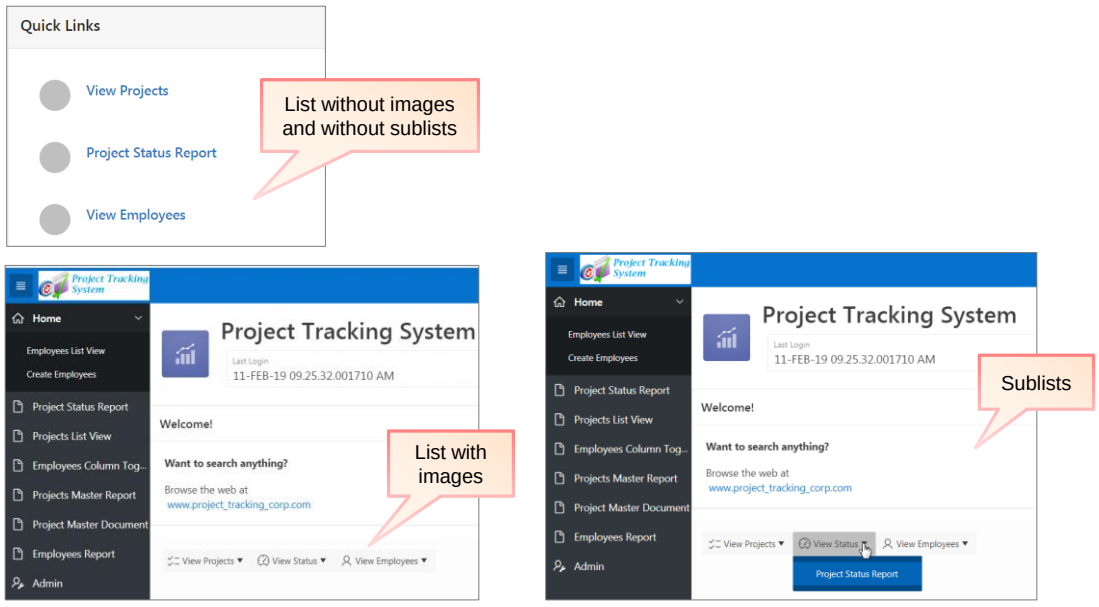
- Build a hierarchical list with images
- Build a database-driven navigation report
- Build a site map
- Enforce authorization on the site map

In this lesson, you learn to build a hierarchical list with images on the home page. You also build a database-driven navigation report and a site map and incorporate security into your site map.

Lesson Agenda

- Building a Hierarchical List with Images
- Building a Database-Driven Navigation Report
- Building a Site Map
- Enforcing Authorization on Your Site Map

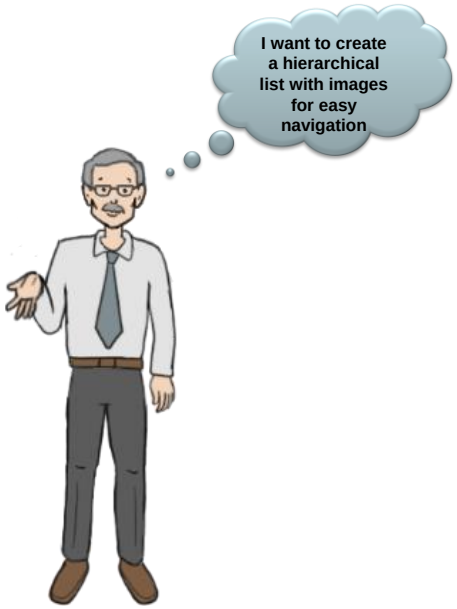
Building a Hierarchical List with Images



As you know, Steve had created a static list in the form of *Quick Links* on his application's home page (in one of the earlier lessons on *Lists and Navigation*). But now he feels that one way to handle a better navigation experience for his users would be if he can change it into a hierarchical list (or sublists) with images.

In the slide above, you see the original navigation and also how the images can be used and the sublists displayed. You will learn how to create a hierarchical list with images in the next few slides.

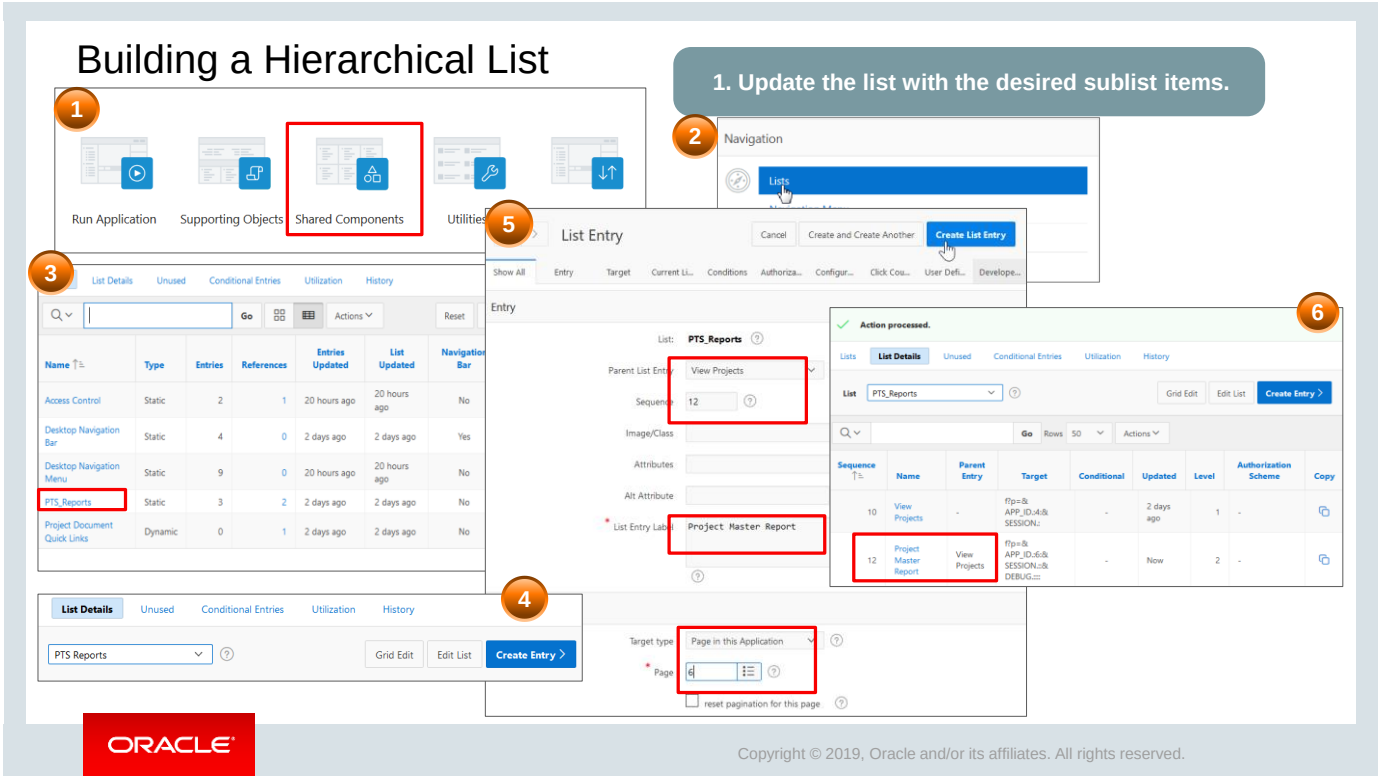
Building a Hierarchical List with Images



1. Update the list with the desired sublist items.
2. Associate an image or icon with a list item.
3. Associate the list with the list region and define region settings.

Steve had created a static list called *PTS Report* earlier, and the *PTS Report* also included list entries for Projects, Status, and Employees report pages. Now Steve wants to display these list items hierarchically and update with the related sublists. This way, instead of having a long list of reports he had earlier, he would have only a few list items (*Projects*, *Status*, *Employees*) and all the reports related to that list under it. For example, *View Status* list (you saw in the previous slide) will include all reports related to the status of the projects, employees, etc. Later, to give a better look and feel, he would assign images to the list items and associate the list with the list region (that is, where he wants the list to be displayed).

In the next slide, let us see how he first updates the list with the sublist items.

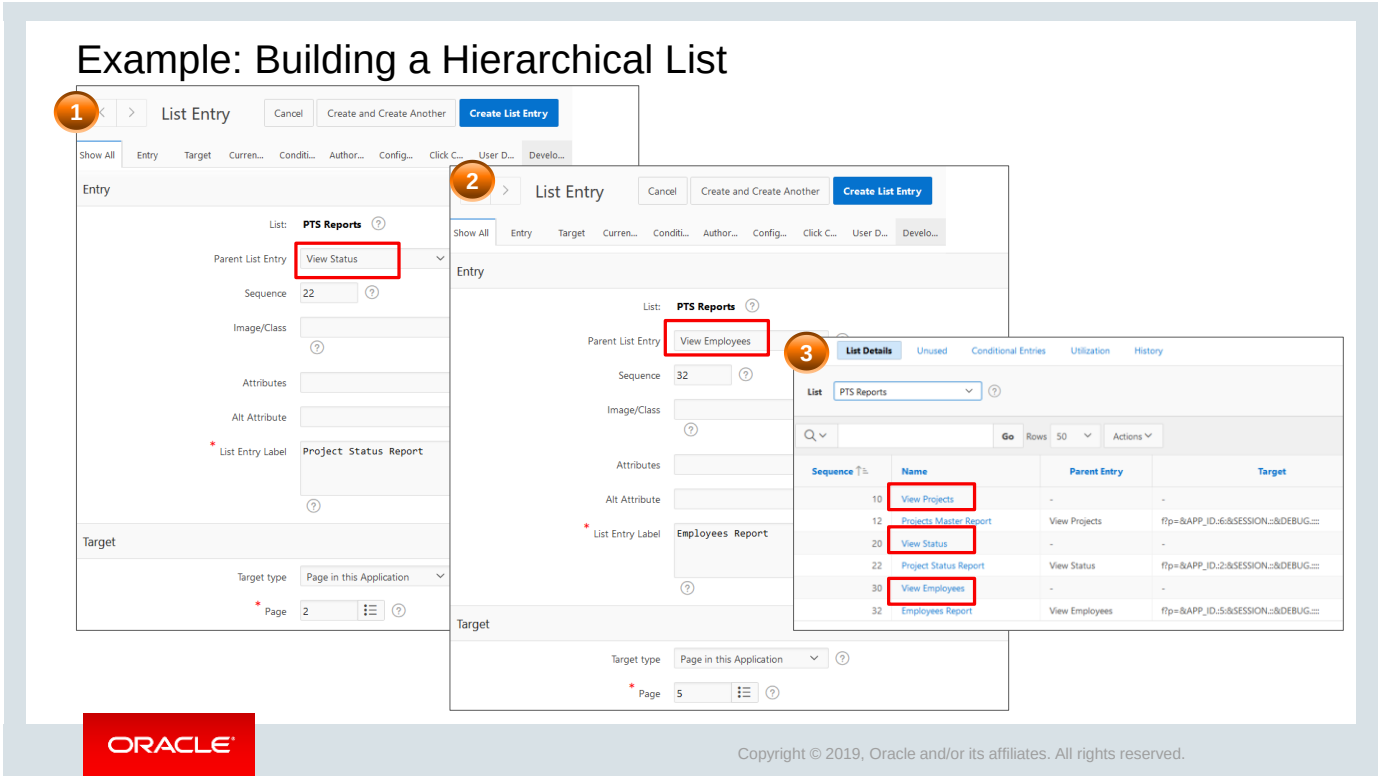


The first step is to update the list with the desired sublist items. Steve first creates a sublist item called *Projects Master Report* (the parent list is *View Projects*). Let's see how:

1. On your application page, click **Shared Components** (screenshot 1).
2. On the **Shared Components** page, click **Lists** under **Navigation** (screenshot 2).
3. Select the list that exists or create one. Here Steve selects PTS Reports (screenshot 3). He had created this earlier when he was creating a static list region on the application's home page.
4. Click **Create Entry** (screenshot 4).
5. Enter the following values and click **Create List Entry** (screenshot 5)
    - **Parent List Entry**: Select **View Projects**
    - **Sequence**: Enter **12**
    - **List Entry Label**: Enter `Projects Master Report`
    - **Target type***:* Select **Page in this Application**
    - **Page***:* Select **6** from the pop-up LOV (this is the page number of *Projects Master Report* page)

The new list entry is created. Now, in the next slide, Steve creates an entry for each item he wants to include in the list. Specify a parent list entry (where appropriate) and a page to branch to when the entry is selected.

Note: The best practice is to sequence each entry by parent list entry and stagger the numbers in case a new list entry needs to be added at a later date.



As mentioned in the previous slide, Steve now creates the other two sublist items:

- *Project Status Report* (*View Status* is the parent list)
- *Employees Report* (*View Employees* is the parent list)

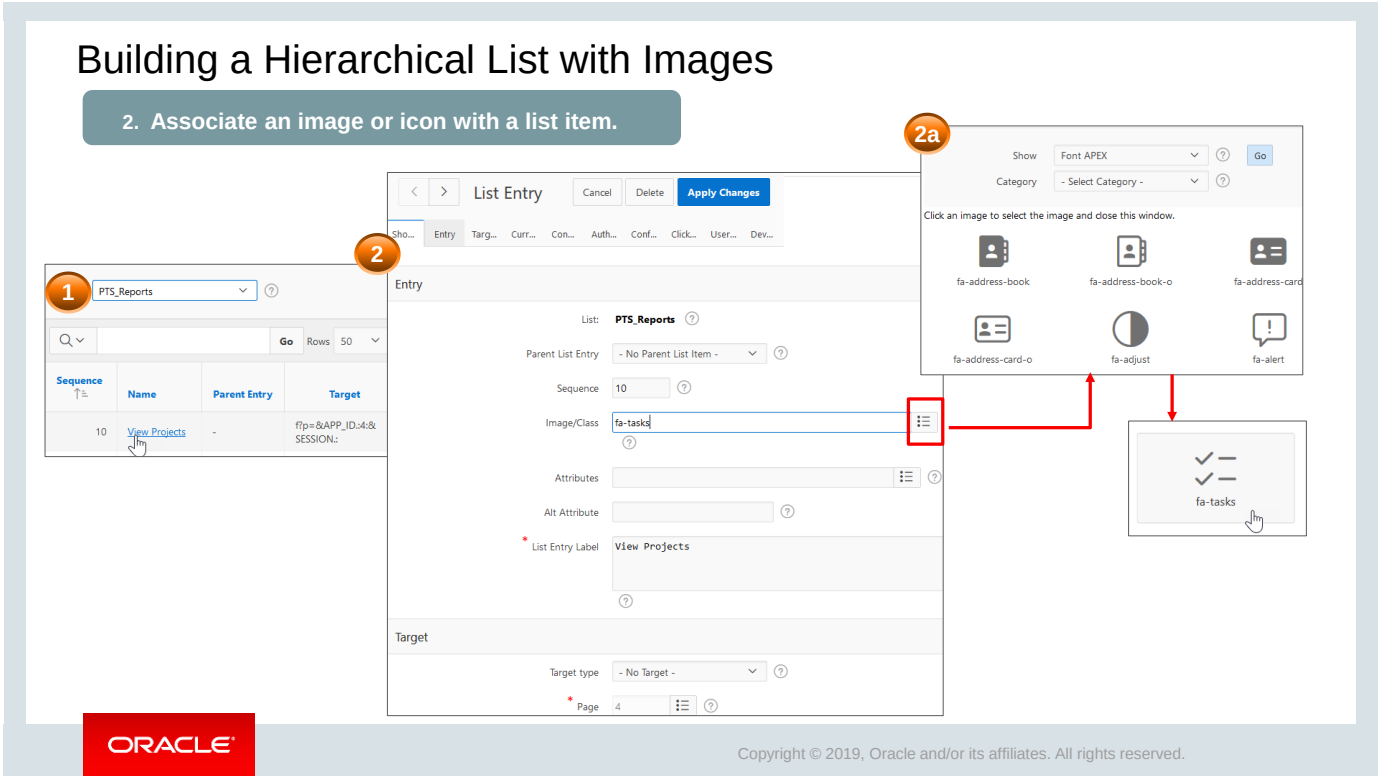
Let's see how. You first follow the same steps as in the previous slide and:

1. Enter the following values and click **Create List Entry** (screenshot 1)
   - **Parent List Entry**: Select **View Status**
   - **Sequence**: Enter `22`
   - **List Entry Label**: Enter `Project Status Report`
   - **Target type***: Select* **Page in this Application**
   - **Page***:* Select **2** from the pop-up LOV (this is the page number of the *Project Status Report* page)

   The new sublist entry (*Project Status Report*) is created.
2. Enter the following values and click **Create List Entry** (screenshot 2)
   - **Parent List Entry**: Select **View Employees**
   - **Sequence**: Enter `32`
   - **List Entry Label**: Enter `Employees Report`
   - **Target type***:* Select **Page in this Application**
   - **Page***:* Select **10** from the pop-up LOV (this is the page number of *Employees Report* page)

   The new sublist entry `Employees Report` is created.

You can see all the three sublist items displayed under the *PTS Reports* List (screenshot 3).



Now, Steve wants to associate an image or icon with a list item to give his list items a better and modern look and feel. Steve starts with *View Projects* list item and then he adds icons for the other two list items *View Status* and *View Employees*. Let's see how.

From your application's home page, click **Shared Components**. On the Shared Components page, click List and select your list. Here Steve selects *PTS Reports*.

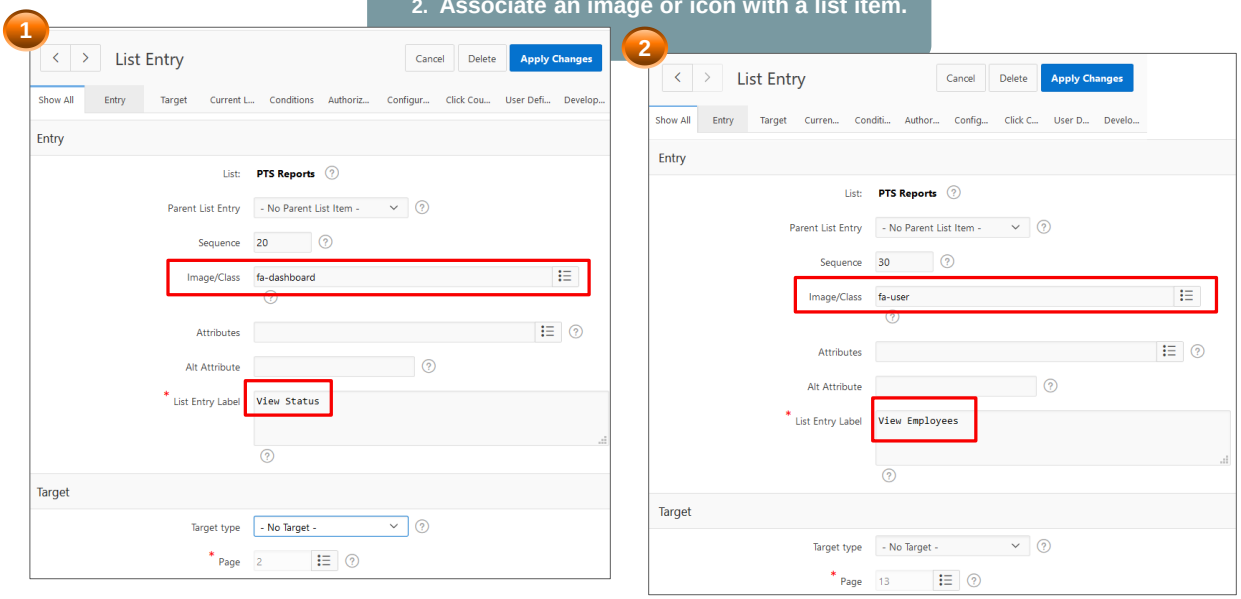
1. Click the list entry for which you want to associate an image or icon to edit its properties. Steve starts with the *View Projects* List entry.
2. Click the pop-up LOV beside the Image/Class field and browse for the font APEX icon. Here, Steve selects *fa-tasks*.
3. Select `-No Target-` for **Target Type**. This is because he has already created sublist item *Projects Master Report* (with links to the related page) for *View Projects* (in the previous slides).
4. Click **Apply Changes**.
5. Steve repeats the same for *View Status* and *View Employees* list items.

So, now, all the three items (View Projects, View Status, and View Employees) under *PTS Reports* have icons assigned to them.

In the next slide, Steve performs the last step of defining list region settings for the list items that he created. This is to select where exactly he wants the list items to be displayed.

Building a Hierarchical List with Images

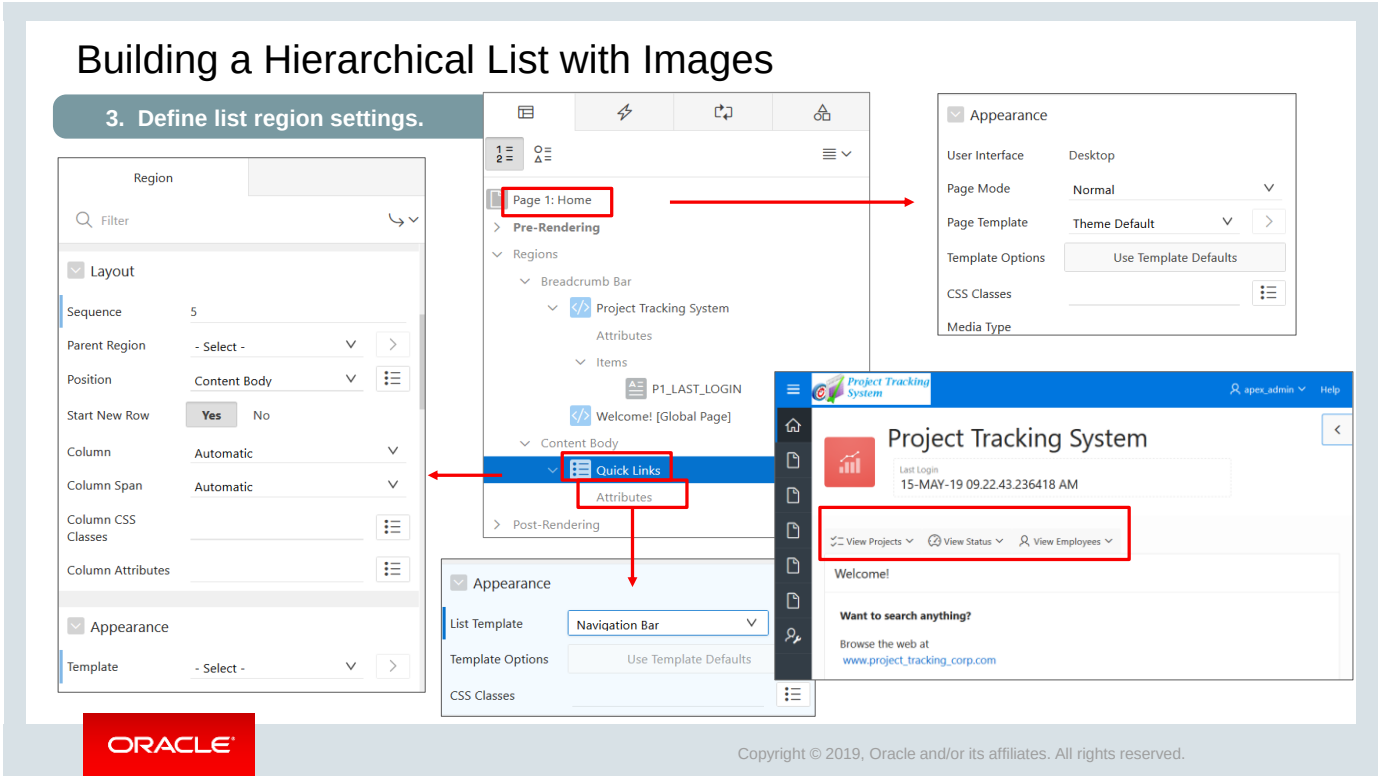
2. Associate an image or icon with a list item.



Steve repeats the same for *View Status* (screenshot 1) and *View Employees* (screenshot 2) list items.

So, now, all the three items (View Projects, View Status, and View Employees) under *PTS Reports* have icons assigned to them.

In the next slide, Steve performs the last step of defining list region settings for the list items that he created. This is to select where exactly he wants the list items to be displayed.



As you know, Steve had already created *Quick Links* region in the home page of his application for his *PTS Reports*. He would now define the list region settings to display the hierarchical list and sublists that he created in the last few slides.

1. Open Home page (where the list region is created) in Page Designer view, and in the Rendering tab, select the **Quick Links** region.
2. In the Property Editor, make the following changes:
   - **Layout > Sequence**: Enter 5
   - **Layout > Position**: Select **Content Body** (this is where Steve wants the list to be displayed)
   - **Appearance > Template**: Select **– Select –** (this shows that you have selected No Template)
3. In the Rendering tab, select **Quick Links > Attributes.** In the Attributes tab, change the **List Template** value to *Navigation Bar*.
4. In the Rendering tab, select **Page 1: Home**. In the Page tab, make sure that the **Appearance > Page Template** value is set to *Theme Default*.
5. Run the page to see the Home Page List. Click the parent entry to see the sublist entries.

Practice 17-1 Overview: Building a Hierarchical List with Images

This practice covers the following topics:

- Updating the existing list
- Associating parent list entries with font awesome icons
- Changing the attributes of the navigation region

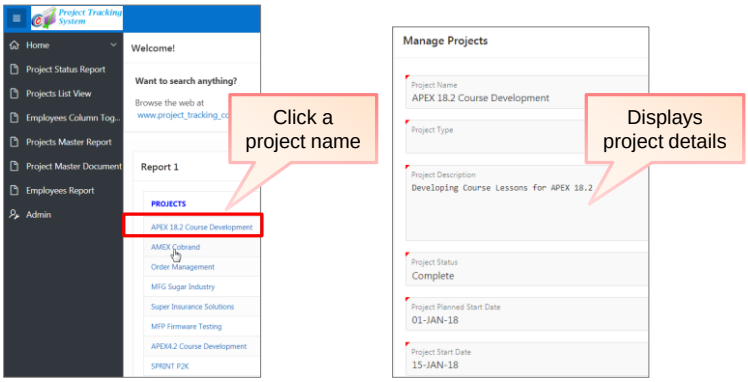
Lesson Agenda

- Building a Hierarchical List with Images
- Building a Database-Driven Navigation Report
- Building a Site Map
- Enforcing Authorization on Your Site Map

Building a Database-Driven Navigation Report

This report is used to navigate between pages by using links defined against values in the database.

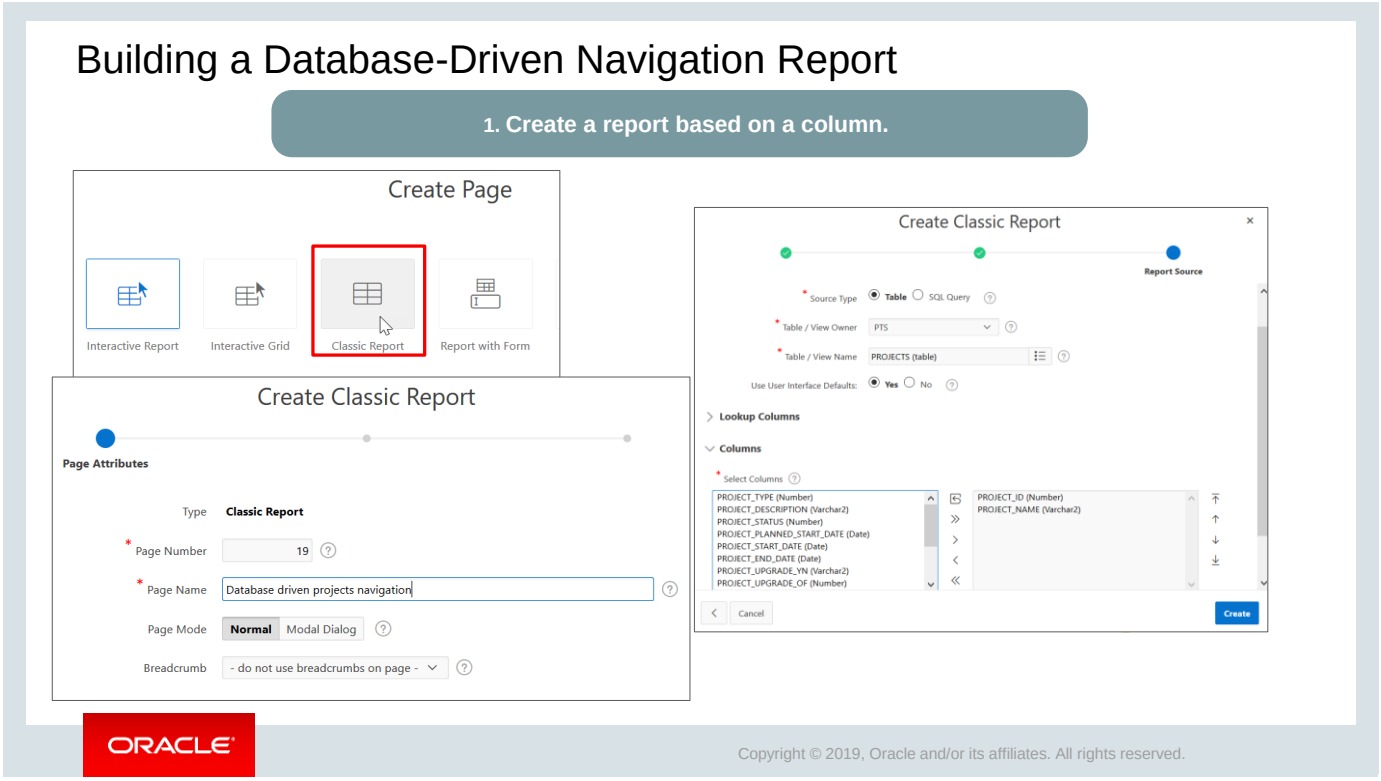
1. Create a report based on a column.
2. Create a link to page and pass an ID value.



Often you want to handle navigation through shortcuts. In this case, Steve wants to now display a list based on values in the database. In this example, a list of Projects shortcuts is shown. The user selects a project, which populates the project detail page based on the project she/he selects. This navigation between pages based on a value in the database is done using a report.

The report also selects the ID column (in this case, `PROJECT_ID`), which is then passed to the linked page so that the page can be populated. In the example in this slide, the user selected the project name, so the `PROJECT_ID` is passed to the Manage Projects page, and the information for the project is displayed.

In the next few slides, you see how Steve creates a navigation-based report for quick access to project information in PTS. But first, he creates a report based on a column. Let's see how in the next slide.

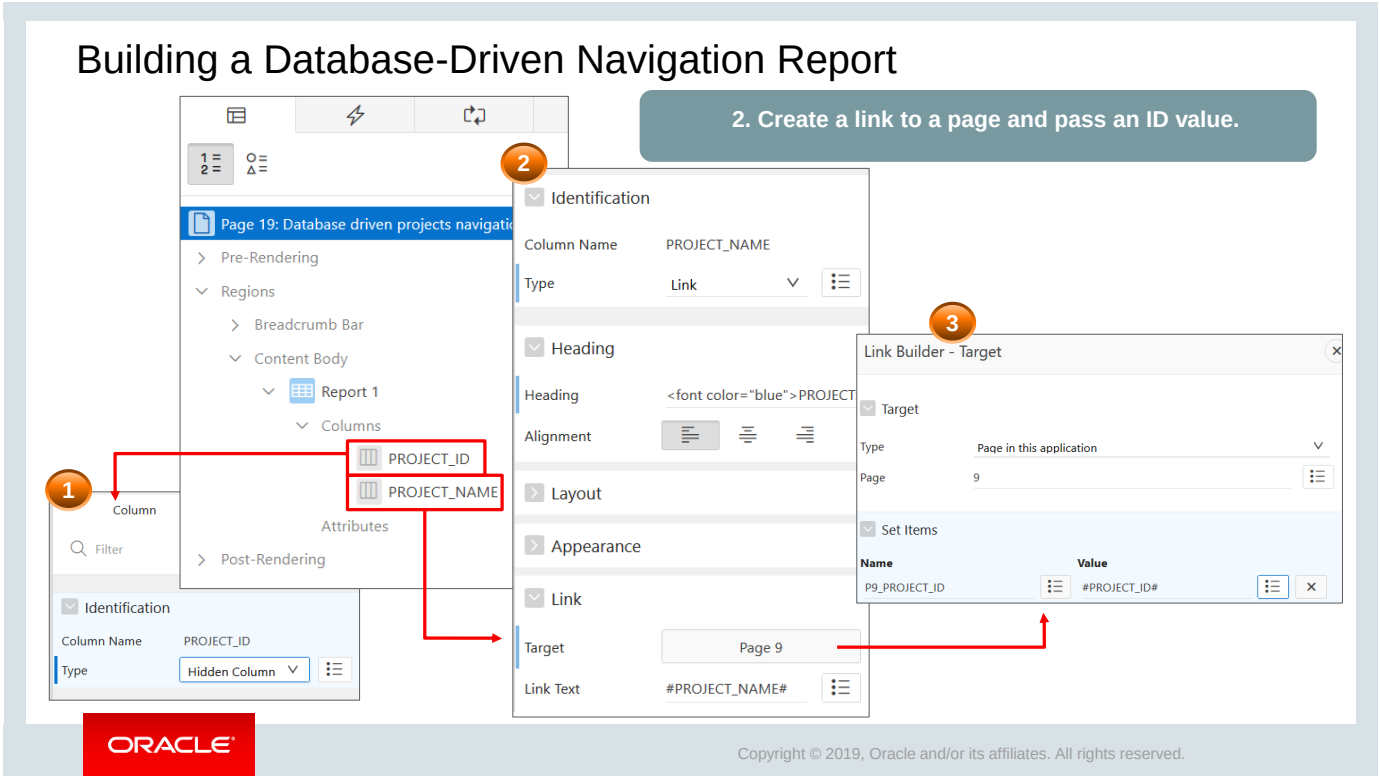


Steve first creates a report and selects `PROJECT_NAME` and `PROJECT_ID` columns. Let's see how.

You open your application's home page and click **Create Page**. You then perform the steps below:

1. Select **Report** and select **Classic Report**.
2. Enter the following values and click **Next** by retaining default values for other fields:
   - **Page Number:** Enter 19
   - **Page Name:** Enter Database driven projects navigation
3. For **Navigation Preference**, accept defaults and click **Next**.
4. Enter the following values and click **Create**.
   - **Source Type:** Select **Table**
   - **Table/View Owner:** Select **PTS**
   - **Table/View Name:** Select `PROJECTS (table)`. All the columns belonging to the *`PROJECTS (table)`* are displayed. Select all fields except `PROJECT_ID` and `PROJECT_NAME` and move them to the left using < symbol.

The Report page is created and opens in Page Designer view.

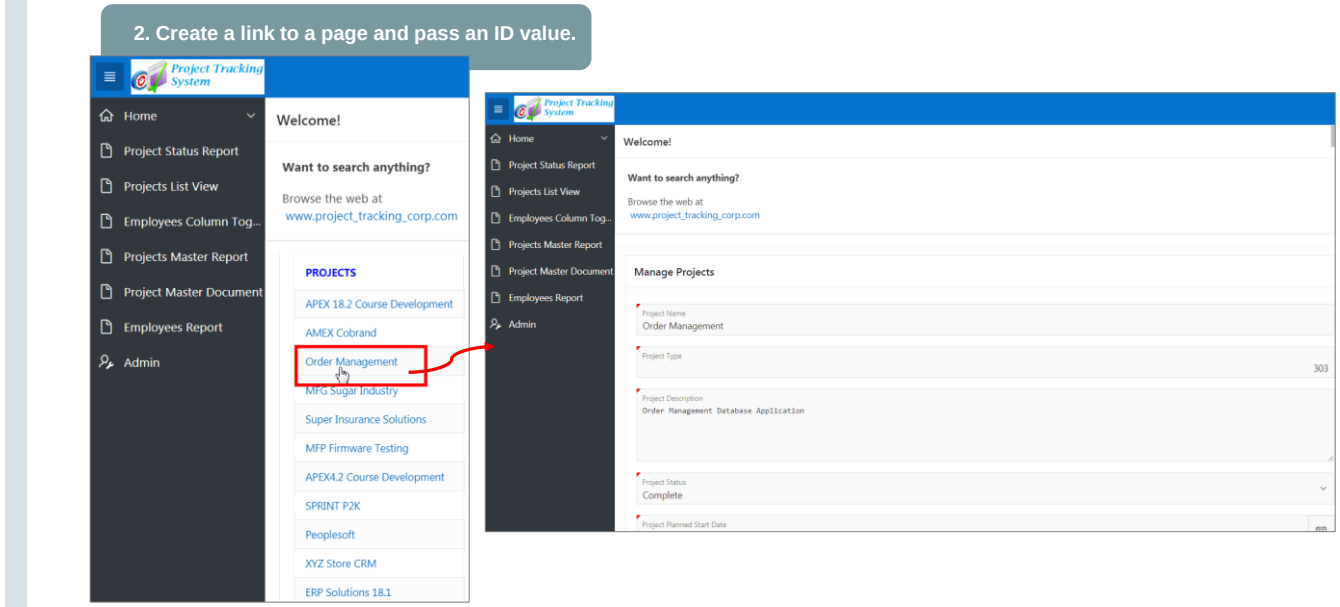


Now that Steve has already created the report, he wants to create a link from the `PROJECT_NAME` column to the *Manage Projects* page (which has all the project details) by passing the `PROJECT_ID` value.

Let's see how the project names are displayed as links.

1. Under **Database driven projects navigation > Columns** region, select `PROJECT_ID`
2. In the Property Editor on the right side, change the **Type** to *Hidden Column*.
3. Under **Database driven projects navigation > Columns** region, select `PROJECT_NAME` and enter the following values in its property editor and click **OK**.
    - **Type**: Select **Link**
    - **Heading**: Enter `<font color="blue">PROJECTS</font>`
    - **Target** (under Link): Select **9** (selected from Popup LOV). This is the *Manage Projects* page number.
    - **Name**: Select `P9_PROJECT_ID`
    - **Value**: Select `#PROJECT_ID#`
    - **Link Text**: By Default `#PROJECT_NAME#`

Building a Database-Driven Navigation Report



Click Save and Run Page icon on top right corner, enter login credentials (if prompted for), and view the report. You see Project Names displayed in 'Blue' color font as links. Upon clicking any project name, it takes to *Page 9: Manage Projects* page, displaying all its details.

Quiz

You can copy list entries from one list to another.

a. True

b. False

Answer: a

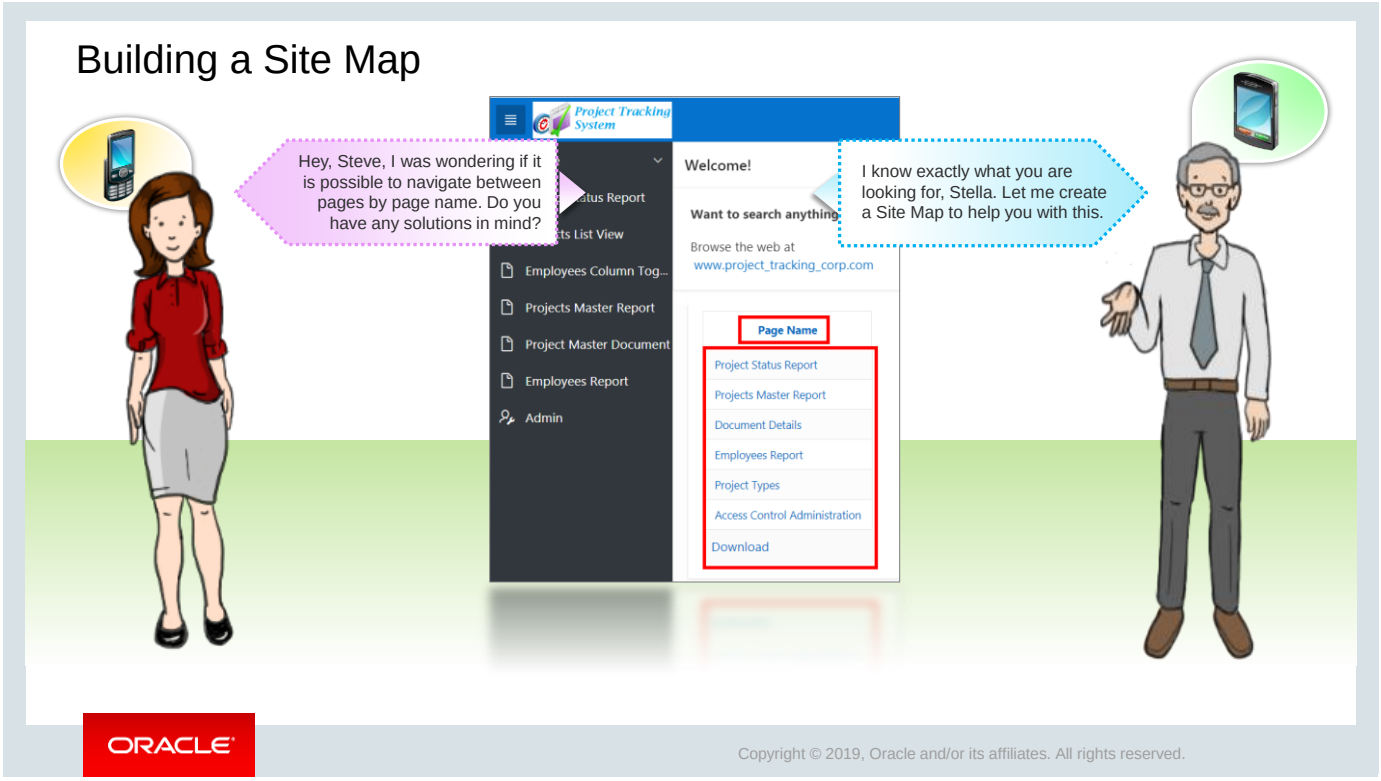
Practice17-2 Overview: Building a Database-Driven Report

This practice covers the following topics:

- Building a report based on the data in a table
- Navigating to the details

Lesson Agenda

- Building a Hierarchical List with Images
- Building a Database-Driven Navigation Report
- Building a Site Map
- Enforcing Authorization on Your Site Map



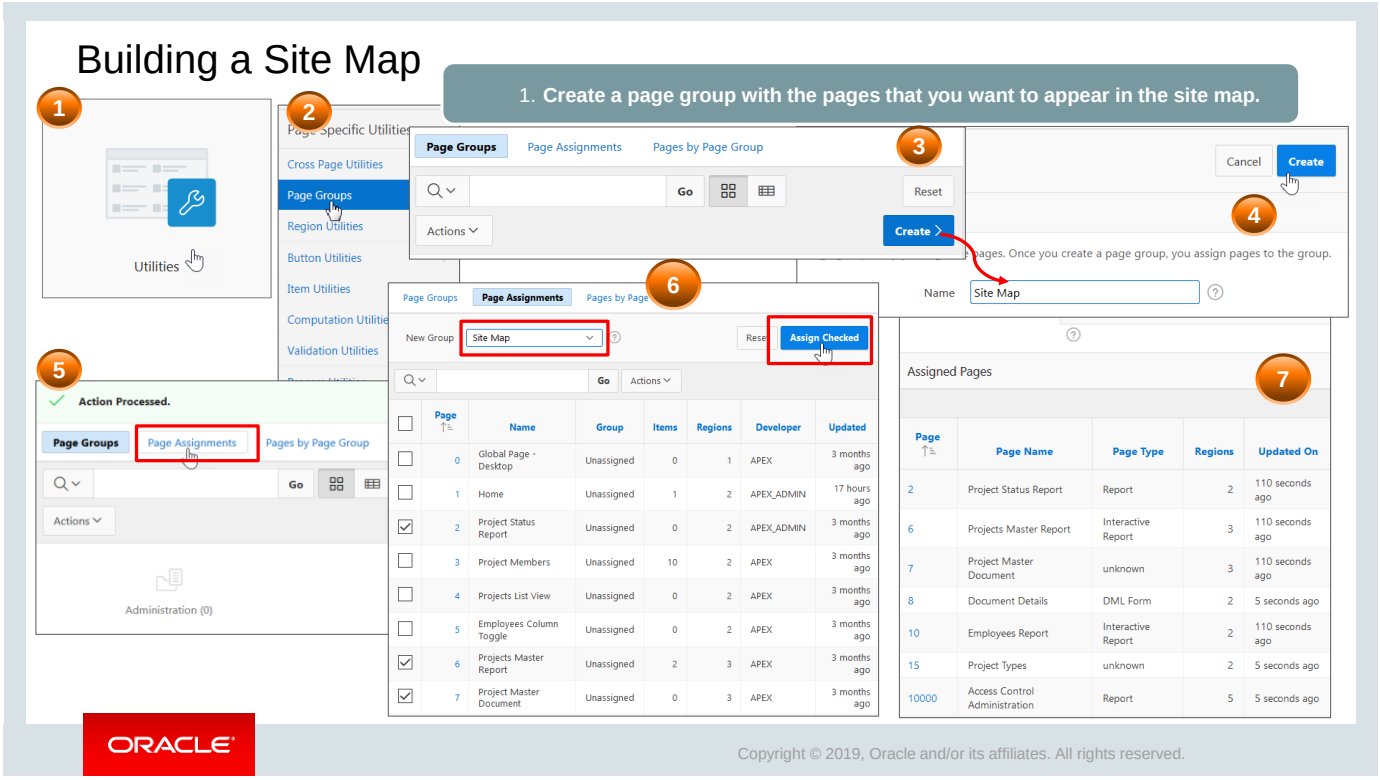
PTS now is a fully developed application and comprises many pages and reports, making navigating from one page to another a difficult task. Stella, the project manager, now places a new requirement for Steve. She wants an option where she can easily navigate from one page to another by clicking on the page title/page name. Steve finds this requirement interesting, and he immediately thinks about the Site Map feature of Oracle Application Express.

Site maps are typically useful for smaller applications. For larger online transaction processing (OLTP) applications, site maps may not be as useful because there are many pages that perform similar functionality. However, Steve can safely try the Site Map feature of Oracle Application Express on his PTS application.

To create a site map, you must perform the following tasks:

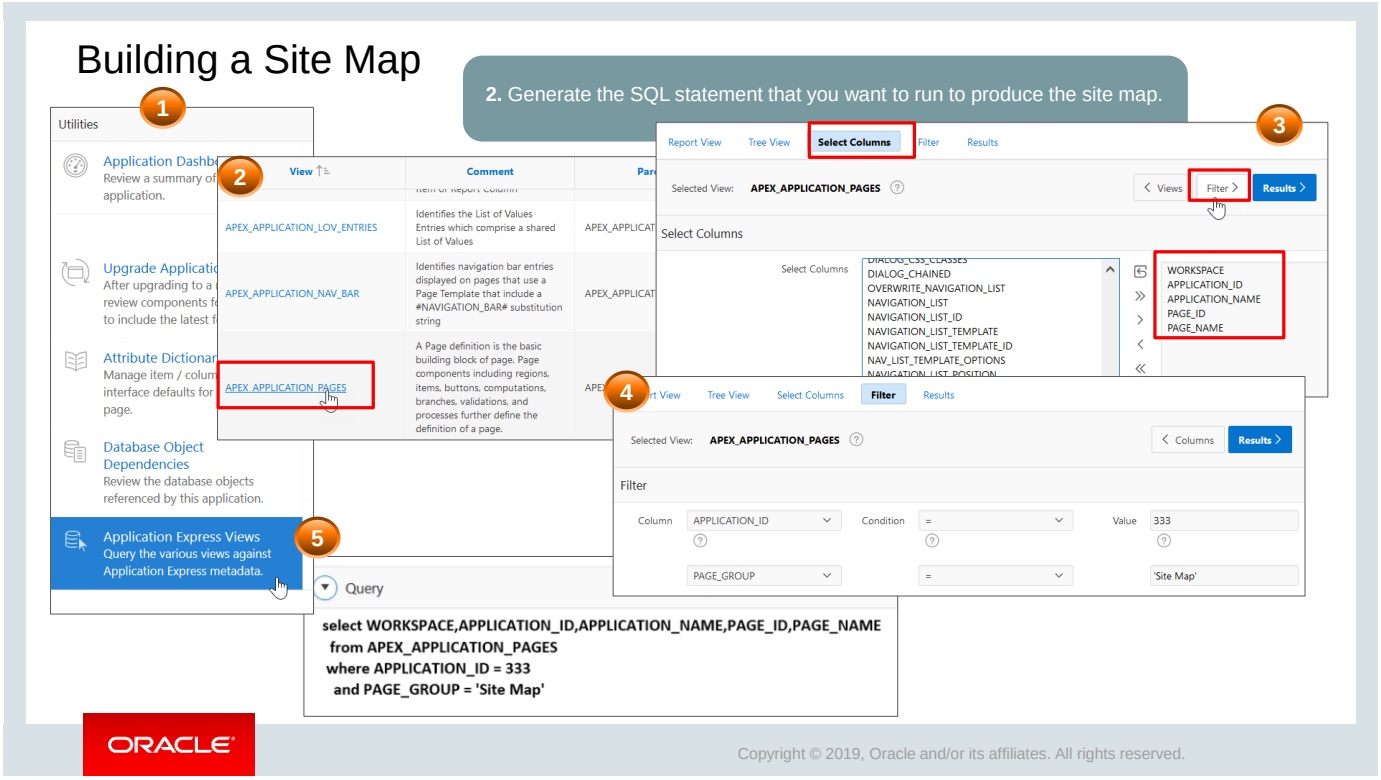
1. Create a page group with the pages that you want to appear in the site map.
2. Generate the SQL statement that you want to run to produce the site map. To do so, under **Utilities > Application Express Views** (here Steve selects from the `APEX_APPLICATION_PAGES` view), create the SQL to select the appropriate `PAGE_NAME` and `PAGE_ID` for your `PAGE_GROUP`.
3. Create a report that shows the page name.
4. Create a link from the page name to the page it corresponds to. Pass the item value `#PAGE_ID#` in the Page field.

In the next few slides, you will learn in detail how to perform each of these steps.



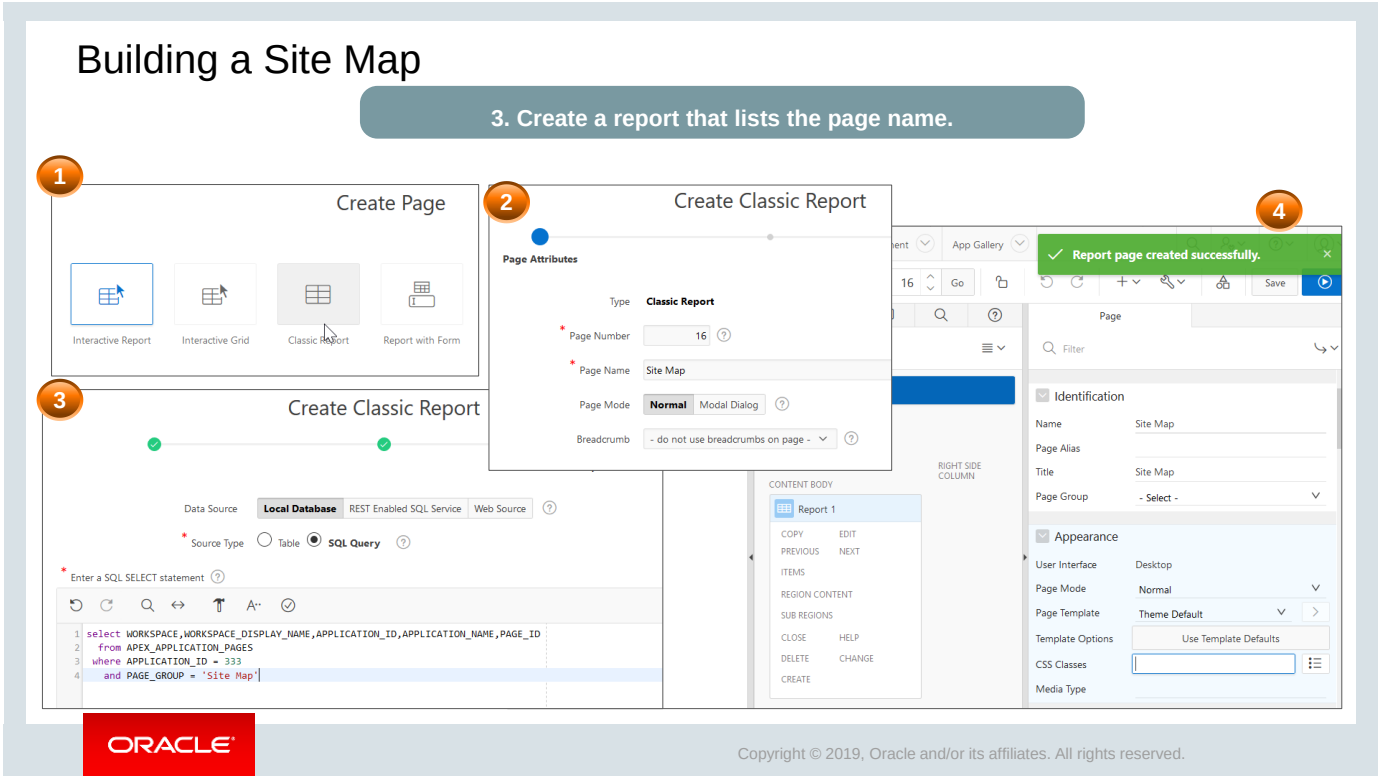
So, as per Stella's request, Steve starts with the first step in creating a site map. And that is creating a page group and assigning the pages that you want to appear in your site map to the page group you created. To create a page group, perform the following steps:

1. On the application home page, click **Utilities**.
2. On the right pane, under Page Specific Utilities, click **Page Groups**.
3. On the Page Groups page, click **Create**.
4. Enter `Site Map` for **Name** and click **Create**.
5. Click the **Page Assignments** tab.
6. Select *Site Map* for New Group. Select all the pages you want to assign for the `Site Map` Page Group and click **Assign Checked**.
7. Click Site Map to see the pages belonging to this Page Group.



The next step in building your site map is to generate the SQL statement that you want to run to produce the site map. To produce the SQL statement, go to your application and click **Utilities**. You then perform the following steps:

1. On the Utilities page, select the **Application Express Views** option (screenshot 1).
2. Select the **APEX_APPLICATION_PAGES** view (screenshot 2).
3. Click the **Select Columns** tab (screenshot 3).
4. On the **Select Columns** tab, if `PAGE_NAME` is not included in the list on the right, then select `PAGE_NAME` column and click the right arrow (>) to move it to the columns selected list. Similarly, you do not want to display the `WORKSPACE_DISPLAY_NAME`. Select `WORKSPACE_DISPLAY_NAME` and click the left arrow (<) to move it to the column list on the left. Click **Filter >** (screenshot 3).
5. Select `APPLICATION_ID` for **Column** and enter `333` for **Value** (screenshot 4).
6. In the next line, enter `PAGE_GROUP` for **Column** and enter `Site Map` for **Value** (screenshot 4).
7. Click **Results >** (screenshot 4).
8. Notice that the pages you selected earlier are on the list.
9. Click and expand Query to review the query that was executed (screenshot 5). Select the query and copy it to your clipboard. In the next slide, you will be using this SQL query to create a report.

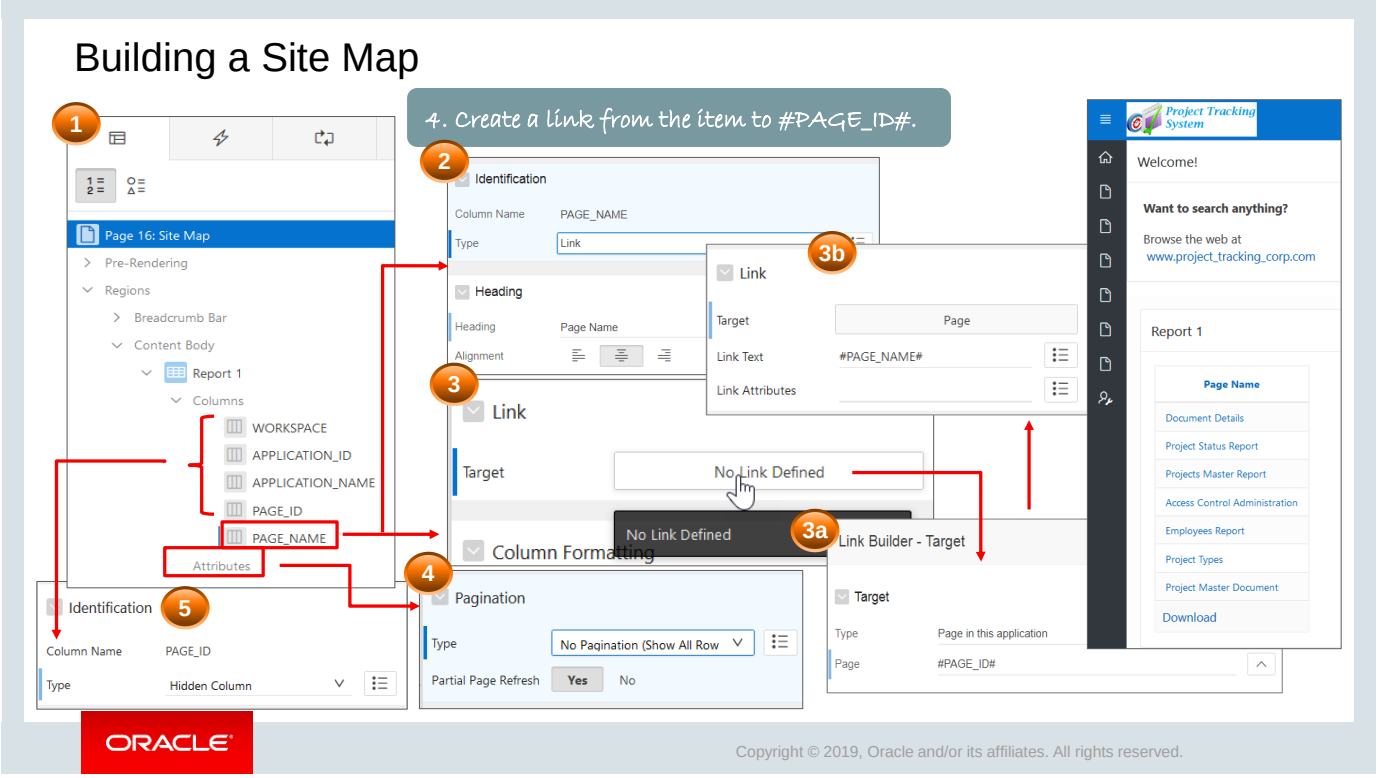


The next step is to create a report that invokes the query you just generated. On the Application page, click **Create Page** and perform the steps below:

1. Click **Report.**
2. Click **Classic Report**.
3. Enter `Site Map` for the **Name** and click **Next**.
4. Keep the default and click **Next**. This page is going to be added to the navigation bar later.
5. Enter the following SQL statement (that you copied in the previous slide) and click **Create.**

```
select WORKSPACE, APPLICATION_ID, APPLICATION_NAME, PAGE_ID, PAGE_NAME
from APEX_APPLICATION_PAGES
where APPLICATION_ID = 333
and PAGE_GROUP = 'Site Map'
```

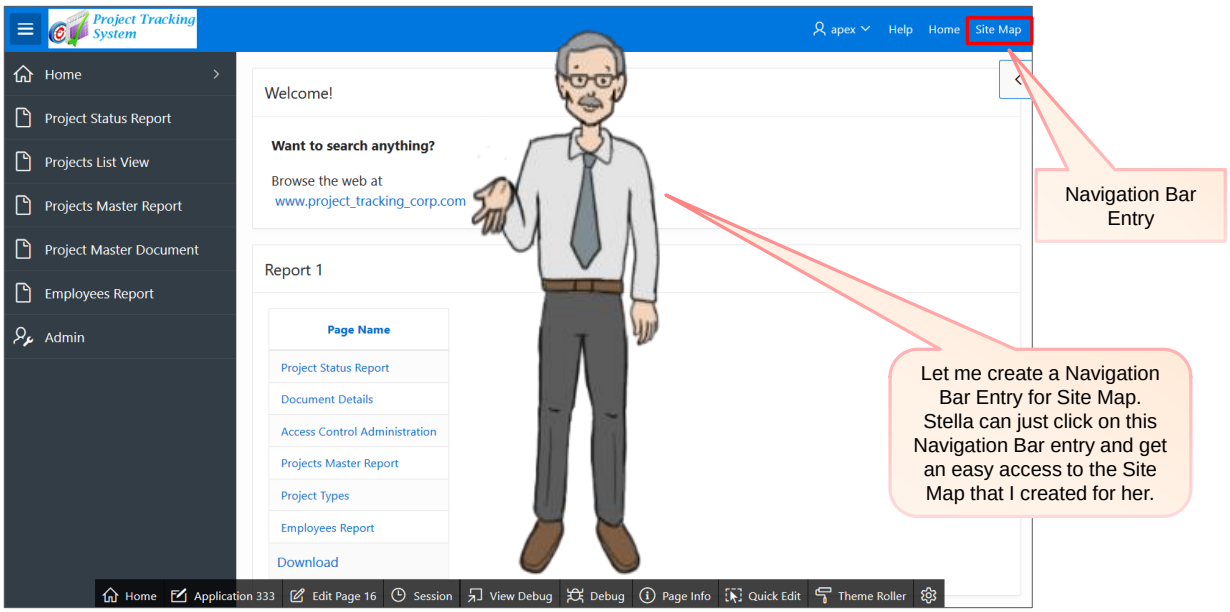
The report page is created. In the next slide, Steve links the `PAGE_NAME` column to the `#PAGE_ID#` so that it opens the corresponding page when clicked.



Steve is now toward the end of creating the Site Map for Stella. He just needs to add the link to view the page when selected. Let's see how.

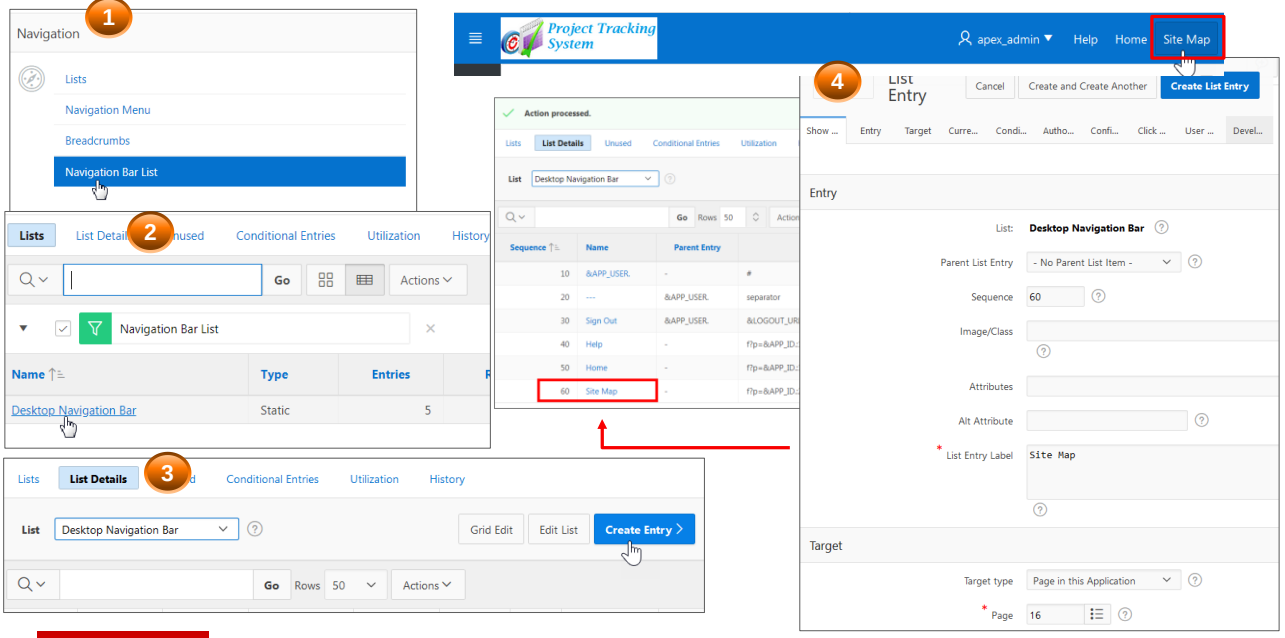
1. Under Rendering, select `PAGE_NAME` under columns.
2. Update the properties in Property Editor as follows:
   - **Type**: Select **Link**
   - **Target**: Select `#PAGE_ID#`
   - **Link Text**: Enter `#PAGE_NAME#`
3. Select `PAGE_ID` under Columns (under Rendering) and change its **Type** to *Hidden*. Repeat the same for the other columns (that is, `WORKSPACE`, `APPLICATION_ID`, and `APPLICATION_NAME`) too. This is because you just want to display the *Page Name* column in your Site Map.
4. Select **Attributes** under Site Map region and change **Pagination Type** as *No Pagination (Show all rows).*
5. **Save and Run** the page, and you can see the Site Map displaying the Page Name with links to all the pages that Steve included in the Site Map page group.

Adding Site Map as a Navigation Bar Entry



Now that Steve has all the important reports listed by Page Name on his Site Map page, he is looking for a way that would give an easy access to Stella and the other managers. And this is when he thought of creating a Navigation Bar entry. Let's see how Steve does it in the next slide.

Example: Adding Site Map as a Navigation Bar Entry



Steve now creates a new list entry called Site Map on the Navigation Bar. Stella and other project managers can just click on this navigation bar entry, and she/he would be directed to all the important pages in the application. Let's see how Steve does it.

You go to your application's home page and click Shared Components and then follow the steps below:

1. From Shared Components, select **Navigation Bar** List.
2. Select **Desktop Navigation Bar**.
3. Click **Create Entry**, and the List Entry page opens.
4. Enter `Site Map` for Entry Label. Select *Page in this Application* for **Target Type** and select the site map page from the Page drop-down list. Here Steve selects *Page 16* (*Site Map*).
5. Click **Create List Entry**.
6. Run the application, and you can see the new navigation bar entry.

Quiz

Navigation bars are different from other shared components in that you do not need to reference them on a page-by-page basis.

a. True

b. False

```
select WORKSPACE,APPLICATION_ID,APPLICATION_NAME,PAGE_ID,PAGE_NAME
 from APEX_APPLICATION_PAGES
where APPLICATION_ID = 333
 and PAGE_GROUP = 'Site Map'
```

Answer: a

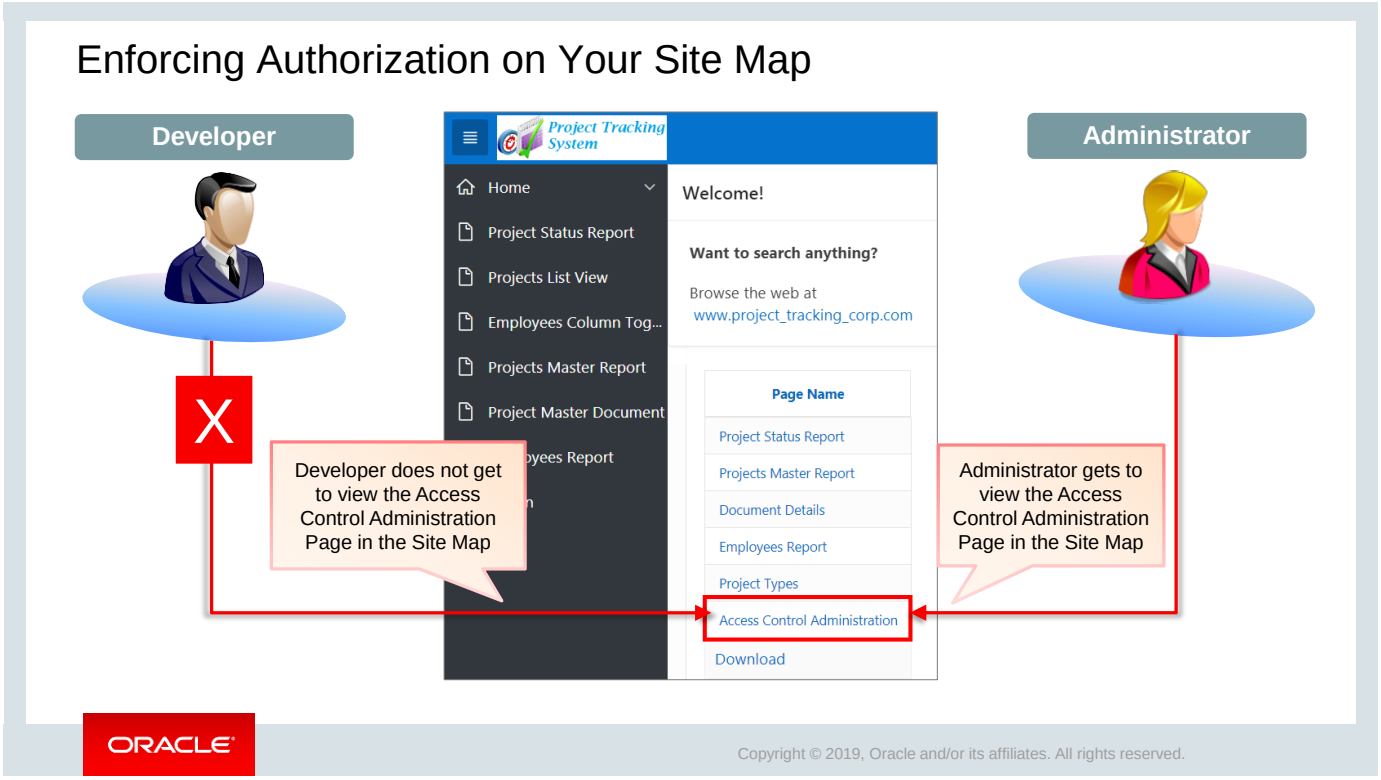
Practice17-3 Overview: Building a Site Map

This practice covers the following topics:

- Building a site map page
- Adding the page as a navigation bar entry

Lesson Agenda

- Building a Hierarchical List with Images
- Building a Database-Driven Navigation Report
- Building a Site Map
- Enforcing Authorization on Your Site Map

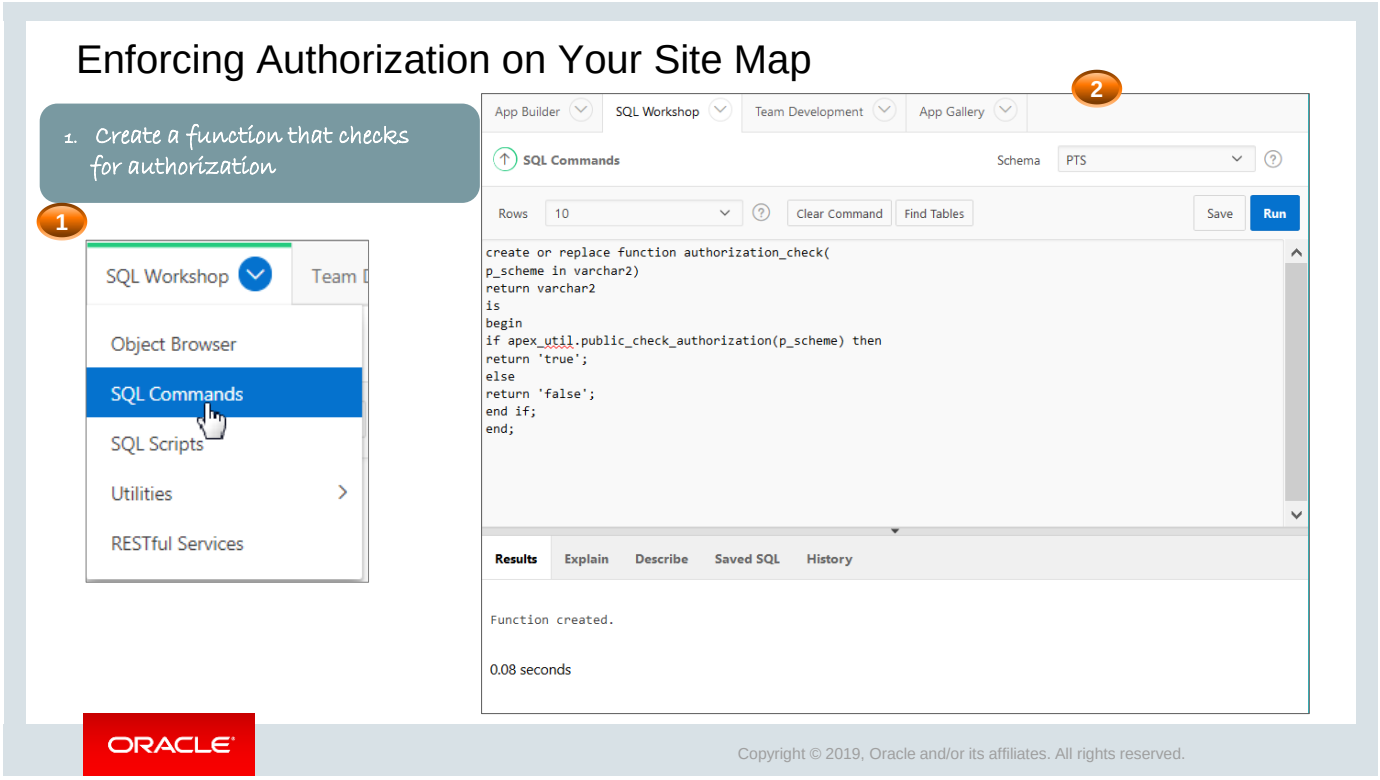


Steve has now created a Site Map so that the project managers can easily navigate between pages by the page name. However, he does not want all the users to have access to each of these pages included in the Site Map. For example, he does not want the team members or developers to see the *Access Control Administration* page. And for that he wants to enforce authorization on his Site Map.

To enforce authorization on the Site Map:

1. Create a function that checks for authorization
2. Update the SQL query on the report to check whether the function is true

In the next couple of slides, you will see how Steve performs these steps.

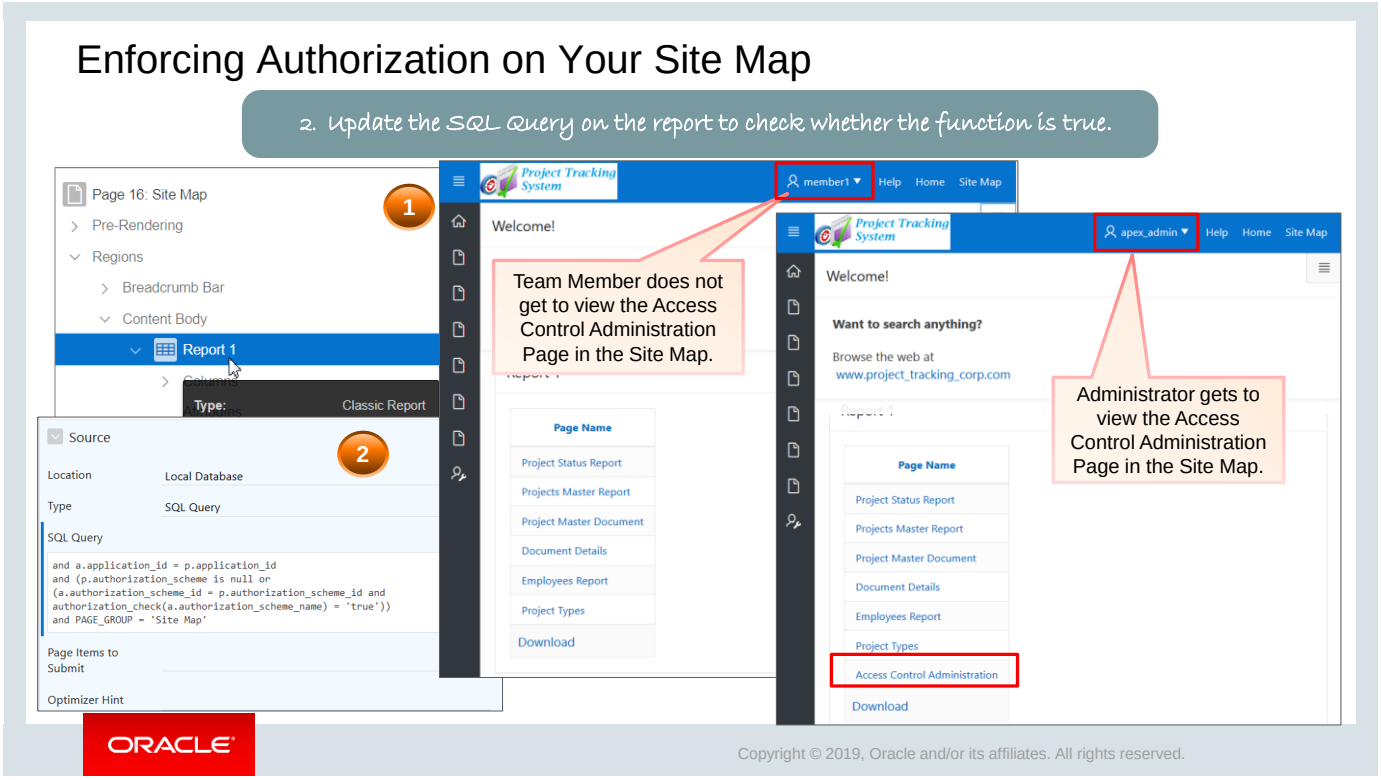


To show only those pages that a particular user is authorized to use on the site map, Steve must create a function that checks the authorization scheme and then selects only those pages.

1. Click the **SQL Workshop** pull-down menu and select **SQL Commands**.
2. On the SQL Commands page, enter the following scripts and click **Run**.

```
create or replace function authorization_check(
p_scheme in varchar2)
return varchar2
is
begin
if apex_util.public_check_authorization(p_scheme) then
return 'true';
else
return 'false';
end if;
end;
```

3. The function is created to check authorization.



Steve is now working on his last requirement; that is, he does not want all the users to view the complete page list under Site Map. For example, he wants only the users with Administrator rights to see the *Access Control Administration* Page under Site Map list. Let's see how he does that.

1. Open the Site Map page in Page Designer view.
2. Under **Regions**, select **Report 1** (screenshot 1).
3. Enter the following code in the **Source > SQL Query** text box (screenshot 2) and click **Save and Run Page**.

```
select distinct PAGE_ID, PAGE_NAME
from APEX_APPLICATION_PAGES p, apex_application_authorization a
where p.APPLICATION_ID = :APP_ID
and a.application_id = p.application_id
and (p.authorization_scheme is null or
(a.authorization_scheme_id = p.authorization_scheme_id and
authorization_check(a.authorization_scheme_name) = 'true'))
and PAGE_GROUP = 'Site Map'
```

4. Sign in as an Administrator to see the *Access Control Administration* page in the Site Map list.
5. Sign in as a team member, and you do not get to see the *Access Control Administration* page in the Site Map list.

Practice 17-4 Overview: Enforcing Authorization on the Site Map

This practice covers the following topics:

- Adding a function that determines authorization of a page in the site map
- Changing the SQL report query for the site map to make sure that the page is displayed only if authorized

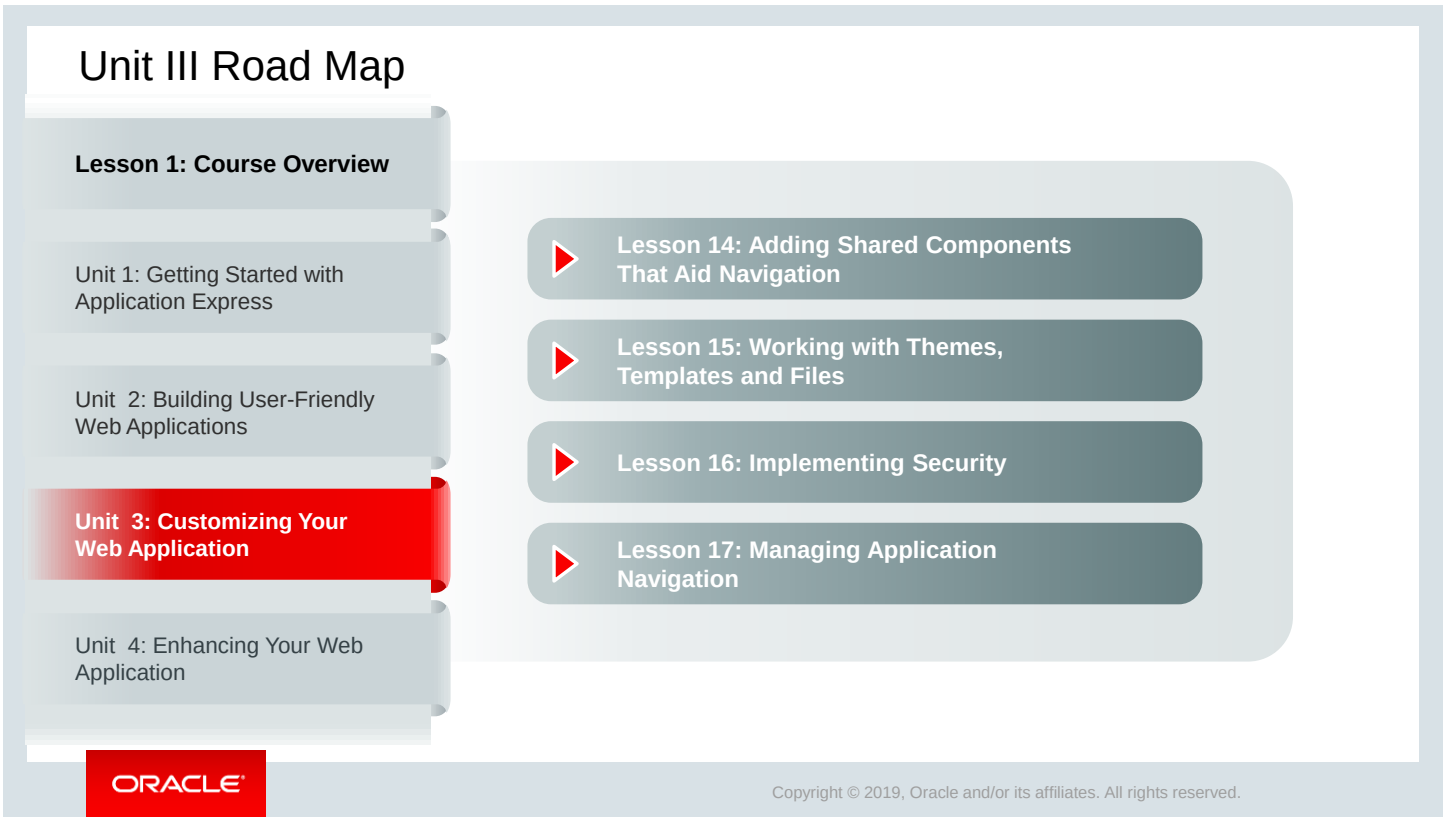
Summary

In this lesson, you should have learned how to:

- Build a hierarchical list with images
- Build a database-driven navigation report
- Build a site map
- Enforce authorization on your site map

In this lesson, you have learned how to build a hierarchical list with images, a database-driven navigation report, and a site map. You should have also learned to authorize access to a site map.

Unit III Summary: Customizing Your Web Application

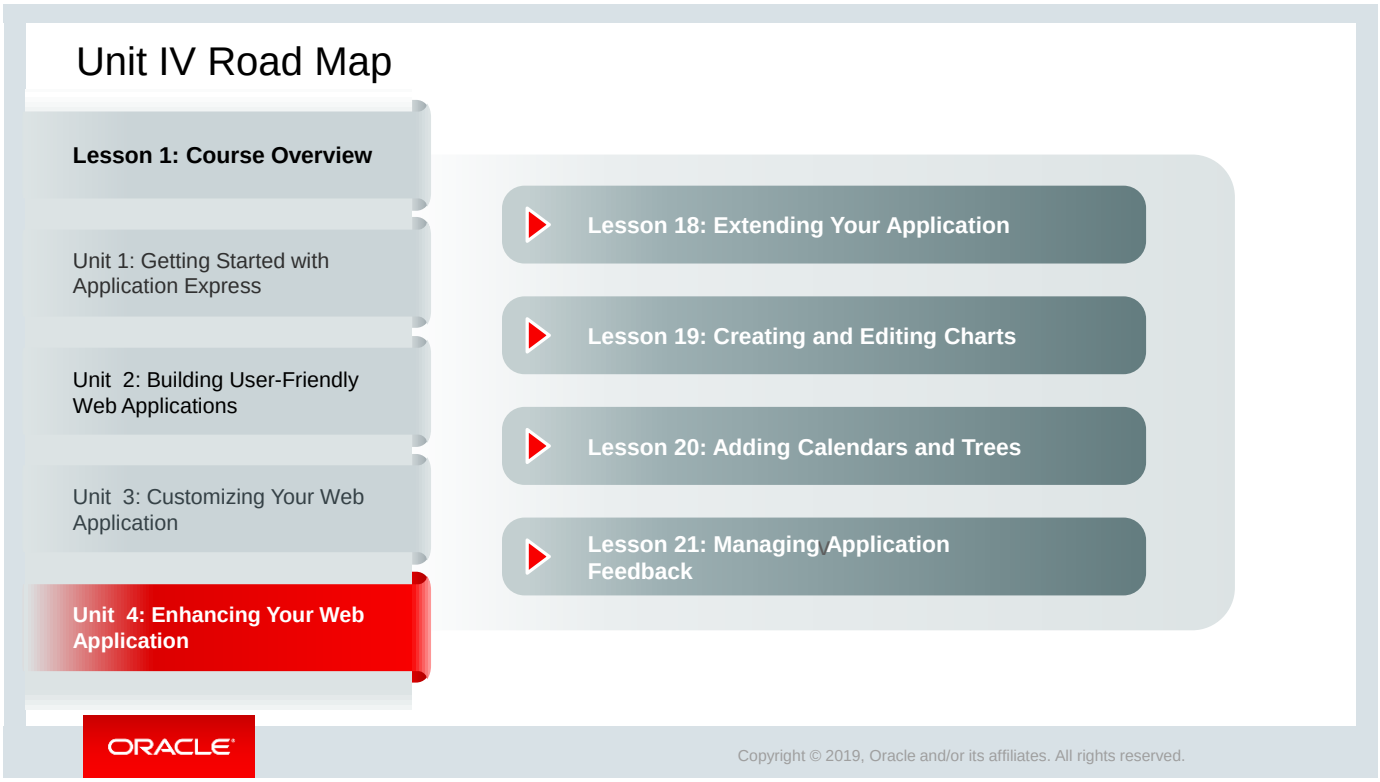


In Unit 3, you completed four topics.

Unit IV Introduction: Enhancing Your Web Application

Steve Enhances the Project Tracking System Application

Steve is almost ready with the PTS application. As a final task, he is planning to enhance the application by extending it and adding charts, calendars, trees, dynamic actions, and plug-ins. He believes that after he adds these features, the PTS application will be production ready and can be used by the company.

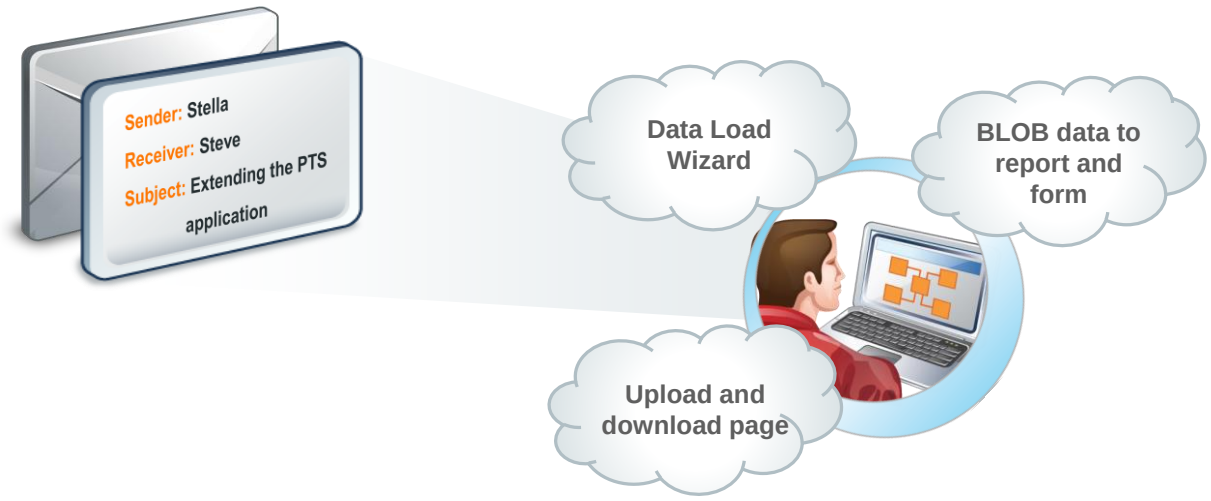


In Unit 4, you add advanced features to your application by creating dynamic actions, plug-ins, calendars, trees, charts, and application feedback.

18

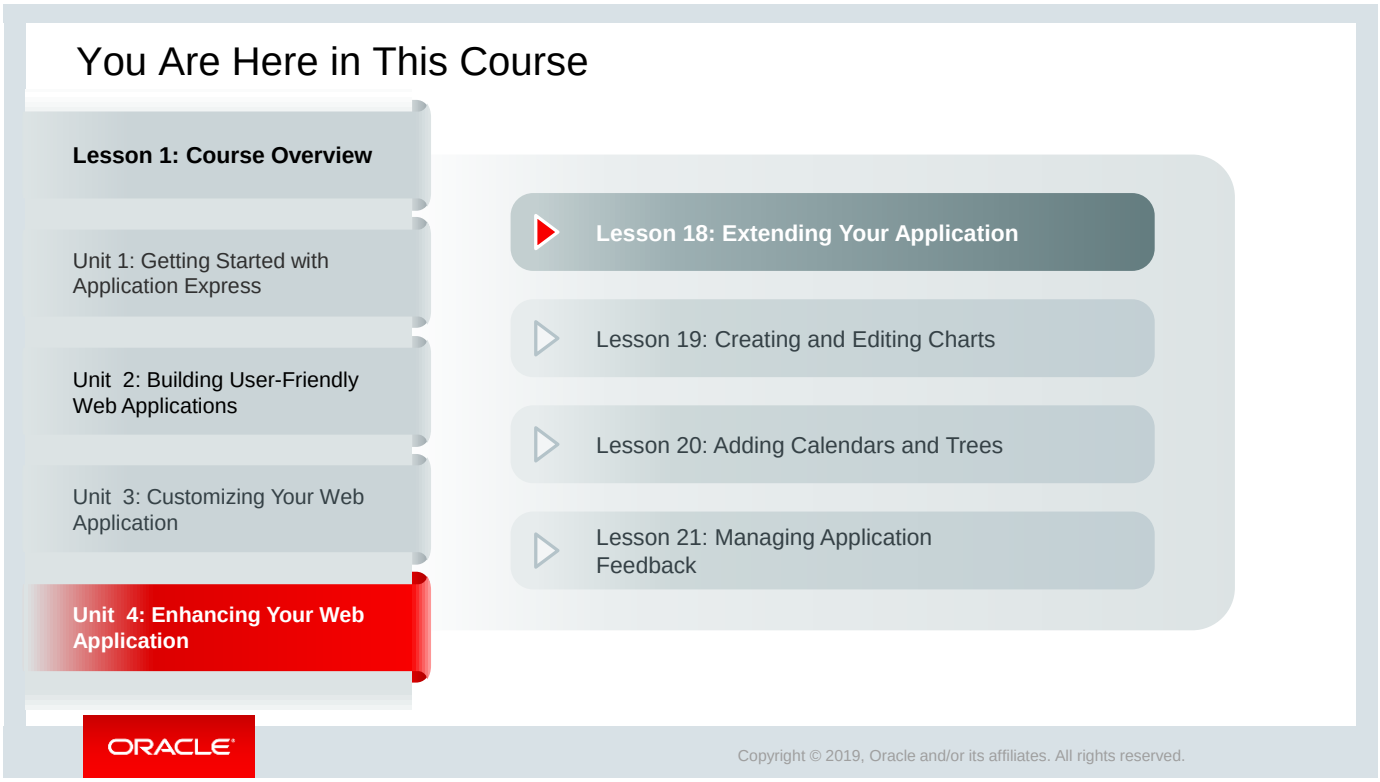
Extending Your Application

Steve Extends the PTS Application



Steve now has a functional PTS application that he developed using Oracle Application Express. Although Stella, his manager, is happy with the application, she has suggested some additional features to enhance the application further. She enquired if it would be possible to incorporate features that would allow users to load data directly into the application, visualize data, and create trees and calendars and feedback mechanism to allow direct interaction with the user.

Steve, being an expert Oracle Application Express developer, knows that it is indeed possible to implement all the suggested features into the PTS application. He now considers enhancing the application to implement these features.



This slide shows a graphical representation of the entire course, highlighting lesson 18 in particular, which is dealt with in these slides.

Objectives

After completing this lesson, you should be able to:

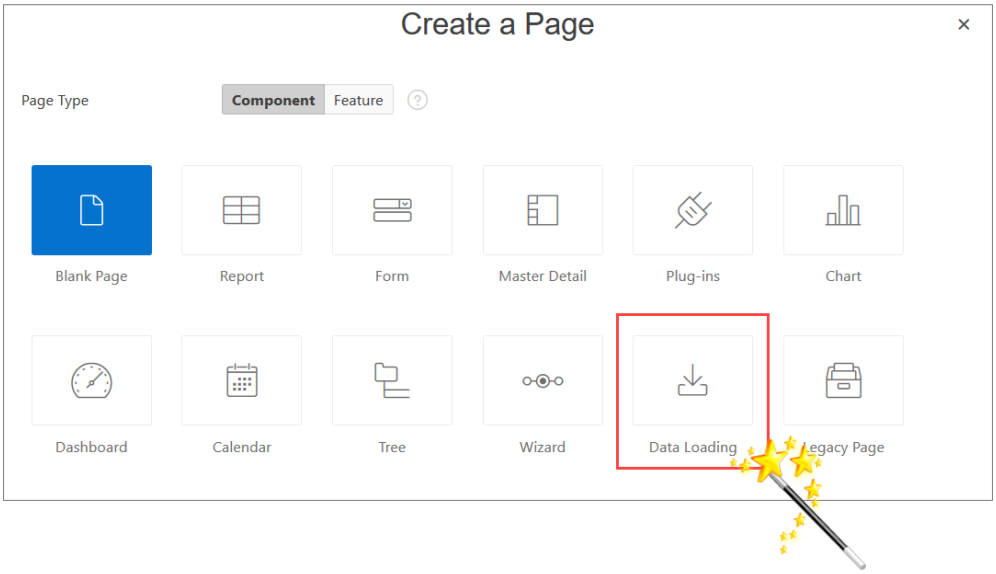
- Create a Data Load Wizard to enable users to load data to their app
- Create an upload and download page that enables users to upload and download files
- Create a process to send an email notification

In this lesson, you learn how to create Data Load Wizard pages and an upload and download page. You also learn how to how to create a process to send email notifications from your application and how to add BLOB data in your application.

Lesson Agenda

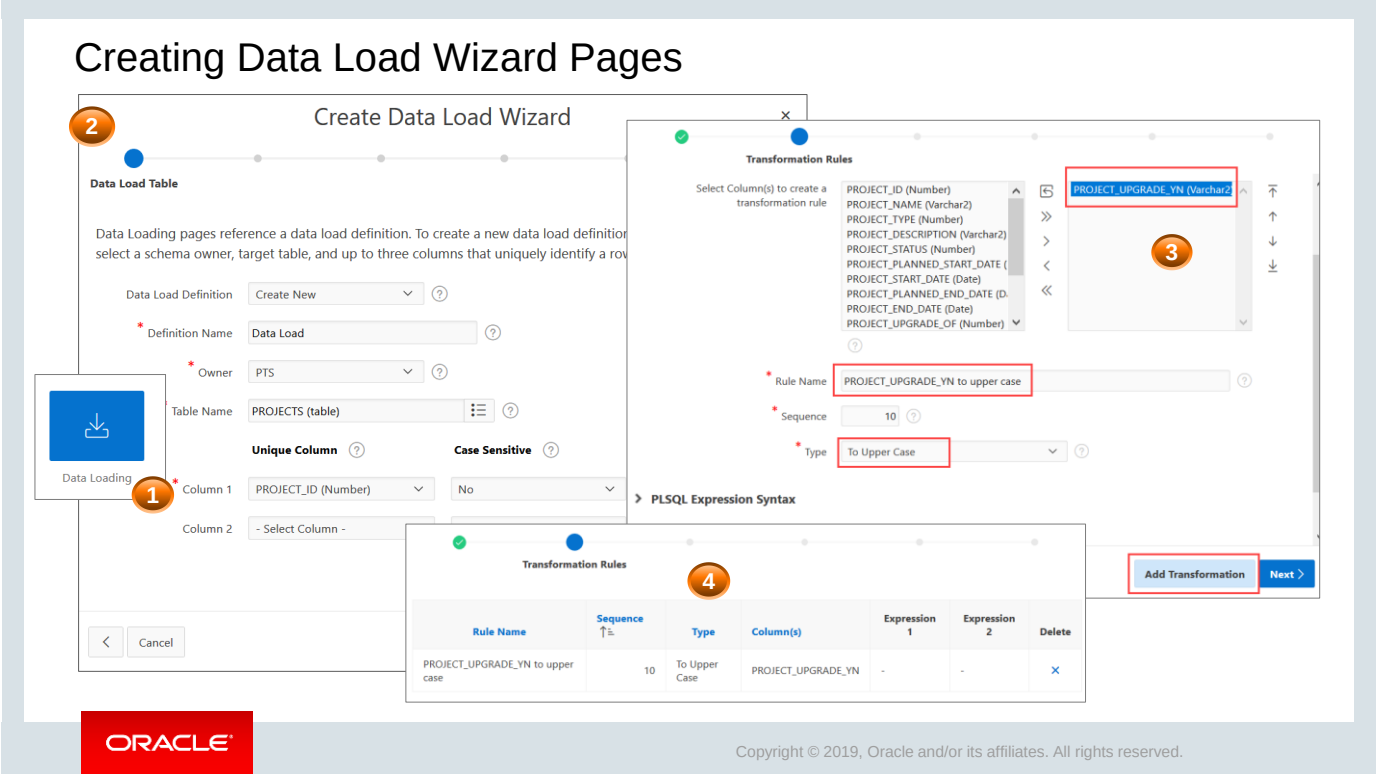
- Creating Data Load Wizard Pages
- Sending an Email from an Application
- Creating an Upload and Download Page

Data Load Wizard



One of the requirements of PTS application is the ability to load data directly into the application. For instance, the preference of a project manager is loading project-related information into the PROJECTS table in a batch instead of using a form. In such scenarios, the data loading feature in an application proves very useful. Steve considers implementing the data loading component into the PTS application using which the end users can dynamically import data into a table.

To import and load data into an application, the end user must run the **Data Load Wizard** that uploads data from a file or copy-pastes data directly into the wizard. In Oracle Application Express 19.1, you can load data from various file types such as `.CSV`, `.XSLX`, `.XML`, and `.JSON` files. You can create a series of data loading pages to build a **Data Load Wizard** in your application. You can use this wizard to add table lookups and transformation rules that are executed when the Data Load Wizard runs.

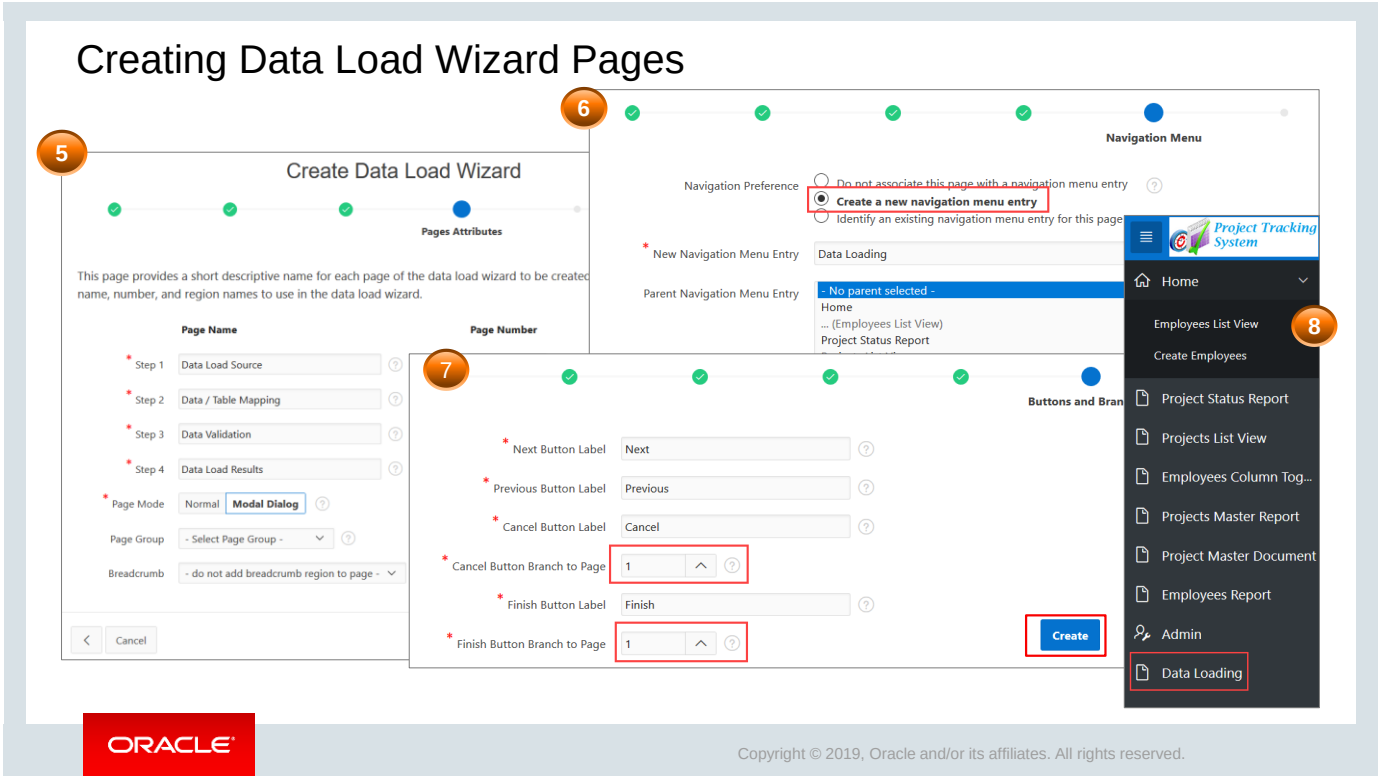


Since managers want the ability to upload data to the `PROJECTS` table, Steve is creating a Data Load Wizard for this specific requirement. The wizard will allow managers to directly upload data into the `PROJECTS` table, either by copying and pasting the data or by uploading a file. One of the key steps for this is to create a data load definition.

Steve creates a new data load definition on the `PROJECTS` table. The wizard allows the user to directly upload data, either by copy-pasting the data or by uploading a file containing the data.

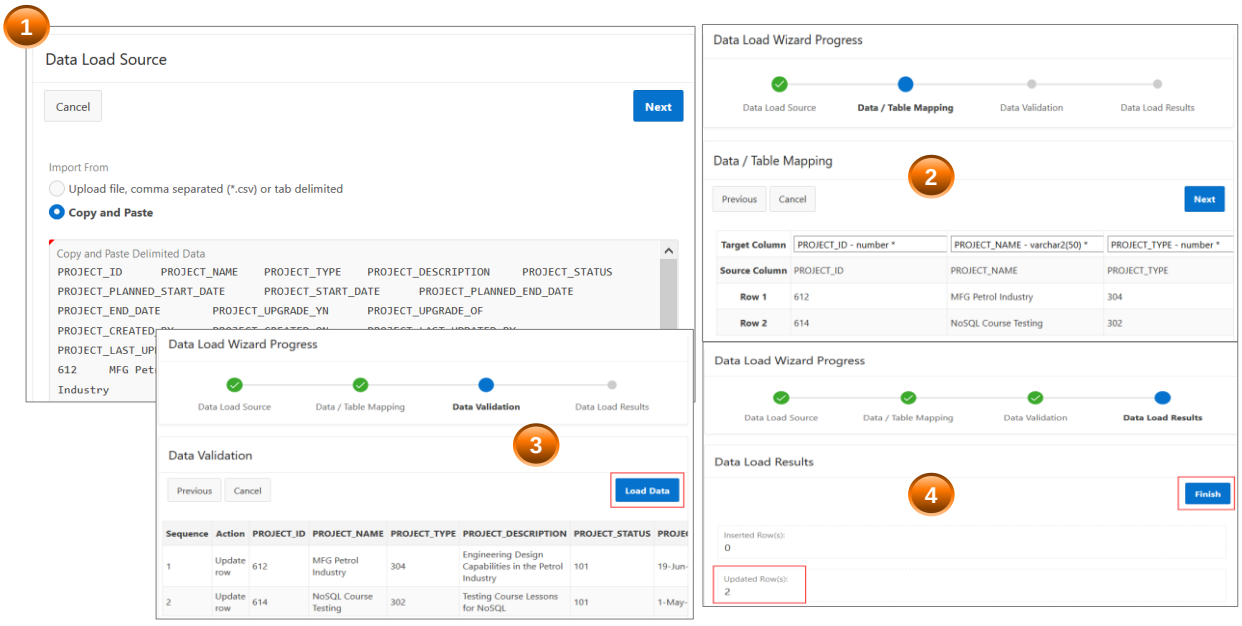
To create Data Load Wizard pages, perform the following steps:

1. Navigate to the application home page, click **Create Page,** and select **Data Loading.**
2. You must create a new data load definition. To do so, select Create New in the **Data Load Definition** field. Then specify the definition name; select a schema owner, table name, and up to three columns that uniquely identify a row, as shown in screenshot 2; and click **Next.**
3. You must now specify a transformation rule. Transformation rules allow you to change the data being uploaded before it is inserted into the base table. If required, you select the column to transform and then the desired rule to apply to it. In this use case, Steve wanted the value of the `PROJECT_UPGRADE_YN` column to be in upper case always. Here, you must also provide the rule name (`PROJECT_UPGRADE_YN` to upper case) and select **To Upper Case** for the rule type. Click **Add Transformation** as shown in screenshot 3. The transformation rule is now created as shown in screenshot 4.
4. If required, you can add a new table lookup by specifying the column name and the lookup definition. Table lookups allow you to match an uploaded value against another table and use the associated key value instead of the uploaded value. In this example, the look up table is not created. Click **Next.**



5. In the Page Attribute section of the wizard, a short descriptive name for each page of the Data Load Wizard to be created is provided, along with their page number. Here, in the **Page Mode** field, select **Modal Dialog** and click **Next.** A Modal dialog is an overlay window that remains active until the end user closes it.
6. Select the navigation preference – **Create a new navigation menu entry**, and click **Next.** Selecting this option would place the Data Load wizard as a separate menu item in the left navigation pane in the application, as shown in screenshot 8.
7. In the last section of the wizard, which is Buttons and Branching, set the navigation for Cancel and Finish button in the wizard:
   - **Cancel Button Branch to Page:** Select **1**
   - **Finish Button Branch to Page:** Select **1**
     Note: Page 1 is the home page. When a user clicks **Cancel** or **Finish** while working with the Data Load wizard, the user will be navigated to page 1, which is the home page of the PTS application.
   - Confirm the wizard attributes and click **Create.**
8. Click **Save** and **Run.** As you can see in screenshot 8, the Data Loading wizard is now created and is listed in the left navigation menu. End users can now use the wizard to upload data directly into the PTS application.

Data Load Wizard Pages



After the Data Loading wizard pages are created, notice the flow of the wizard. Four wizard pages are created. When the end user clicks Data Loading in the PTS application, these pages open in sequence where he can upload the data:

- The first wizard page – *Data Load Source*, is where you specify the data load source. You want to upload a file with data. Select **Upload file, comma separated (*.csv) or tab delimited** for **Import From** and click **Browse.** Select the file. Enter the separator value and click **Next.**
- In the second wizard page - *Data/Table Mapping,* select the columns to match the columns in the database and click **Next.**
- The third page is the *Data Validation* that displays the data that will be inserted into the database. Here the lookup is applied. Verify the data and click **Load Data.**
- The fourth and the last page of the wizard - *Data Load Results* page, shows the rows that were inserted and updated, that failed, and that need to be reviewed. Click **Finish** to complete the data load process.

Practice 18-1: Creating Data Load Wizard Pages

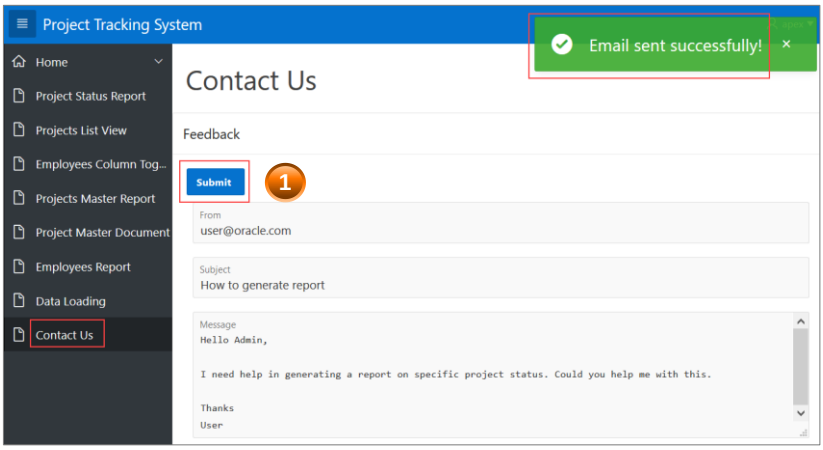
This practice covers the steps to create a data load wizard to enable end users to upload data to their app.

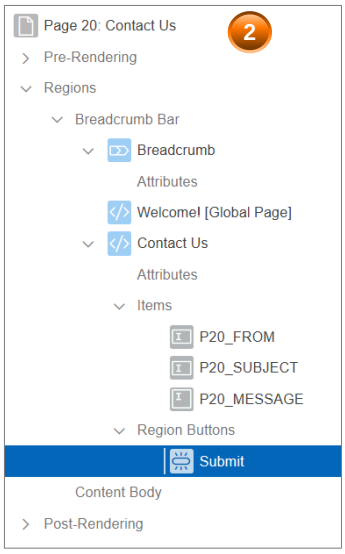
Lesson Agenda

- Creating Data Load Wizard Pages
- Sending an Email from an Application
- Creating an Upload and Download Page

Contact Us Page

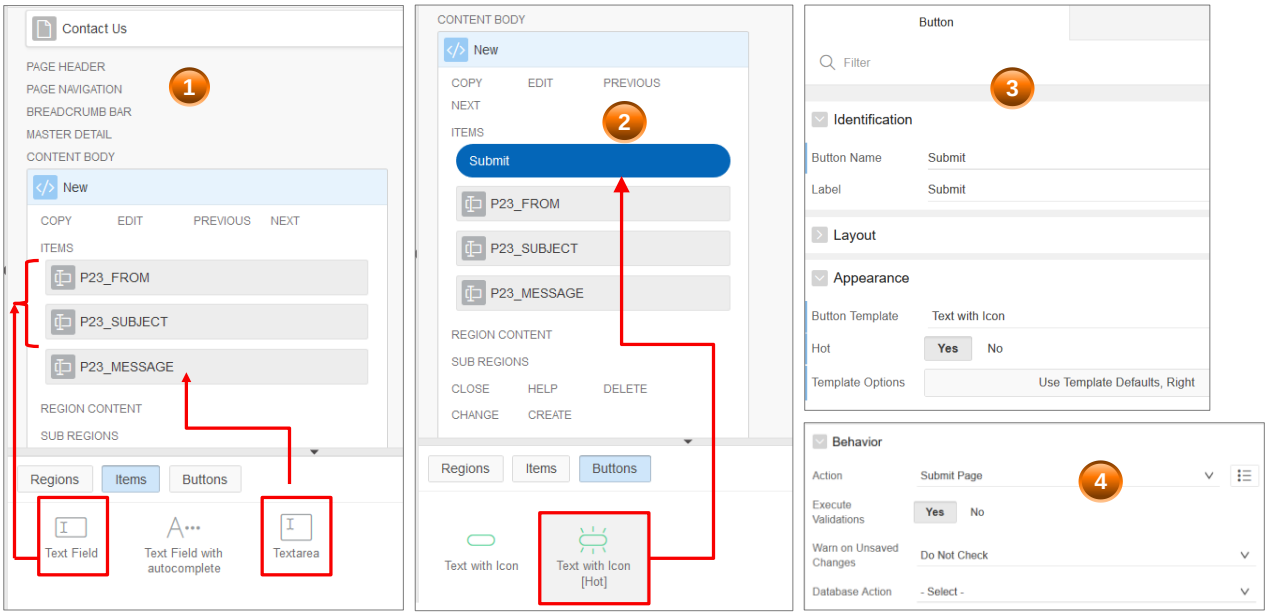
Send an email with values from a form



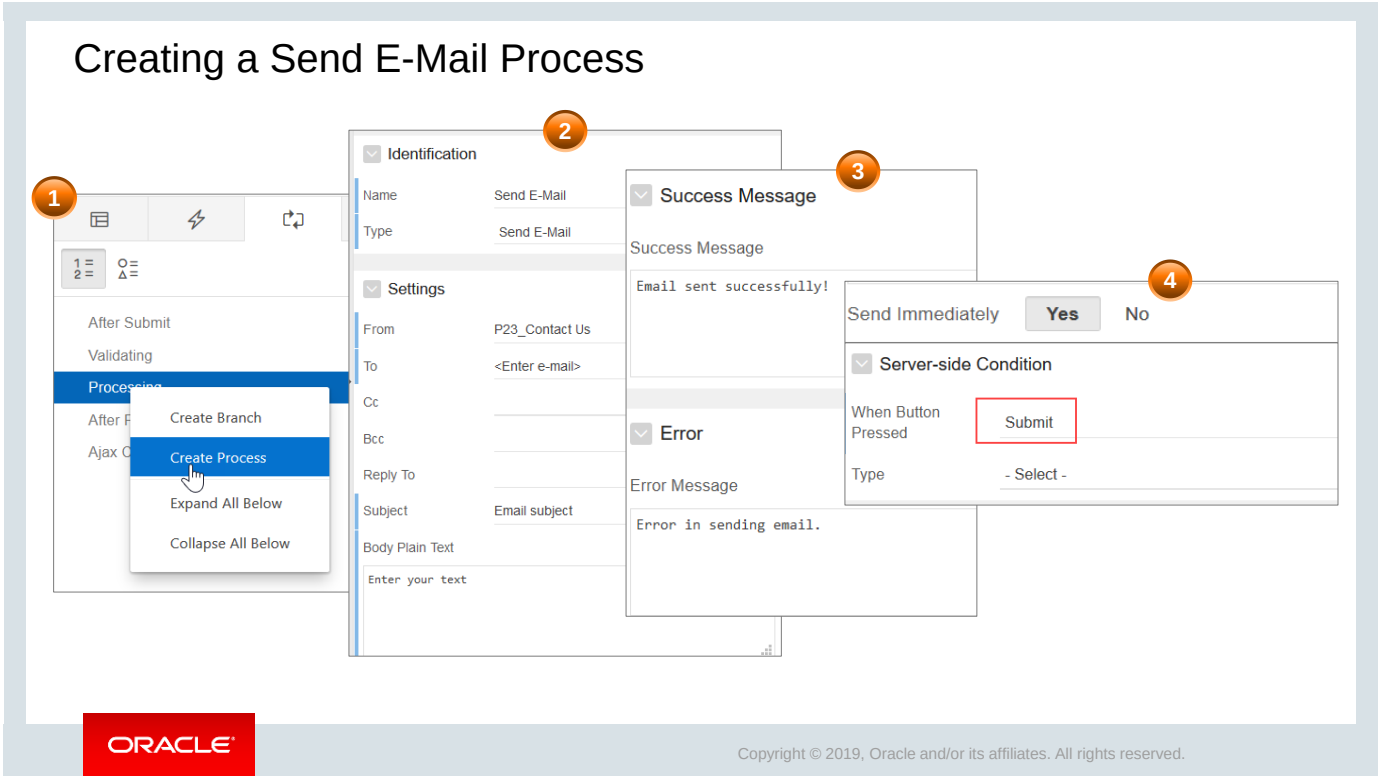


To gather feedback from end users, Steve created a *Contact Us* page, as shown in screenshot 1. It is a page where users enter their email address, the subject, their feedback, or any queries in the respective fields and then submit the page. When the page is submitted, a process is fired that will send an email to the desired recipient. There are two methods to create the send email process: declaratively or by using the `APEX_MAIL` package API. In the example in the following slide, you learn about the declarative approach.

Contact Us Page



For the Contact Us page, you first create a blank page where you create a region named Contact Us. Then you add two text fields and a text area for the fields **From, Subject,** and **Message,** respectively. This is depicted in screenshot numbers 1 and 2. Next, you add a button and define its behavior as shown in screenshots 3 and 4.



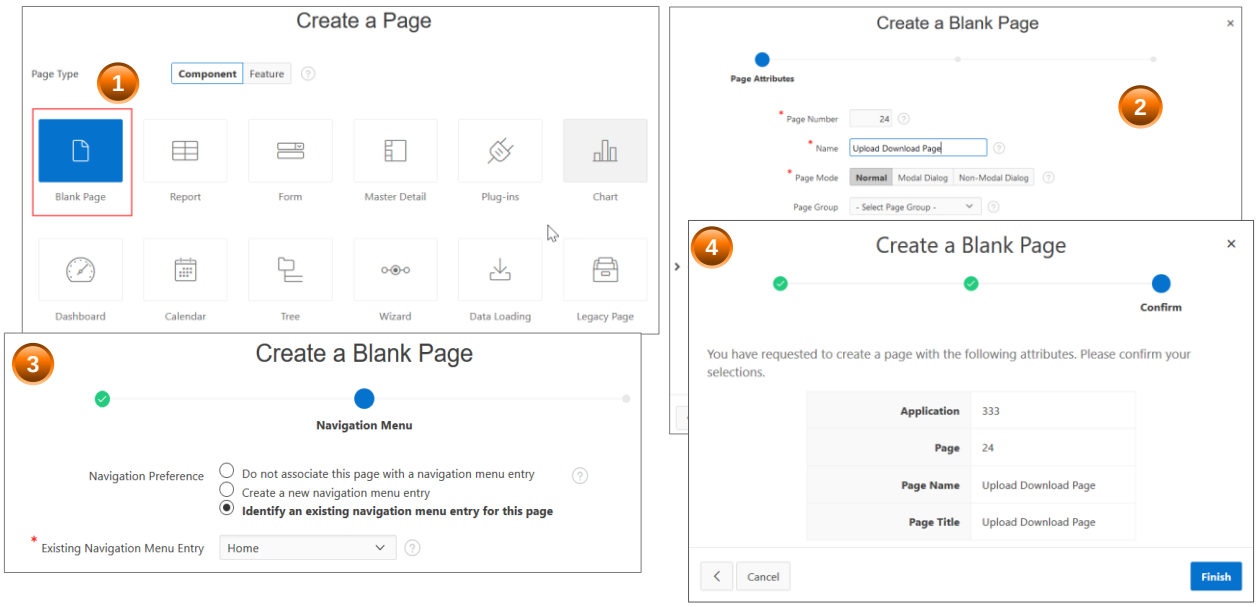
After creating the Contact Us page, you must create a process that will facilitate end users of the app to send messages such as queries or feedback. To create the Send E-Mail process, perform the following steps:

1. In the left pane, under Processing, right-click **Processing** and select **Create Process.**
2. The new process is created. In the Property Editor, enter a name for the process and select **Send E-Mail** for Type.
3. Enter a value or a page item name for all the mandatory fields and indicate whether you want the email to be sent immediately or not.
4. Enter messages for Success and Error.
5. Select **Submit** for **When Button Pressed.** Save and run the page. After you enter the email ID, message and click the **SUBMIT** button; the process is executed and the email is sent.

Lesson Agenda

- Creating Data Load Wizard Pages
- Sending an Email from an Application
- Creating an Upload and Download Page

Creating an Upload and Download Page

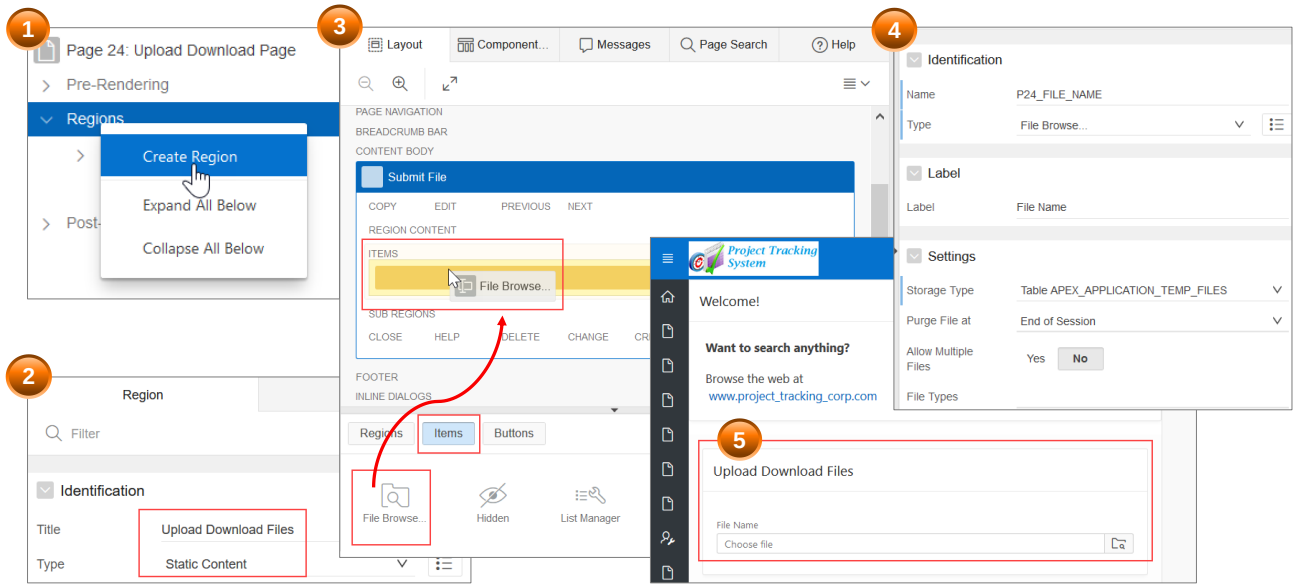


Another useful requirement that Steve envisages is the feature to upload and download a file, which can be any project-related document. To implement this feature, Steve creates an **Upload and Download** page, where he adds the **File Browse** item.

To create a blank page, perform the following steps:

1. Navigate to the application home page, click **Create Page,** select **Blank Page,** and click **Next.**
2. To define page attributes, in the **Name** field, enter **Upload Download Page.**
3. For Navigation preference, click **Identify an Existing navigation menu entry for this page.** In the Existing Navigation Menu Entry, select **Home.** This creates the *Upload Download Page* under the PTS application home page. Click **Next.**
4. In the Confirm page dialog, click **Finish.** This creates the *Upload Download Page* under the PTS application home page.

Adding a File Browse Item on the Upload Download Page



On the Upload Download page, Steve now adds a File Browse item that allows the end user to browse for a file in his system and upload it into the application workspace memory. You can add the **File Browse** item on any page in your application where you want to implement this functionality. When you use the File Browse item type, the files that you upload are stored in a table called `APEX_APPLICATION_TEMP_FILES`. Every workspace has access to this table through a view called `APEX_APPLICATION_TEMP_FILES`.

To create the File Browse item type on a blank page, perform the following steps:

1. In the Rendering tab, right-click **Content Body** under Regions and click **Create Region** (screenshot 1). In the newly created region on this page, you add the File Browse item.
2. In the Property Editor, name this region as **Upload Download Files.** The **Type** is set to Static Content, as shown in screenshot 2.
3. In the central pane in Page Designer, drag the **File Browser** item from the Items gallery to under Items in the Grid Layout, as shown in screenshot 3. Alternatively, you create the item by right-clicking the Rendering pane and selecting **Create Page Item.**
4. In the Property Editor, define the following attributes as shown in screenshot 4:
   - **Name:** Enter **P24_FILE_NAME**
   - **Type:** Select **File Browse**
   - **Label:** Enter **File**
   - **Storage Type:** Select `APEX_APPLICATION_TEMP_FILES`
5. Save and run the page. The item is successfully created on the Upload Download page as shown in screenshot 5.

Practice 18-2: Adding an Upload and Download Page

This practice covers the following topics:

- Creating a form in an HTML region with a file upload item and a button
- Creating a report on the document table that has links to download documents
- Providing links to download the documents in the report

Summary

In this lesson, you should have learned how to:

- Create a Data Load Wizard to enable users to load data to their app
- Create an upload and download page that enables users to upload and download files
- Create a process to send an email notification

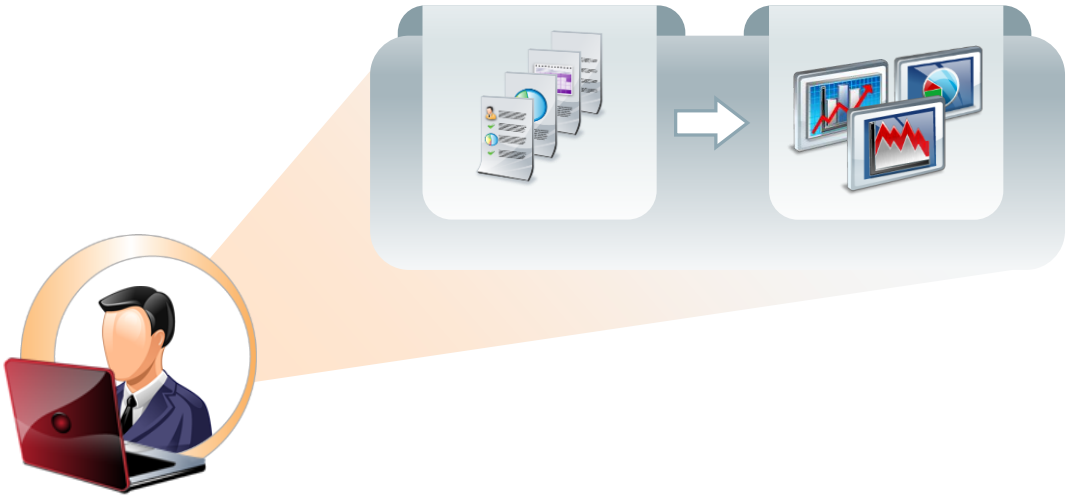
ORACLE

In this lesson, you should have learned how to extend your application to use advanced techniques such as creating Data Load Wizard pages, send email notifications with feedback and queries, and create an upload and download page in your application.

19

Creating and Editing Charts

Introducing Visual Aids for Representing Data



In a casual talk about PTS, Stella expresses that along with textual reports, it will be better if PTS can generate charts also, which will make things clearer for any project manager who wants to get a quick snapshot of the project at any point of time. To get this done, Steve looks into creating and editing charts with Oracle Application Express.

This slide shows a graphical representation of the entire course, highlighting lesson 19 in particular, which is dealt with in these slides.

Objectives

After completing this lesson, you should be able to:

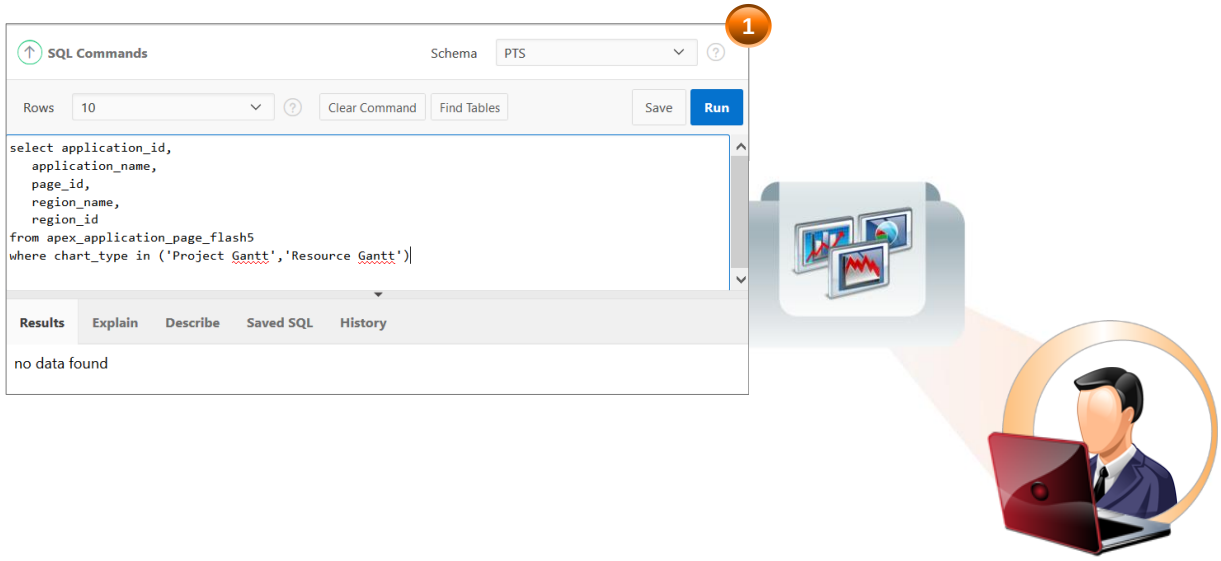
- Switch from AnyChart to JET Chart
  - Upgrade to JET Chart using Application Upgrade Utility
  - Search a workspace for AnyChart using SQL commands
- Create and use charts in applications
- View and edit Chart Attributes
- Explain some additional chart examples

In this lesson, you learn how to create and use charts in applications. You also learn some of the additional charting examples that can be used in your application.

Lesson Agenda

- Switch from AnyChart to JET Chart
  - Search a Workspace for AnyChart Using SQL Commands
  - Upgrade to JET Chart Using Application Upgrade Utility
- Create and Use Charts
- View and Edit Chart Attributes
- Reviewing Additional Charting Examples

Search a Workspace for AnyChart Using SQL Commands



In Oracle Application Express 18.2 and later, the support for charts is based on the **Oracle JET Data** Visualizations. The Oracle JET data visualization components include customizable charts, gauges, and other components that you can use to present flat or hierarchical data in a graphical display for data analysis. Oracle JET charts provide different ways of data visualization such as bar, line, area, range, combination, scatter, bubble, polar, radar, pie, donut, funnel, and stock charts.

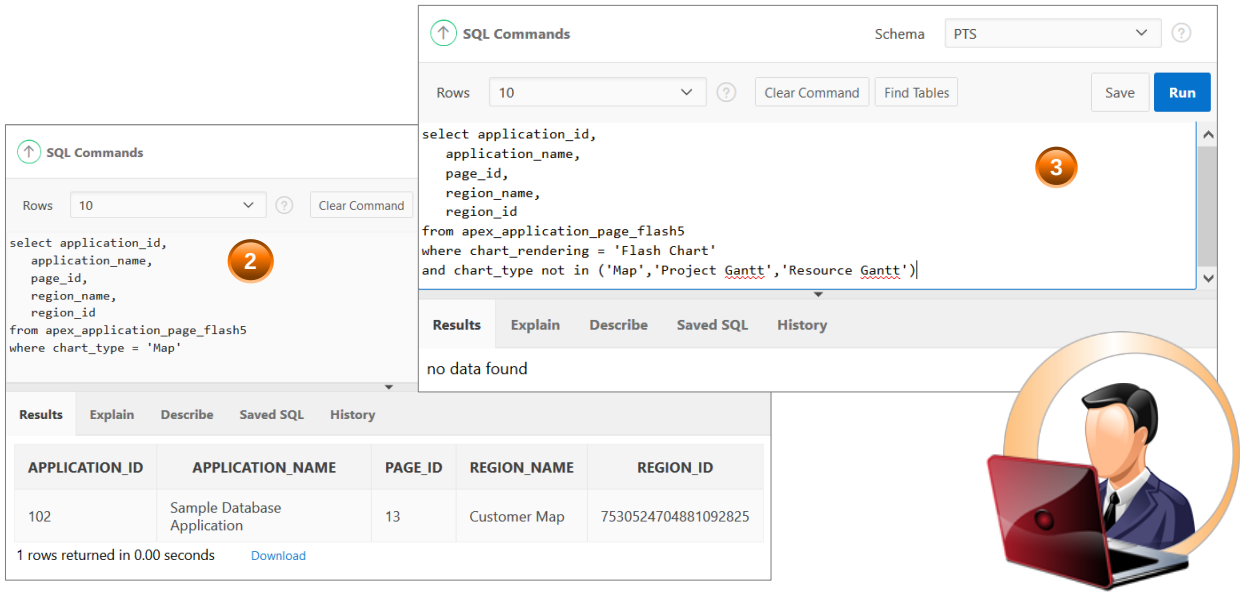
On the other hand, **AnyChart** chart is based on a third-party charting solution provided by AnyChart. This is a flexible Flash and JavaScript (HTML5) based solution that enables you to create animated and compact interactive charts.

So if you have created any charts in a previous version of Oracle Application Express, or if you have imported an application that is created in any previous version of Oracle Application Express, then you can use the following SQL commands to search your workspace for AnyChart components:

- **AnyGantt Chart:** To identify pages that have AnyGantt charts (screenshot 1):

```
select application_id,
       application_name,
       page_id,
       region_name,
       region_id
       from apex_application_page_flash5
where chart_type in ('Project Gantt','Resource Gantt')
```

Search a Workspace for AnyChart Using SQL Commands



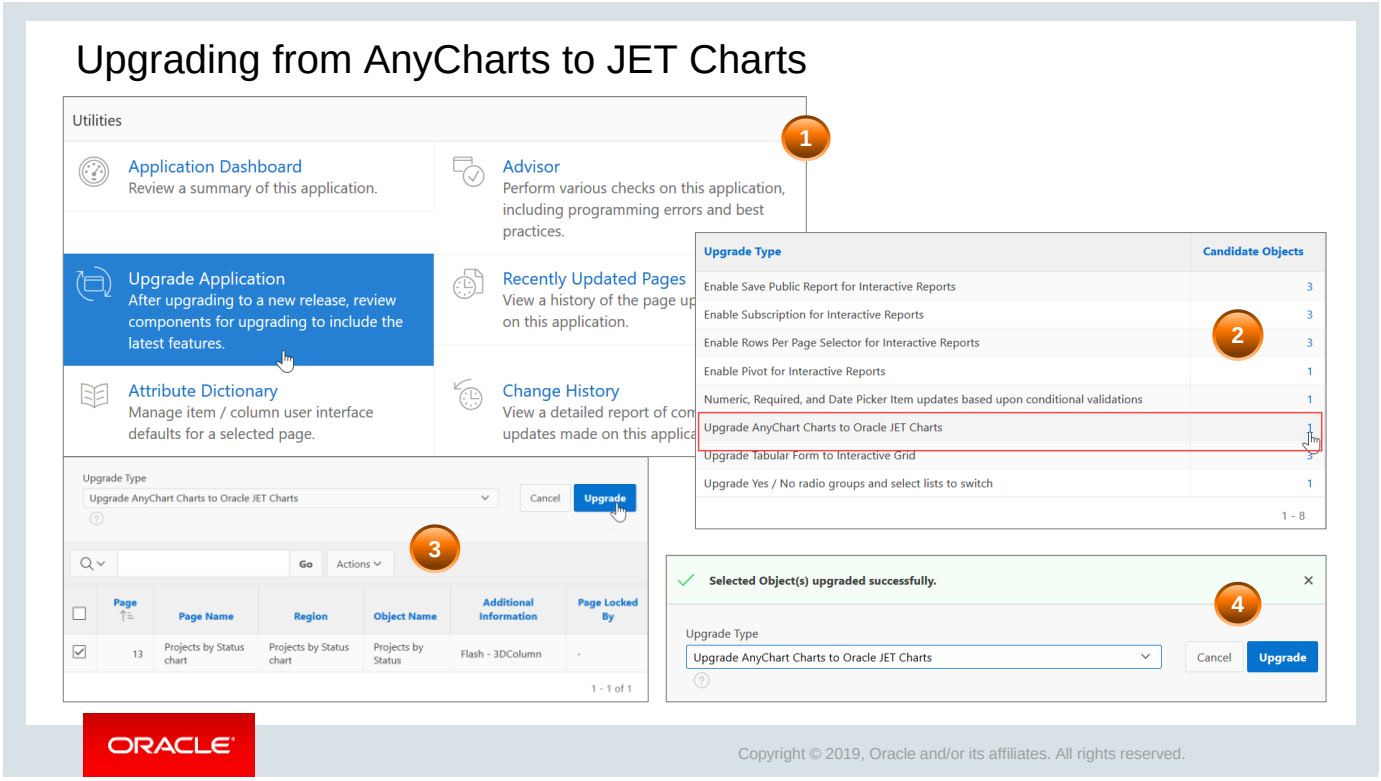
- **AnyChart Maps:** To identify pages that have AnyGantt maps (screenshot 2):

```
select application_id,
       application_name,
       page_id,
       region_name,
       region_id
       from apex_application_page_flash5
where chart_type = 'Map'
```

- **Flash-based AnyChart components:** To identify pages that have Flash-based AnyChart components (screenshot 3):

```
select application_id,
        application_name,
        page_id,
        region_name,
       region_id
from apex_application_page_flash5
where chart_rendering = 'Flash Chart'
and chart_type not in ('Map','Project Gantt','Resource Gantt')
```

Steve uses these commands to search his workspace and finds an AnyChart chart.



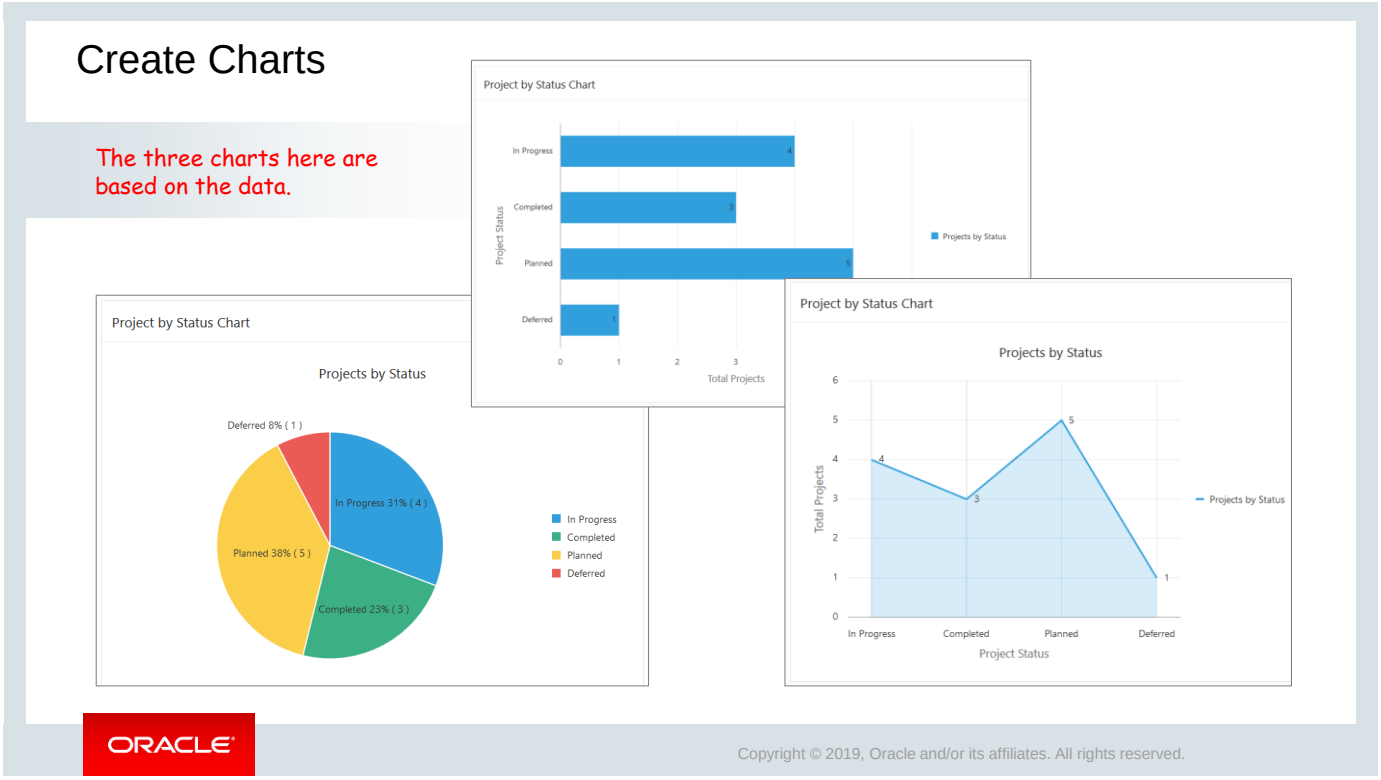
Steve successfully upgrades the AnyChart chart to Oracle JET Charts. If you find any AnyCharts charts in your workspace, then you must upgrade to Oracle JET.

To upgrade from AnyCharts to JET charts:

1. Go to your application home page and click **Utilities**
2. On the Utilities page, click **Upgrade Application** (screenshot 1). The Upgrade Type page opens, which lists the components that require upgrade (screenshot 2).
3. Click the number listed on the right for **Upgrade AnyChart Chart to Oracle JET Charts** (screenshot 2)
4. On the Upgrade AnyChart Chart to Oracle JET Charts page, click **Upgrade** (screenshot 3)
5. After the upgrade is completed successfully, you get the message *Selected Object(s) upgraded successfully* (screenshot 4)

Lesson Agenda

- Switch from AnyChart to JET Chart
- Create and Use Charts
  - Create a Bar Chart
- View and Edit Chart Attributes
- Reviewing Additional Charting Examples



Steve creates a few charts to graphically represent the count of total projects based on project statuses in the PTS application. He uses the built-in wizards for generating JET charts in Oracle Application Express. The charts in the slide include a Pie chart, a Horizontal Bar chart, and a Line with Area chart.

Note: The charts shown on the slide are created using the same data and SQL query to depict total projects for each of the following statuses – Planned, Completed, In Progress, and Deferred.

Creating SQL Queries for Charts

```
SELECT link, label, value
FROM …
```

Example:

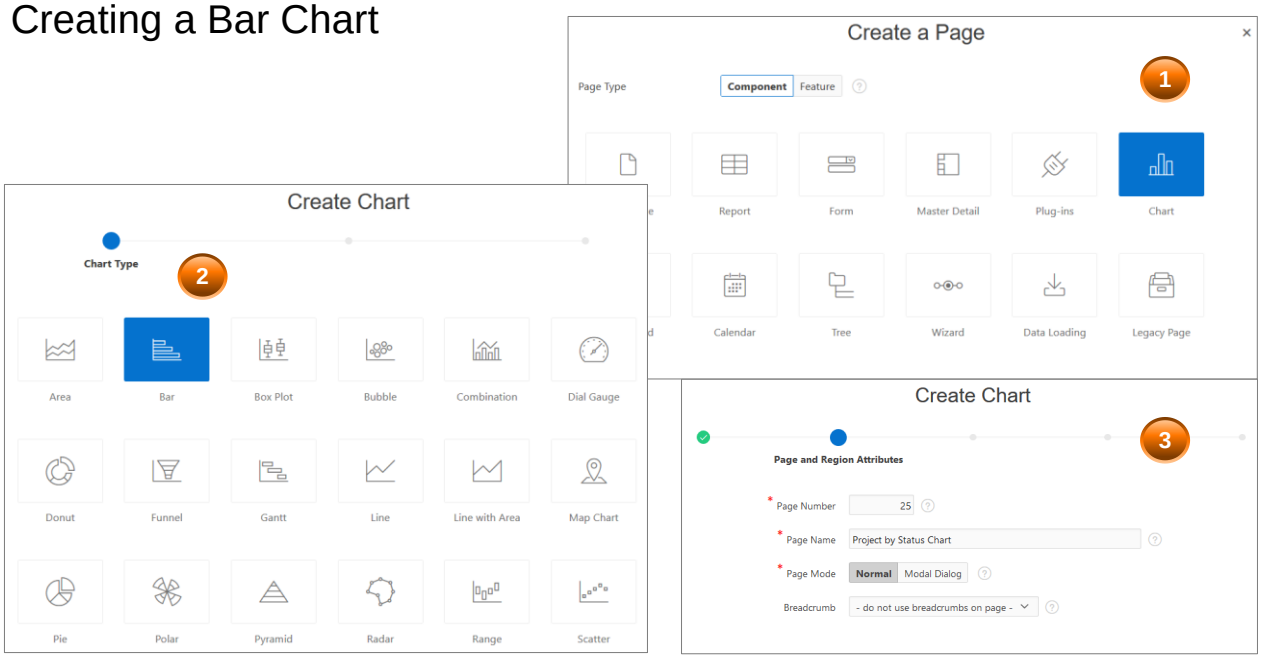
```
select null link, STATUS_NAME label, COUNT(PROJECT_STATUS) value1from
"PTS"."PROJECT_STATUS_VIEW" group by STATUS_NAME
```

To show the number of projects for each possible status in the Project Tracking System, Steve generates the SQL query shown in the slide to build the chart.

You define a chart in Application Builder using a wizard. For most chart wizards, you select a chart type and provide a SQL query by using the syntax shown in the slide. In the syntax,

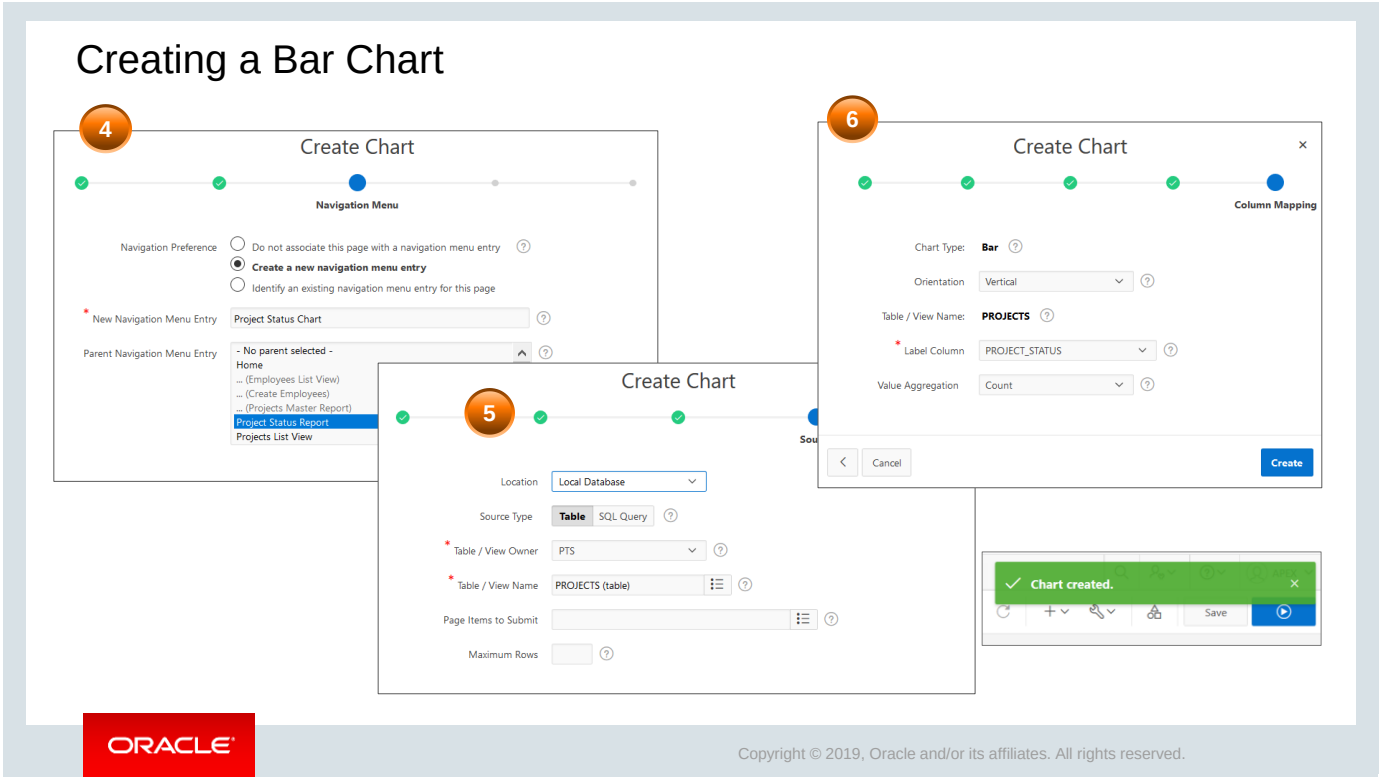
- `link` is a URL
- `label` is the text that displays in the bar
- `value` is the numeric column that defines the bar size

Creating a Bar Chart



Steve creates a Bar chart using the wizard to show the count of projects based on its statuses. To create the Bar chart:

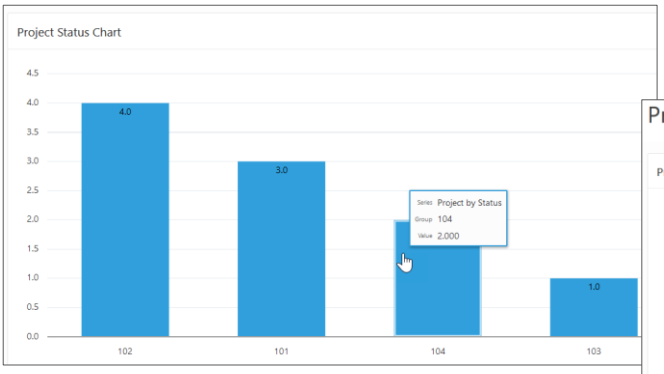
1. Go to the PTS application home page and click **Create Page.**
2. Under Page Type – Components, click **Chart** (screenshot 1).
3. For Chart Type, select **Bar** and click **Next (**screenshot 2).
4. For Page and Region Attributes, enter `25` for Page Number, enter `Project by Status Chart,` and click **Next** (screenshot 3).



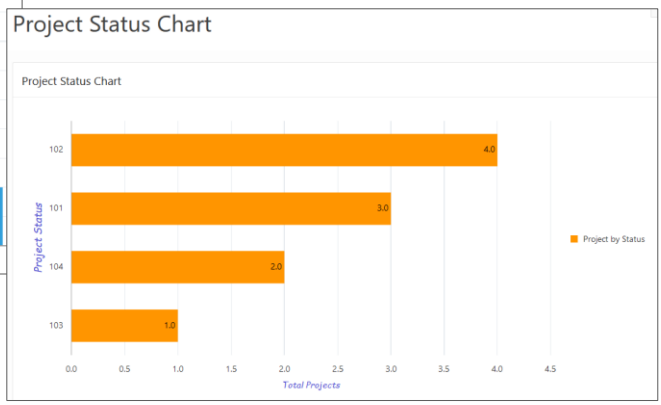
5. For Navigation Menu, select **Create a new navigation menu entry** and click **Next** (screenshot 4).
   - **New Navigation Menu Entry:** Enter `Project by Status Chart.`
   - **Parent Navigation Menu Entry:** Select `– no parent selected –`
6. To define the chart Source, enter the following (screenshot 5) and click Next:
   - **Source Type:** Click **Table**
   - **Table/View Name:** Select `PROJECTS(table)`
7. Define the Column Mapping for the chart as follows:
   - **Label Column:** Select `STATUS_NAME`
   - **Value Aggregation:** Select `Count`
8. Click **Create**. The chart is created, and the page opens in Page Designer.
9. Click **Save** and **Run** to view the chart.

View and Edit Chart Attributes

Vertical Bar Chart



Horizontal Bar Chart



ORACLE

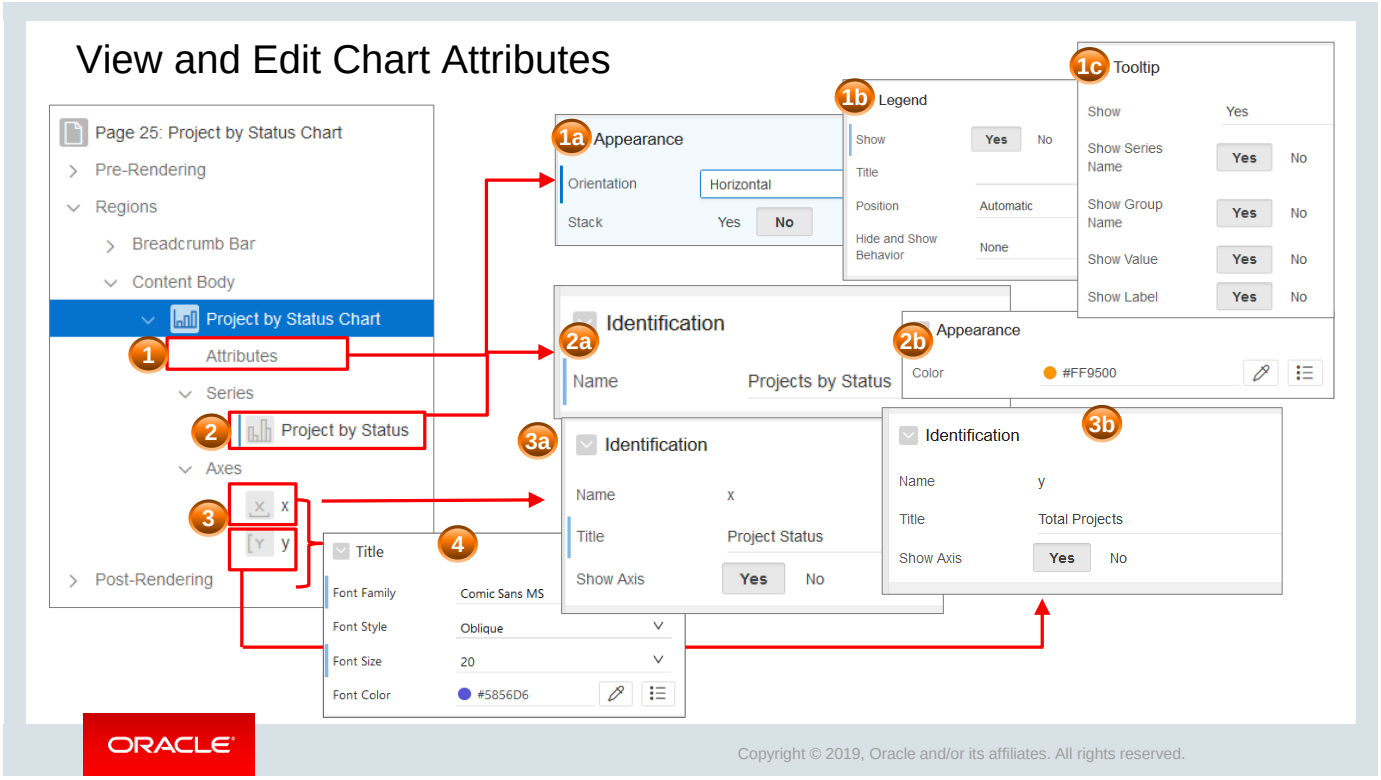
Steve observed that although the Vertical Bar chart is created, it appears very basic. The axes do not have labels. See screenshot 1. Hence, he considers modifying the chart attributes to add some additional information. After modifying some of the chart attributes, this is how the Project by Status chart looks, as shown in screenshot 2. Place the cursor on any of the bars in the chart. The series name (*Project by Chart)*, the group name (in this case – Planned), and the value (5.00) or the number of projects that are in Planned status are shown.

The edits to the chart attributes are described in the section “View and Edit Chart Attributes.”

Steve also decides to render the same data in different charts such as Pie chart and a Line with Area chart, which is described in the subsequent slide.

Lesson Agenda

- Switch from AnyChart to JET Chart
- Create and Use Charts
- View and Edit Chart Attributes
  - Visualizing Data in a Line with Area and Pie Chart
- Reviewing Additional Charting Examples

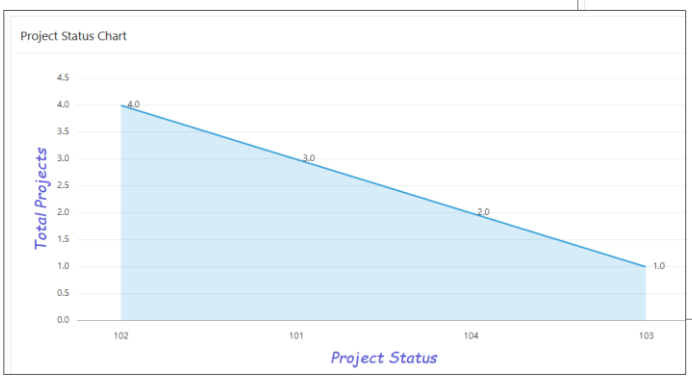


Steve edits the chart attributes to convert it into a Horizontal Bar chart, label the axes, change the color, and show tooltip and legend.

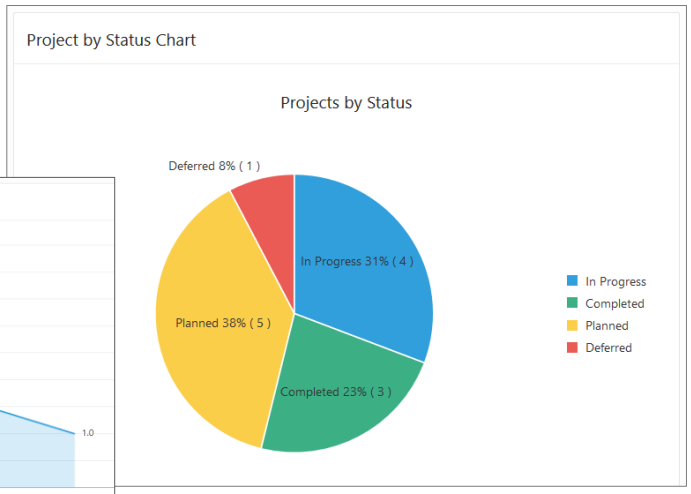
To edit these attributes:

1. Open the *Project by Status Chart* page in Page Designer
2. Go to the Rendering tab and click **Attributes** (1)
3. On the Property Editor for Attributes, make the following changes:
   - **Appearance:** Change the Orientation to **Horizontal** (1a)
   - **Legend:** Click **Yes** for Show (1b)
   - **Tooltip:** Click **Yes** for Show and set **Yes** to all the attributes under Tooltip (1c)
4. Click **Project by Status** under **Series** (2). On the Property Editor for Series:
   - Enter **Project By Status** in the **Name** field (2a)
   - Under **Appearance,** select a color of your choice (2b)
5. Click **x** under Axes (3). For **Title,** enter **Project Status.** (3a)
6. Click **y** under Axes. For **Title,** enter **Total Projects.** (3b)
7. For both x and y axes, define the font and style of the text as shown in screenshot 4. **Font Family:** Comic Sans MS, **Font Style:** Oblique, **Font Size:** 20 and **Font Color:** Select `#5856D6`
8. Click **Save** and **Run.** Now the chart is rendered as shown in screenshot 2 in the section “View and Edit Chart Attributes” earlier in this lesson.

Visualizing Data in a Line with Area Chart and Pie Chart



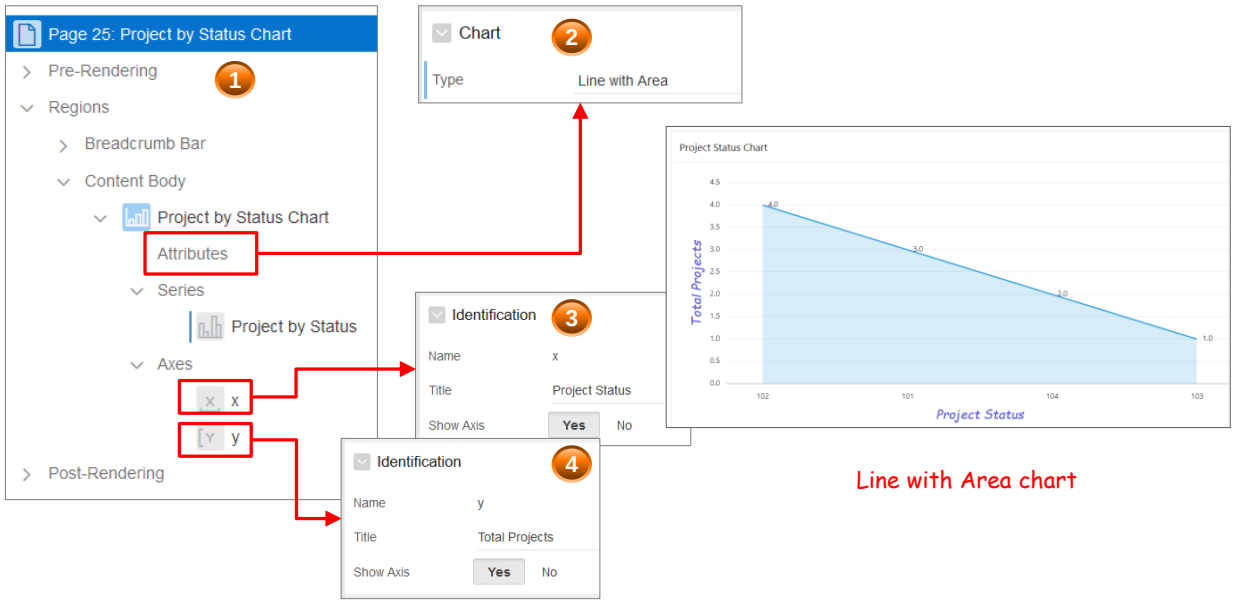
Line with Area chart



Pie chart showing the project count in percentage and count

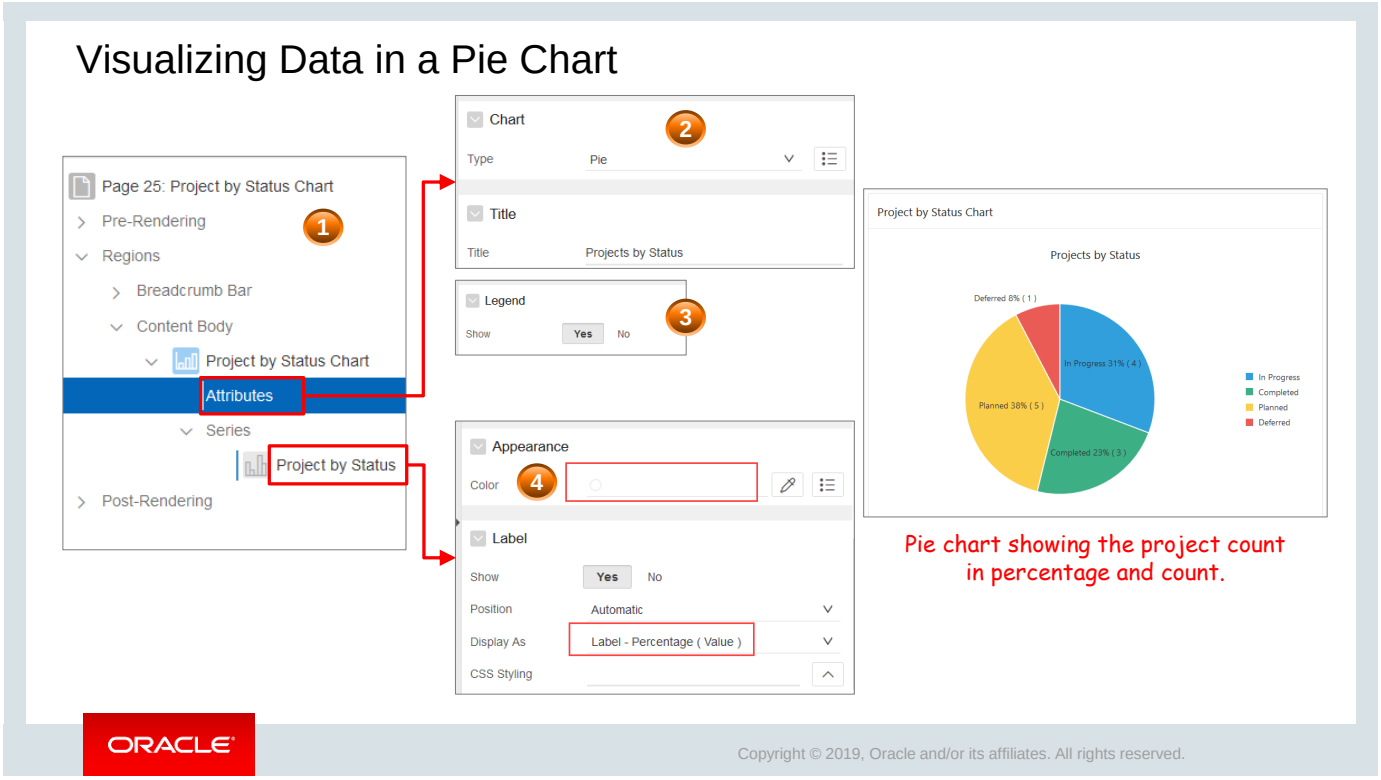
Now Steve uses the same data from the `PROJECTS` table to visualize the data in a Line with Area chart and in a Pie chart.

Visualizing Data in a Line with Area Chart



To visualize the data in a Line with Area chart, click **Attributes** in the Rendering tab (1) and change the Chart Type to **Line with Area** in the Property Editor (2). You can choose to view the chart in default color by removing the color settings from Appearance under Series.

Second, click on each of the axes – **x** and **y**, in the Rendering tab, and in the corresponding Identification section in Property Editor, enter *Project Status* (3) and *Total Projects* (4) in the respective **Title** field.



If you want to depict numerical data in proportions, then Pie chart is the simplest and easiest option. Each slice of the Pie chart in the slide shows the proportion of projects (percentage as well as count) in terms of their status – In Progress, Completed, Planned, and Deferred.

To visualize the data in a Pie chart, click **Attributes** in the Rendering tab and change the chart **Type** to **Pie** in the Property Editor (2). Note that the slices in the Pie chart show the value as percentage as well as the count.

For this, in the Rendering tab, click **Project by Status** under Series. On the Property Tab, scroll down to Label. Under Label, select **Label - Percentage (Value)** in the **Display As** field. To get the default colors for slices in the Pie chart, click Series > Project by Status, and in the Property Editor, go to Appearance and remove the color selection from **Color (4).**

Note that the “Axes” in the Rendering tab under Project by Status Chart are not present for Pie chart, as Pie charts do not use axes. It only represents data as part of a whole or in proportions.

Practice 19-1 Overview: Creating and Editing Charts

This practice covers the following topics:

- Creating a chart page that includes a Horizontal Bar chart
- Modifying your chart and changing it to a Vertical Bar chart

Practice19-2 Overview: Creating a Pie and a Donut Chart

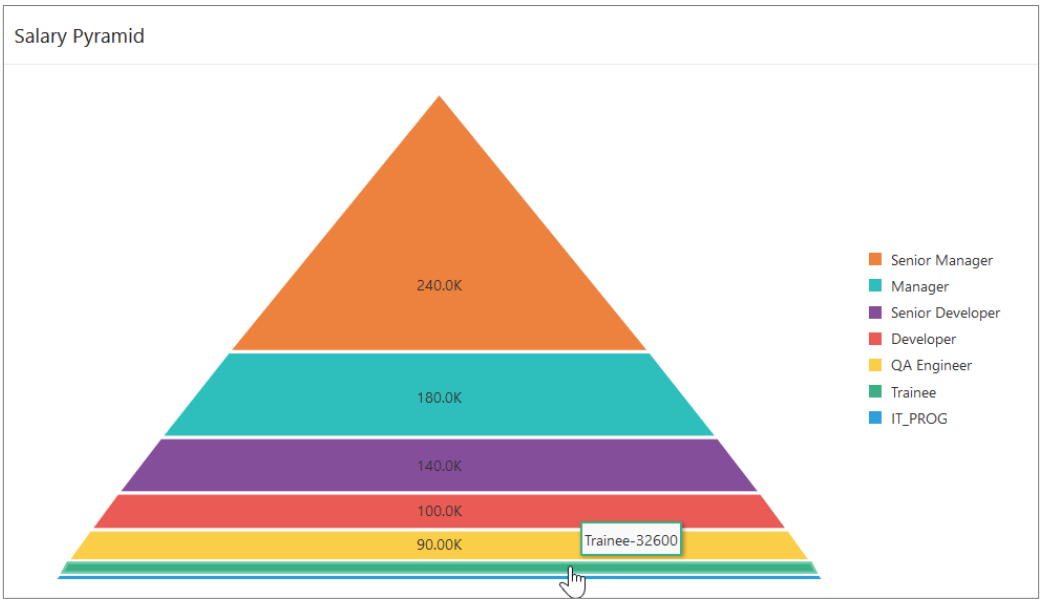
This practice covers the following topics:

- Creating a Pie chart
- Modifying your chart and changing it to a Donut chart

Lesson Agenda

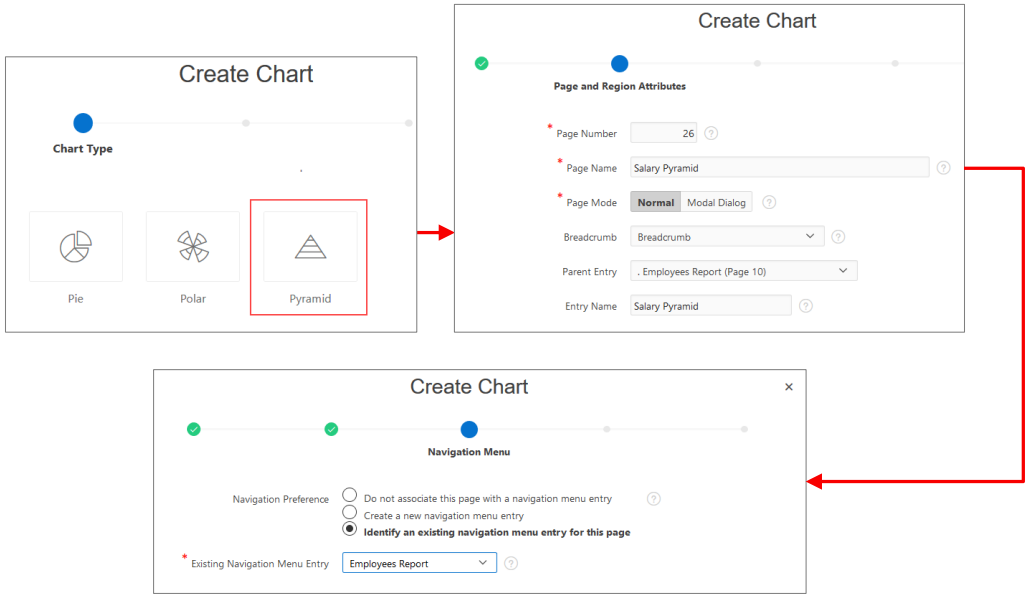
- Switch from AnyChart to JET Chart
- Create and Use Charts
- Review Additional Chart Examples
  - Create a Pyramid chart
  - Create a Status Meter Gauge chart
  - View and analyze a Combination chart

Create a Pyramid Chart



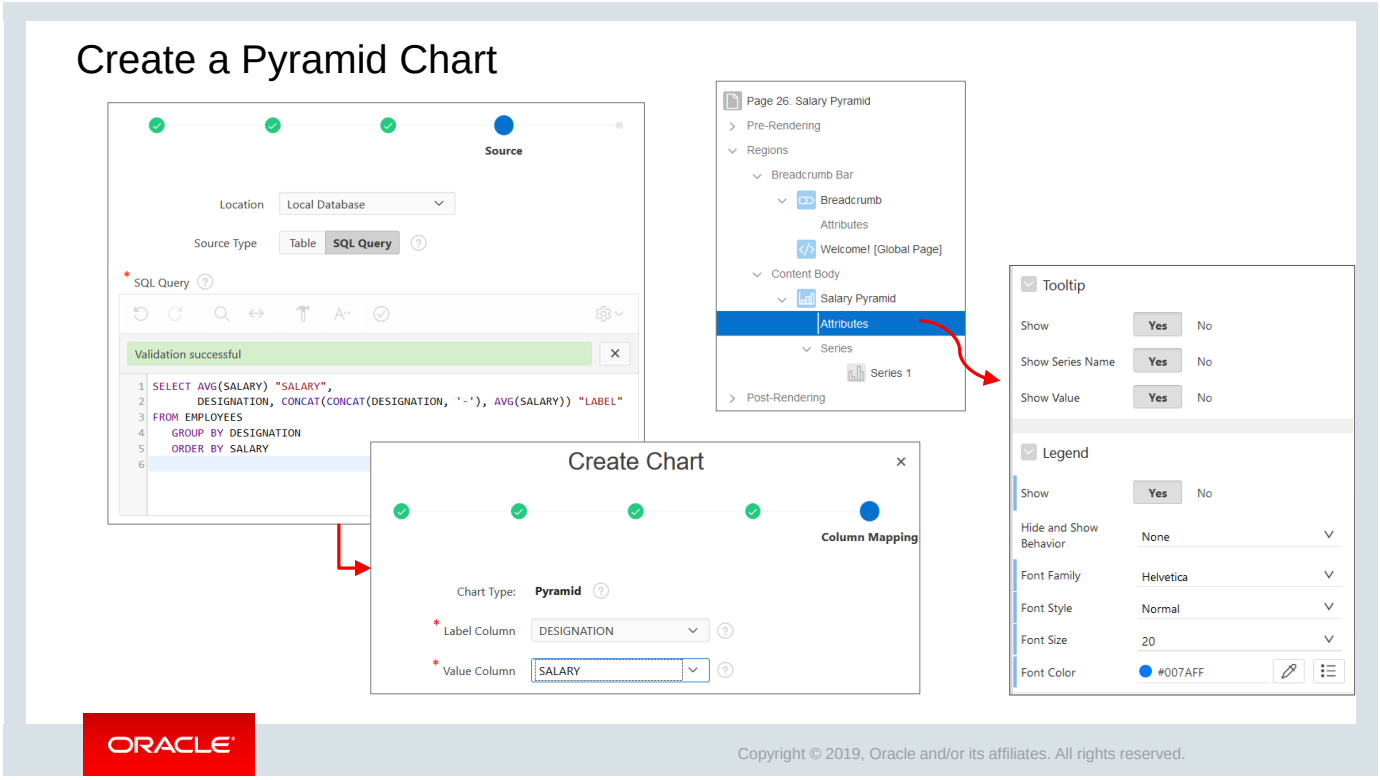
Let us explore the Pyramid chart *Salary Pyramid* that Steve created to portray the salaries of different groups of employees based on their designation. A Pyramid chart consists of different segments, each segment representing a data point. The height of each segment in the pyramid represents the value of the data with respect to the entire pyramid. The *Salary Pyramid* chart portrays the average salary of employees grouped by their designation.

Create a Pyramid Chart



To create a Pyramid chart:

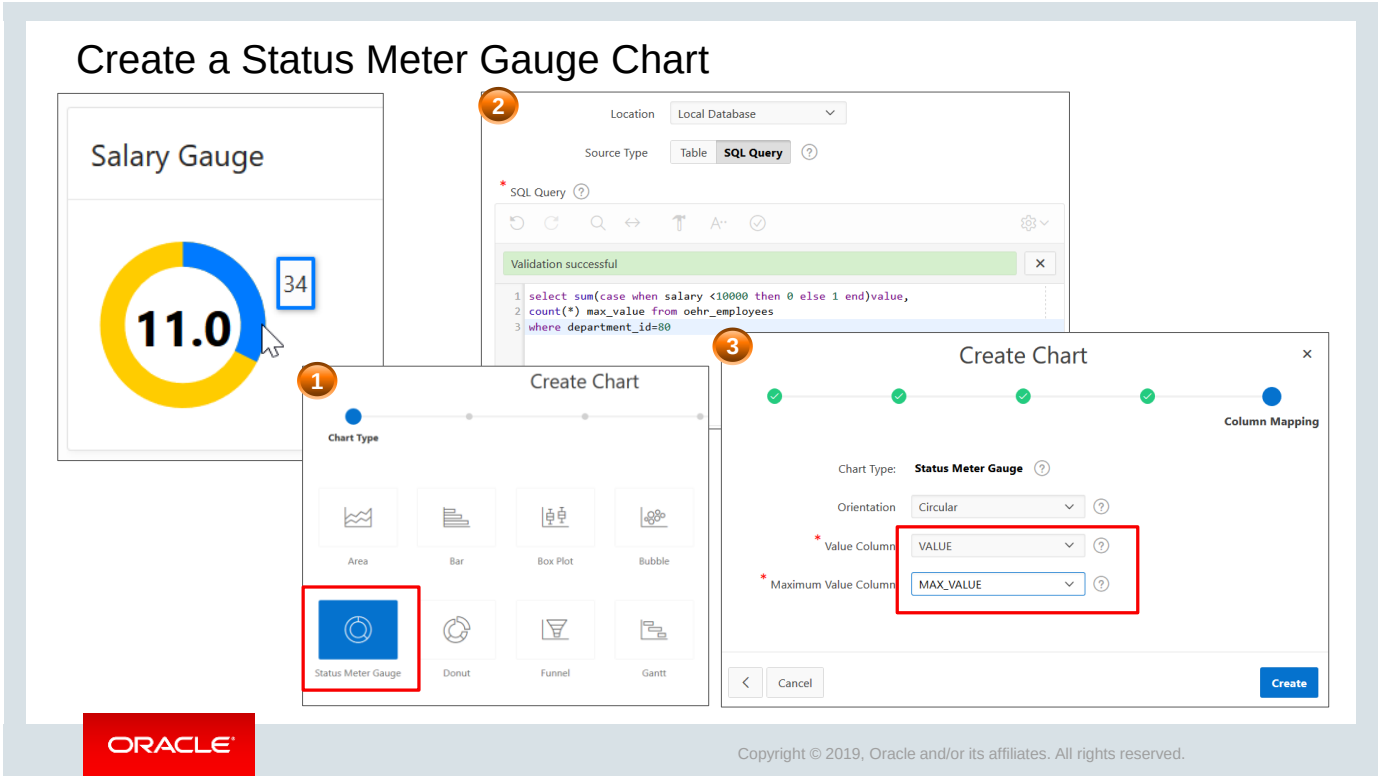
1. Go to the PTS application home page and click **Create Page**
2. Click **Charts**, and then for Chart Type, select **Pyramid**
3. In the Page and Region Attributes, define the following:
   - **Page Number:** `26`
   - **Page Name:** `Salary Pyramid`
   - Select **Breadcrumbs**
   - **Parent Entry:** `Employees Report`
   - **Entry Name:** `Salary Pyramid`
4. For Navigation Preference, select **Identify an existing navigation menu entry for this page**.
   - **Existing Navigation Menu Entry:** Select **Employees Report**
   - Click **Next**



5. For Source, select **SQL** and enter the following SQL Query:

```
SELECT AVG(SALARY) "SALARY",
      DESIGNATION, CONCAT(CONCAT(DESIGNATION, '-'), AVG(SALARY)) "LABEL"
    FROM EMPLOYEES
     GROUP BY DESIGNATION
       ORDER BY SALARY;
```

6. After the SQL query validation, click **Next**.
7. In the last step for Column Mapping, enter `DESIGNATION` for **Label Column** and `SALARY` for **Value Column**.
8. Click **Create.** The page opens in Page Designer.
9. In Page Designer, click **Attributes** on the Rendering tab. In Property Editor, scroll down to **Legend** and click **Yes.** Define the Font style as per your choice. Ensure that for Tooltip, **Show, Show Series Name,** and **Show Value** are set to **Yes.**
10. Click **Save** and **Run.** The Pyramid chart is now displayed, as shown earlier in the section "Create a Pyramid Chart."



The example on the slide shows a JET Status Meter Gauge chart in its circular orientation. It depicts that the salary of 11 employees in a department is greater than or equal to 10000. The total number of employees in the department is 34.

To create a Status Meter Gauge chart, you provide a SQL query by using the following syntax:

```
SELECT value , maximum_value [ ,low_value [ ,high_value] ]
FROM ...
```

To create a Status Meter Gauge chart:

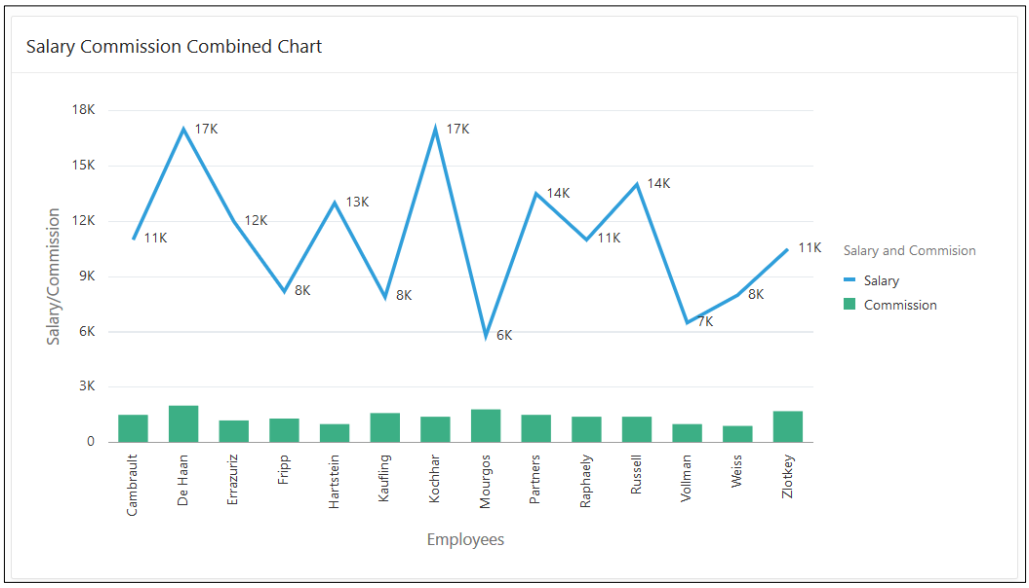
1. In your application, click **Create Page.** Select **Chart** and click **Next.**
2. Select **Status Meter Gauge** (screenshot 1). Specify the page attributes and click **Next.**
3. Accept the default navigation options and click **Next.**
4. To define the chart source, click SQL Query, enter the following SQL and click **Next** (screenshot 2).

```
select sum(case when salary <10000 then 0 else 1 end)
value,count(*) max_value from oehr_employees
where department_id=80
```

5. On the Column Mapping Page, define the value columns and click **Create** (screenshot 3).
   - **Orientation:** Circular (default)
   - **Value Column:** Select `VALUE`
   - **Maximum Value Column:** Select `MAX_VALUE`
6. The Status Meter Gauge chart is created, and it opens in Page Designer. You can edit the chart attributes here. Click **Save and Run** to view the chart.

Note: This chart is created using the `EMPLOYEES` table in `OEHR` schema.

Viewing and Analyzing a Combined Chart



Steve wonders if it is possible to depict two different data sets in one chart. In other words, Steve wonders if creating a combination chart is possible in Oracle Application Express.

Oracle Application Express provides the option to create a chart that is a combination of Line, Bar, and Marker chart types. If you want to show different chart types on the same chart, you must create different data series for the different but combinable data types. The example in the slide shows a combined chart where the Salary series data is displayed as a Line chart and Employees Commission data is displayed as a Bar chart.

In this example, you first create a chart page of Combination chart type. Then you create another series in the chart for the commission. The chart type for the second series for commission is a Bar chart, whereas the chart type for the first series for salary is a Line chart.

This chart is created using the `EMPLOYEES3` table in the `OEHR` schema and is part of Practice 19-3.

Practice19-3 Overview: Enhanced Charting Examples

This practice covers the following topics:

- Building a Combination chart

Quiz

Your chart query syntax looks like the following:

```
SELECT link, label, value
FROM ...
```

In the syntax, value refers to the:

a. Text that is displayed in the bar
b. Column that defines the bar size
c. Starting point
d. URL

Answer: b

Summary

In this lesson, you should have learned how to:

- Switch from AnyChart to Oracle JET Charts
- Create a Bar chart and modify its attributes
- Visualize the Bar chart in a Line with Area chart and in a Pie chart
- Create a Combination chart
- Create a Pyramid chart
- Create a Status Meter Gauge chart

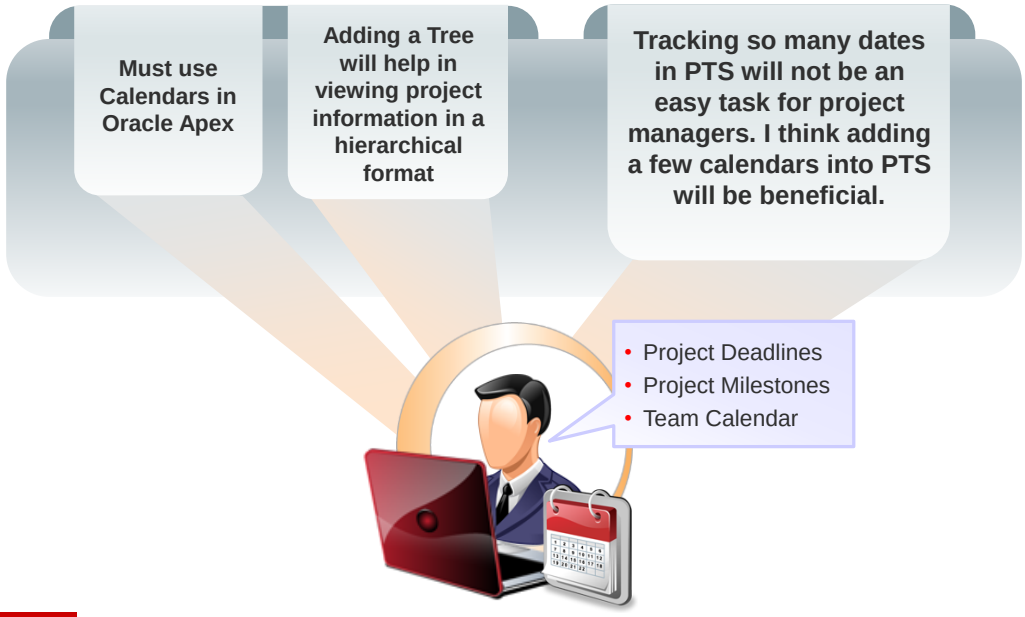
ORACLE

In this lesson, you should have learned about how to search your workspace for AnyChart charts and upgrade to Oracle JET charts, create and edit charts, and some additional chart examples for desktop applications.

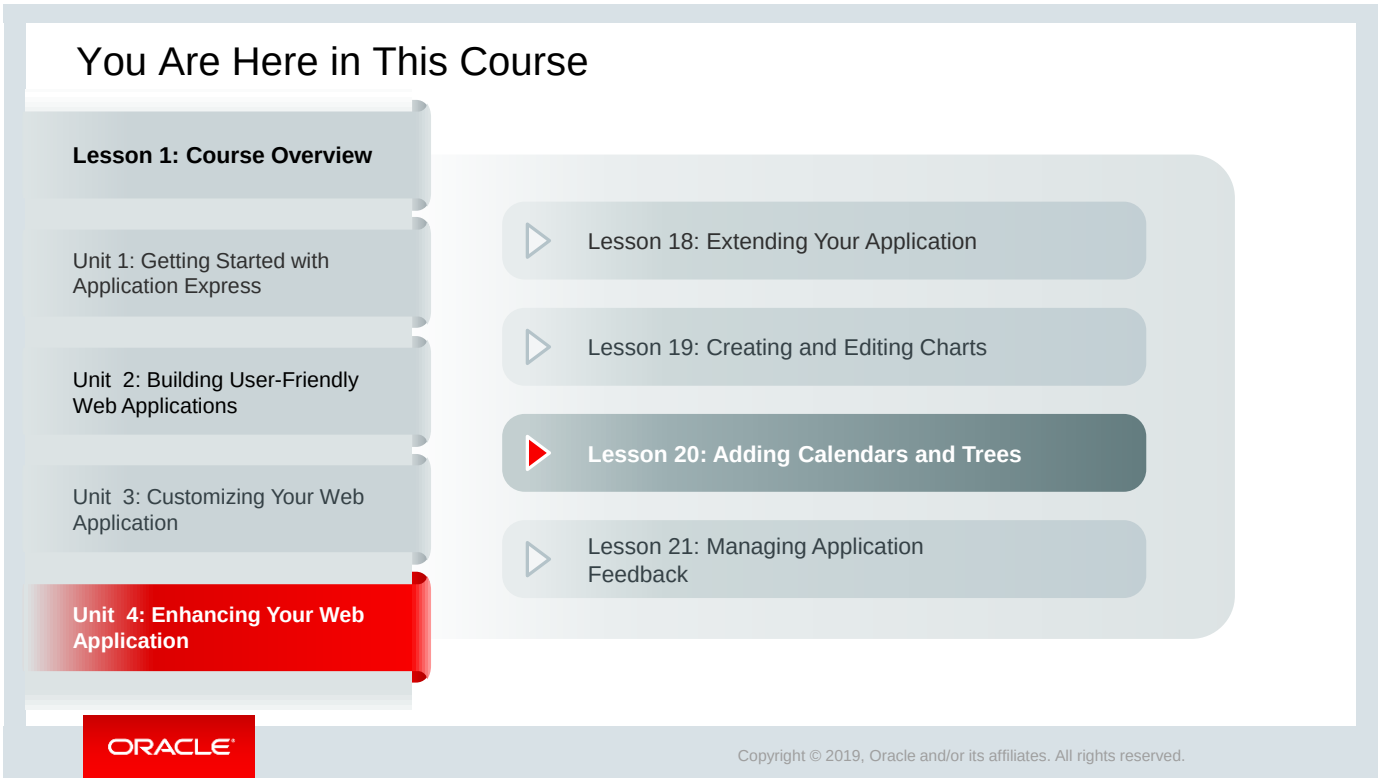
20

Adding Calendars and Trees

Explores Tools to Organize Project Activities



By now, Steve has a good version of PTS application that can be used for real-time project management. While working on PTS with real-time project-related data, Steve observes that timelines and dates are an important aspect of project management tasks. And having clarity on dates will help in identifying tasks that are lagging behind, tasks that are on schedule, deadlines to be met, and other project planning well in advance. He also considers adding a tree to the PTS application to present the project information in a hierarchical format. Therefore, Steve decides to implement the calendars and tree features provided by Oracle Apex into PTS.



This lesson explains how to implement calendars and create trees in your application.

Objectives

After completing this lesson, you should be able to create and manipulate:

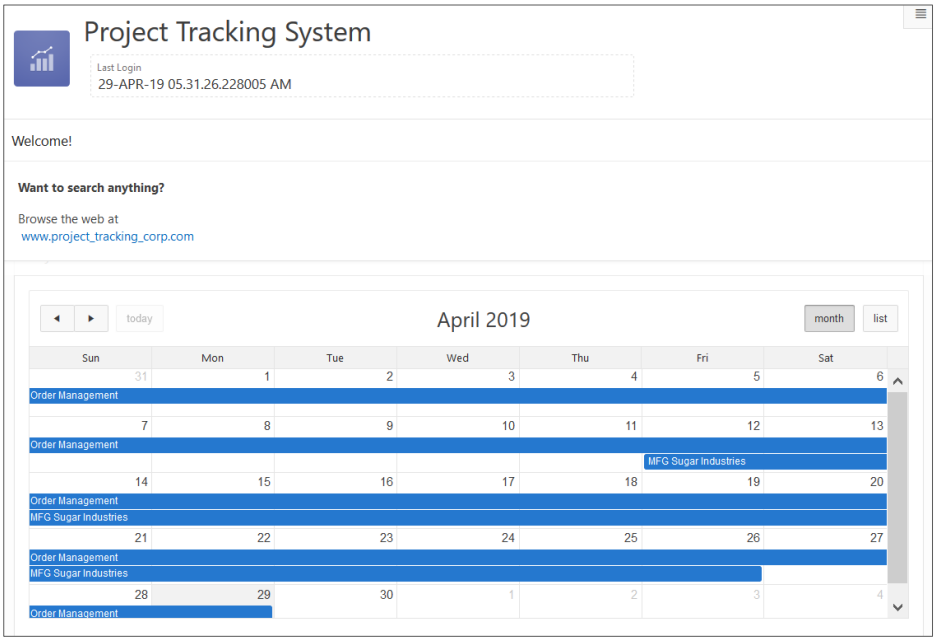
- Calendars
- Trees

In this lesson, you learn how to create calendars and trees in your application.

Lesson Agenda

- Using Calendars
  - Creating a Calendar region on Home Page
  - Editing Calendar Attributes
  - Dragging and Dropping Calendar Entries
  - Linking to the Calendar from a Button
- Using Trees

Creating a Calendar



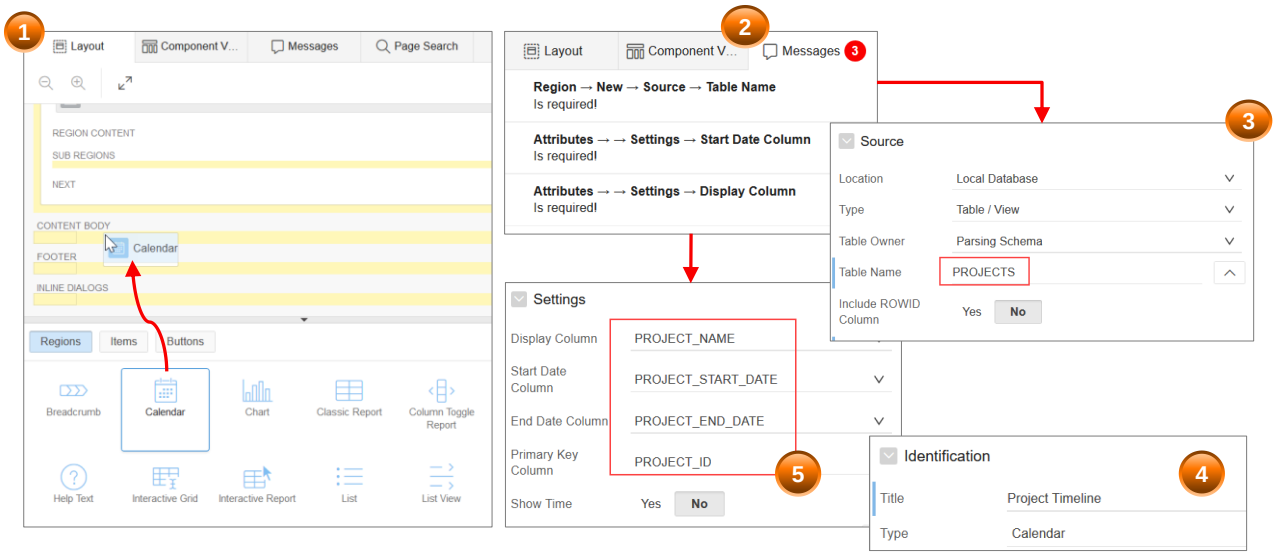
Steve decides to add a calendar as a region to the home page in the PTS application. He calls it Project Timeline.

Oracle Application Express supports calendar that is based on the FullCalendar jQuery library and can be customized only through CSS. While creating a calendar, you need to provide the following basic information:

- **Table** or **SQL query:** The calendar you create is based on a table or SQL query. You can use the wizard with query builder or paste the SQL query directly while creating a calendar.
- **Date Column:** The date column determines which days on the calendar will contain entries.
- **Display Column:** The display column defines a specific row in the table that will display a calendar date. This column holds the text displayed for events on this calendar.

The calendar can be viewed in multiple modes: monthly, weekly, and daily or list. You can create a calendar as a new page or as a new region on an existing page. In the screenshot on this slide, the calendar named Project timeline is created on the home page in the PTS application.

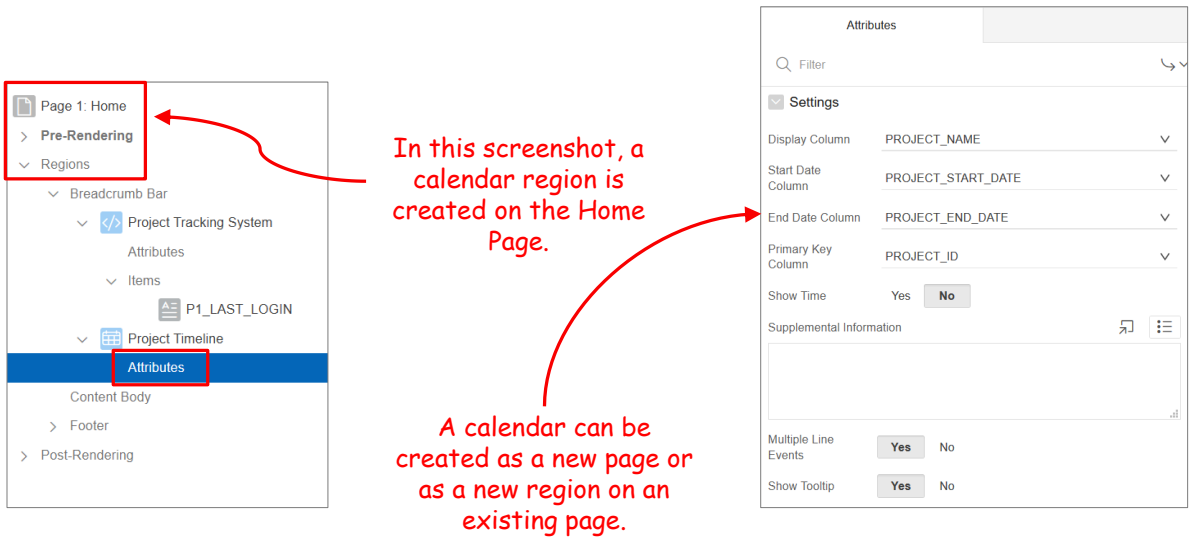
Creating a Calendar on Home Page



To create a calendar on the home page, navigate to your application home page and click **Create Page.** To create the *Project Timeline* calendar that Steve created, as shown in the previous slide, perform the following steps:

1. In the PTS application home page, click 1-Home. The home page opens in Page Designer.
2. From the Region gallery, drag the Calendar region and drop it in the Content Body section in the grid layout, as depicted in screenshot 1.
3. A new region of type Calendar is created with 3 error messages. Click the **Messages** tab (screenshot 2).
4. In the Messages tab, click the first message. It takes you to the Source section in Property editor. In the **Table Name** field, select `PROJECTS`, as shown in screenshot 3.
5. In the Identification section in the Property editor, enter the name `Project Timeline` in the **Title** field, as depicted in screenshot 4.
6. Go back to the Messages tab and click the second message. It opens the Settings section in Attributes. Here, you can define the settings for Display Column, Start Date Column, End Date Column, and Primary Key Column. Select `PROJECT_NAME`, `PROJECT_START_DATE`, `PROJECT_END_DATE`, and `PROJECT_ID` in the **Display Column, Start Date Column, End Date Column,** and **Primary Key Column**, respectively, as shown in screenshot 5. This addresses all the 3 error messages.
7. To show the timeline of projects on the calendar, select `PROJECT_END_DATE` from the drop-down list in the **End Date Column.**
8. Click **Save** and then **Run** the page. The Calendar is now created on the home page in the PTS application as shown in the previous slide. You can modify an existing project by clicking it in the calendar. If you want to create a new project, click on any day in the calendar.

Editing Calendar Attributes

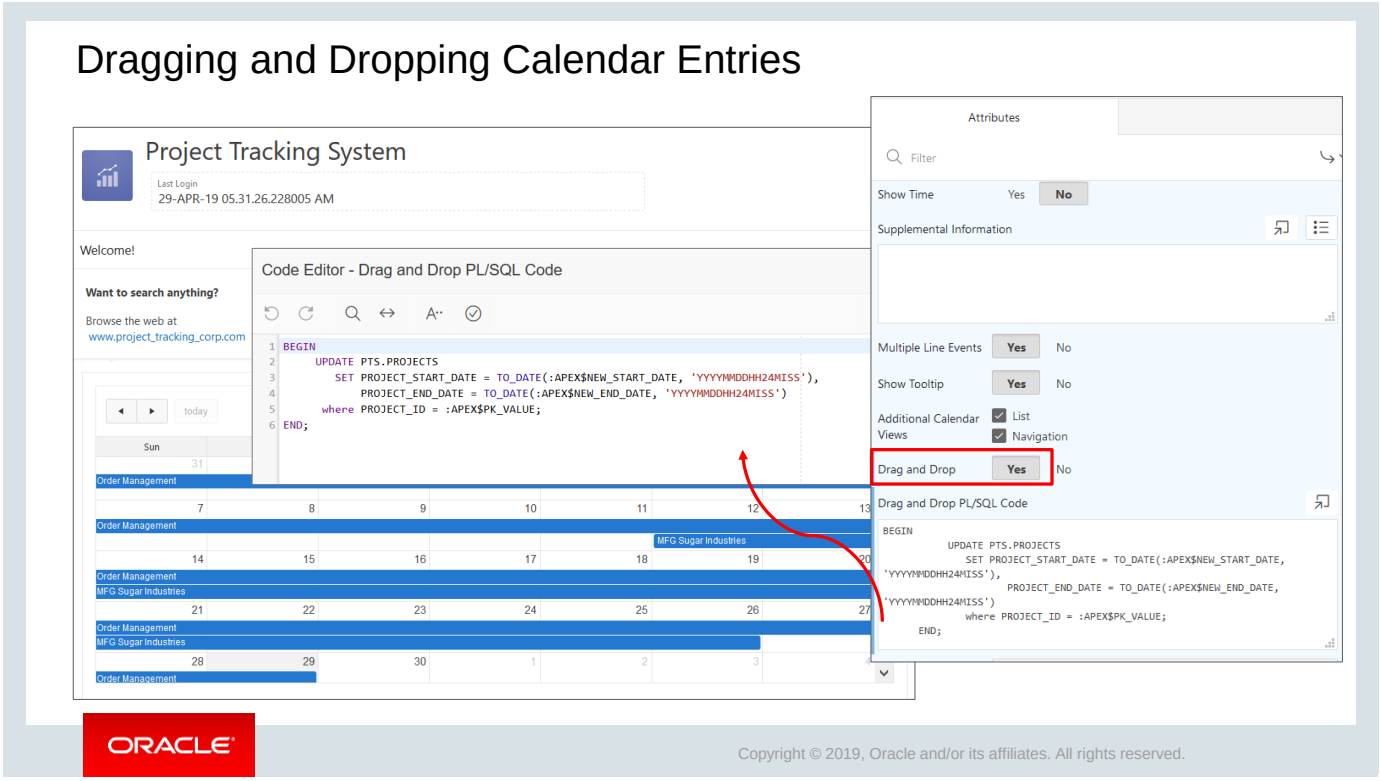


You can use the calendar attributes' Property Editor on the calendar page in the Page Designer view to specify a template, date columns, and general calendar formatting. In addition, you can define the interval in which the calendar is displayed, as well as define the links to be placed on a day or a column in the calendar.

To modify calendar attributes, perform the following steps:

1. Navigate to the page definition where your calendar was created.
2. Open the Calendar page in Page Designer view.
3. Under Rendering, locate **Attributes** under Region and click it. On the right pane, you can see its properties in the Property Editor.
4. Update the required attributes and click **Save** and **Run** the page to notice the changes in the application.

Note on the use of Start Time/End Time**:** If the date column specified does not have a time component (or if individual records have no time), then by default the time is 0:00 hours. It will not be displayed if the start time is set to a later time, for example, 8:00 AM.



Steve considers enabling the drag-and-drop feature in the calendar that he created. By enabling drag-and-drop of calendar entries, you can move a project from one day to another in the calendar itself. When you enable the drag-and-drop feature in a calendar, you must provide a PL/SQL code to update the date of the project in the database.

To enable drag-and-drop:

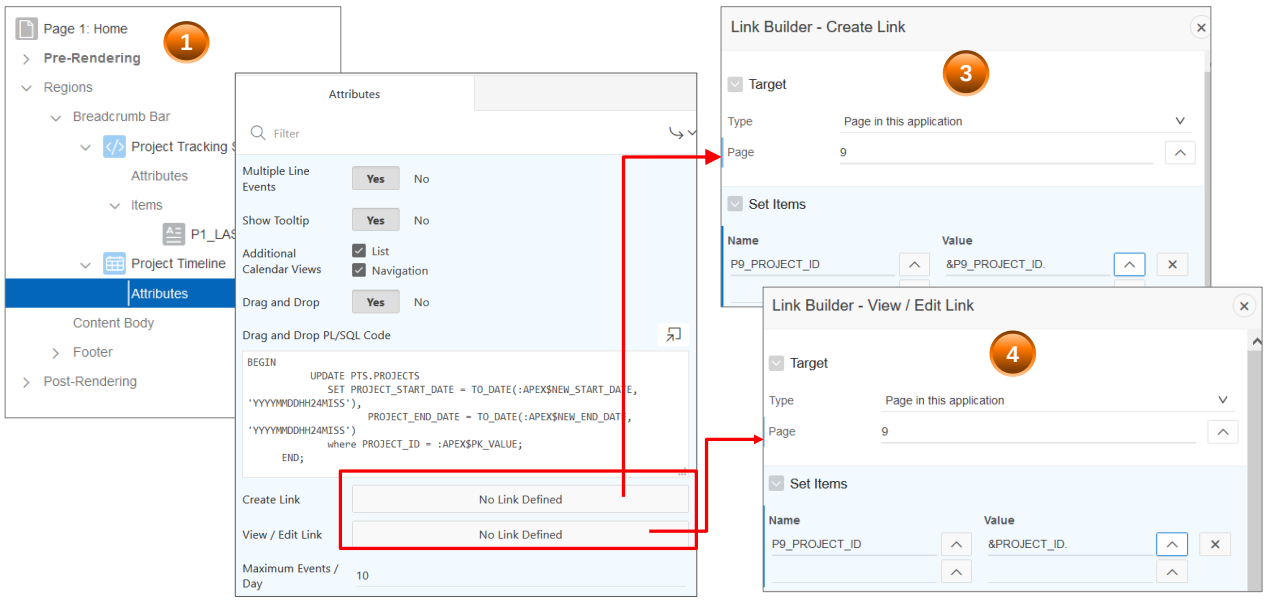
1. Open the Calendar page in Page Designer view.
2. Under Rendering tab, click **Attributes** under Region. On the right pane, in the Property Editor scroll down the Settings section.
3. In the Drag and Drop field, click **Yes.**
4. In the Drag and Drop PL/SQL Code, enter the following:

```
BEGIN
      UPDATE PTS.PROJECTS
           SET PROJECT_START_DATE = TO_DATE(:APEX$NEW_START_DATE,
'YYYYMMDDHH24MISS'),
                  PROJECT_END_DATE = TO_DATE(:APEX$NEW_END_DATE,
'YYYYMMDDHH24MISS')
          where PROJECT_ID = :APEX$PK_VALUE;
END;
```

The query updates the `PROJECTS` table with the new values for `PROJECT_START_DATE` and `PROJECT_END_DATE` after a user drags and drops an entry in the calendar.

5. Click **Save** and **Run.** Now in the calendar, the drag-and-drop feature is enabled, as you can see in the screenshot. You can drag any calendar entry from one date and drop it on another date.

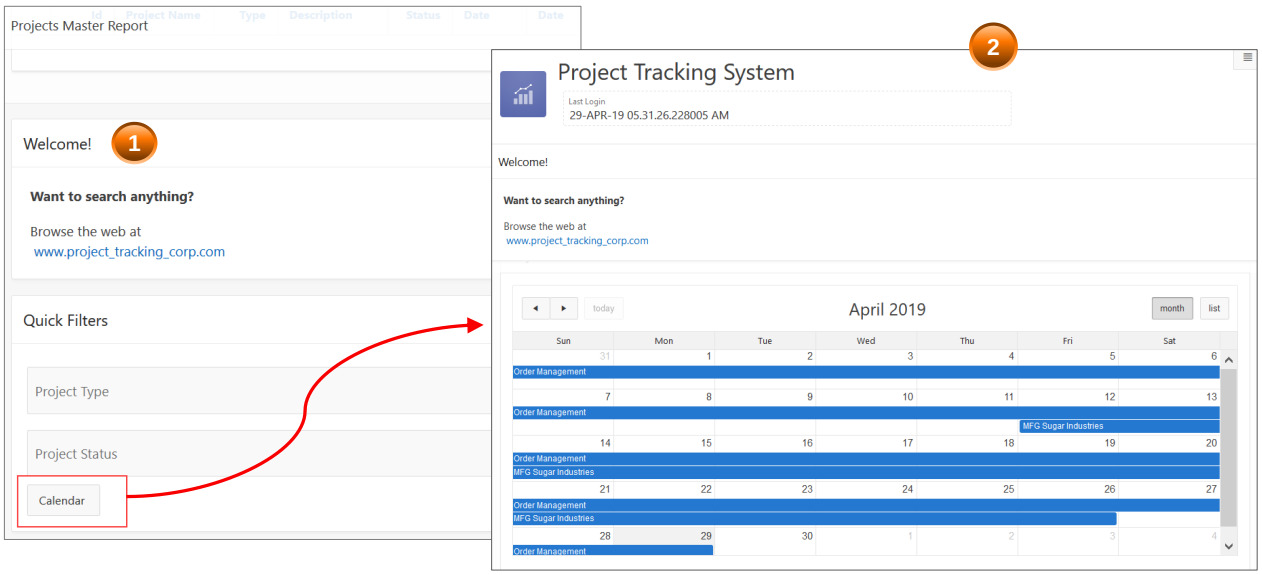
Defining Links in a Calendar



Oracle Application Express also allows you to define links that you can place on a day or on a column in the calendar. In the Projects Timeline calendar, Steve defined links to Page 9: Manage Projects. For this, he defined the pages to be linked for both the Create Link and View/Edit Link options so that on clicking an entry or a day in the calendar, it opens the Manage Project page where you can create a new project or modify the selected project. To define links:

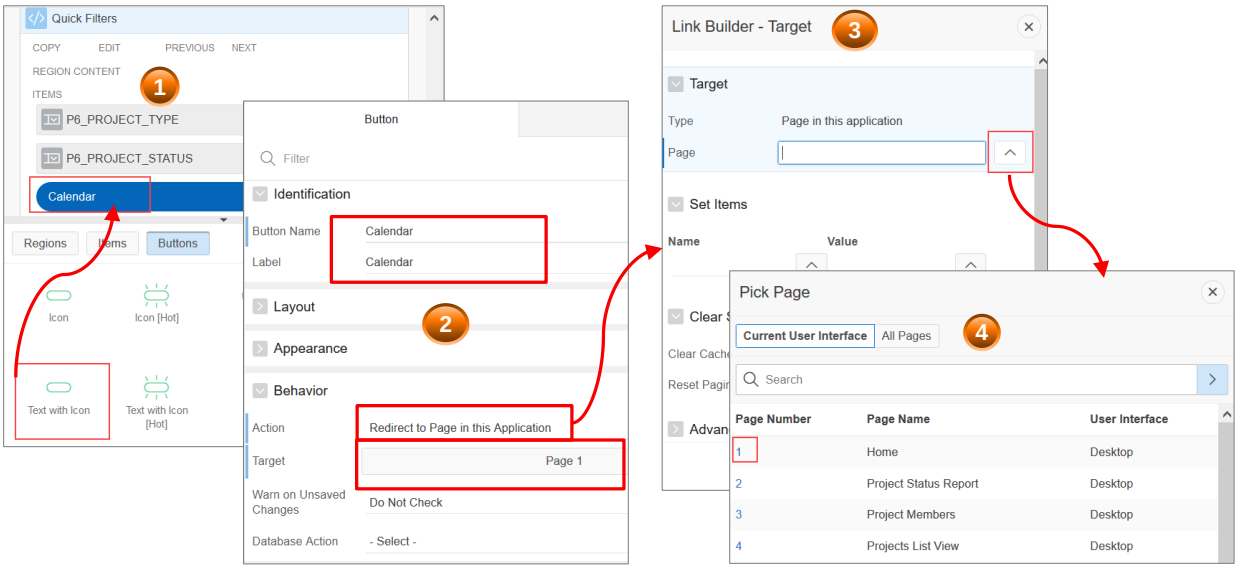
1. In the PTS application, open Page 1 – Home in Page Designer, expand Regions, and click **Attributes** on the Rendering tab (screenshot 1).
2. In the Property Editor, click Create Link. The Link Builder – Create Link dialog opens (screenshot 2).
3. In the Link Builder – **Create Link** dialog, define the following (screenshot 3):
   - **Type:** Select **Page in this application**
   - **Page:** Select **Page 9 – Manage Projects**
   - **Set Items:** Select `P9_PROJECT_ID` and `&P9_PROJECT_ID`
4. Once again, in the Property Editor, click **View/Edit Link**. The Link Builder – View/Edit Link dialog opens (screenshot 2).
5. In the Link Builder – View/Edit Link dialog, define the following (screenshot 4):
   - **Type:** Select Page in this application
   - **Page:** Select Page 9 – Manage Projects
   - **Set Items:** Select `P9_PROJECT_ID` and `&P9_PROJECT_ID`
6. Click **Save** and **Run**. Now, when you click on any date in the Project Timeline calendar, it opens the Manage Projects page where you can define an entry or modify any attribute, as applicable.

Linking to the Calendar from a Button



Since all projects are closely related to timelines, Steve considers linking the Project Timeline calendar in the Project Master Report. As you can see in screenshot 1, he created the Calendar button on the Project Master Report that links it to the Project Timeline calendar. Clicking this button opens the Project Timeline calendar.

Linking to the Calendar from a Button



You may also want to link a calendar from a page in an application. In this example, you create a button on the Project Master Report page that links to the calendar page. To do so, perform the following steps:

1. Navigate to the Projects Master Report page (page 6 in the PTS application) and open it in the Page Designer.
2. Drag a button from Buttons gallery to the Report region in Grid Layout, as shown in screenshot 1.
3. Update button properties such as name and label in the Property Editor (screenshot 2). Enter *Calendar* in both the **Button Name** and **Label** fields.
4. Under Behavior, define the following:
   - In the **Action** field, select **Redirect to Page in this Application**, as shown in screenshot 2.
   - In the **Target** field, select **Page 1,** that is, the Calendar page, as depicted in screenshots 3 and 4.
5. Click **Save** and **Run.** Now, in the Project Master Report, you can see the Calendar button. When you click the Calendar button on the Project Master Report, it opens the Project Timeline, as shown in the previous slide.

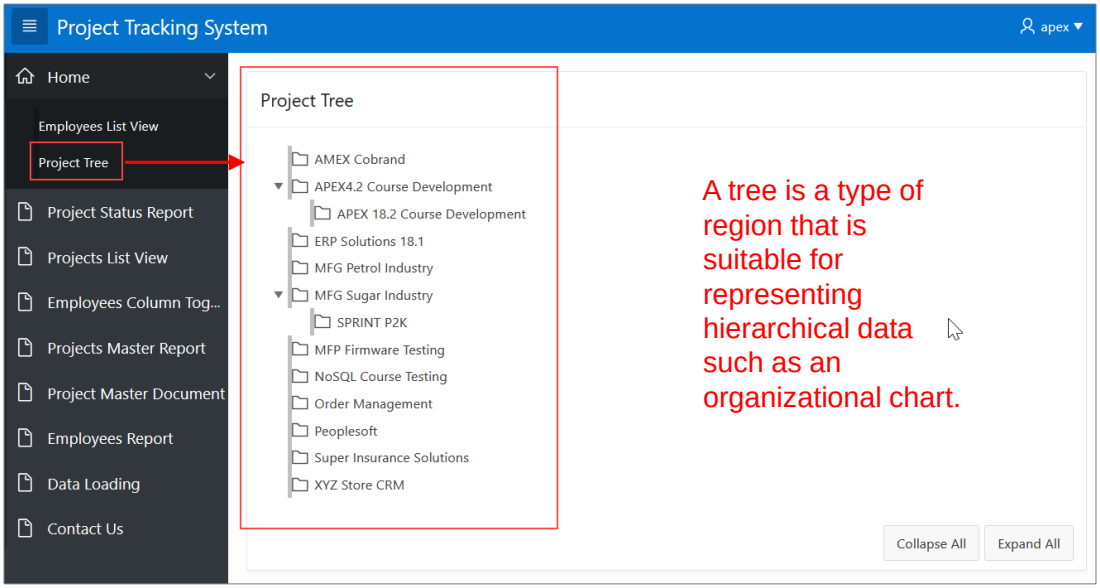
Practice 20-1 Overview: Creating a Calendar

This practice covers creating a calendar for a database application.

Lesson Agenda

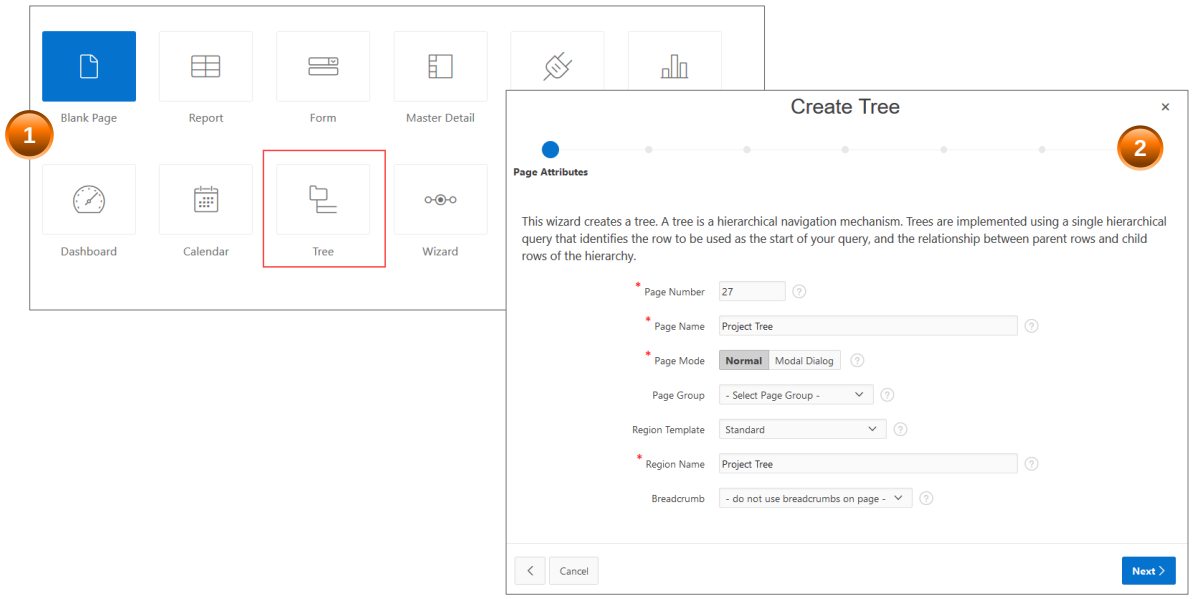
- Using Calendars
- Using Trees
  - What is a Tree?
  - Creating a Tree
  - Exploring a Tree

What Is a Tree?



A tree is a graphical presentation of a hierarchical relationship that is based on a table or view. Steve created a tree view of the projects that are included in the PTS application. As you can see in the screenshot, he created the tree named *Project Tree* under the home page. By clicking *Project Tree,* you can view all the projects that are listed in the PTS application and the hierarchical relationship among the projects.

Creating a Tree

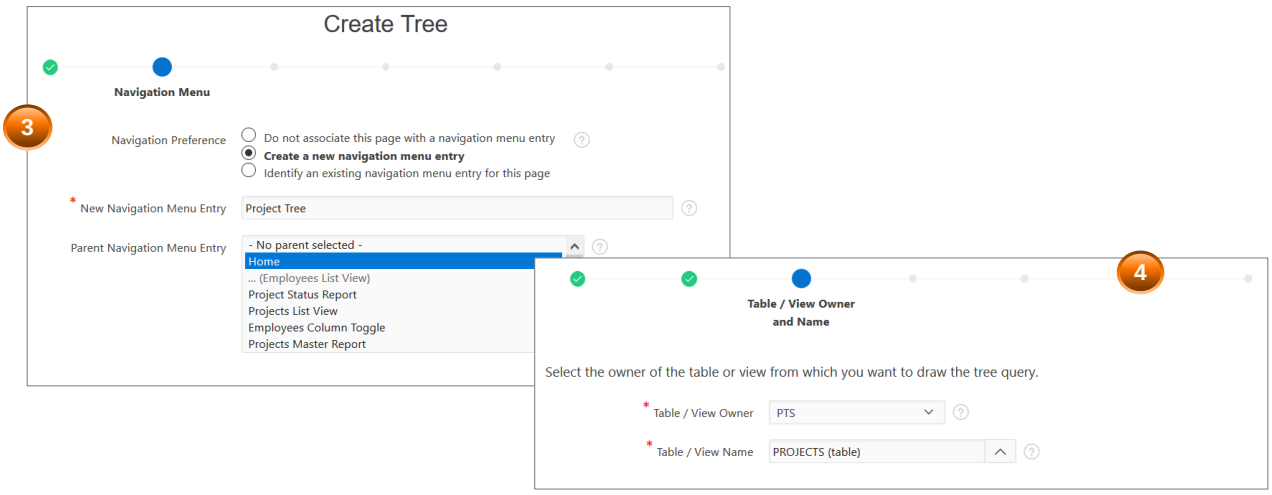


You can create a tree, a navigation mechanism, in your application to present hierarchical data. You can create a tree from a query by identifying an ID and a parent ID in a table or a view. A tree definition contains a starting point and is displayed in a region on a page. The tree can also be referenced by multiple regions.

When you create a tree, it can be included on a new page or added to an existing page. To create a tree under an existing page, navigate to the application home page and select **Create Page.**

1. Select **Tree** and click **Next.**
2. Enter a Page Number, Page Name, and Region Name and click **Next** (screenshot 2).

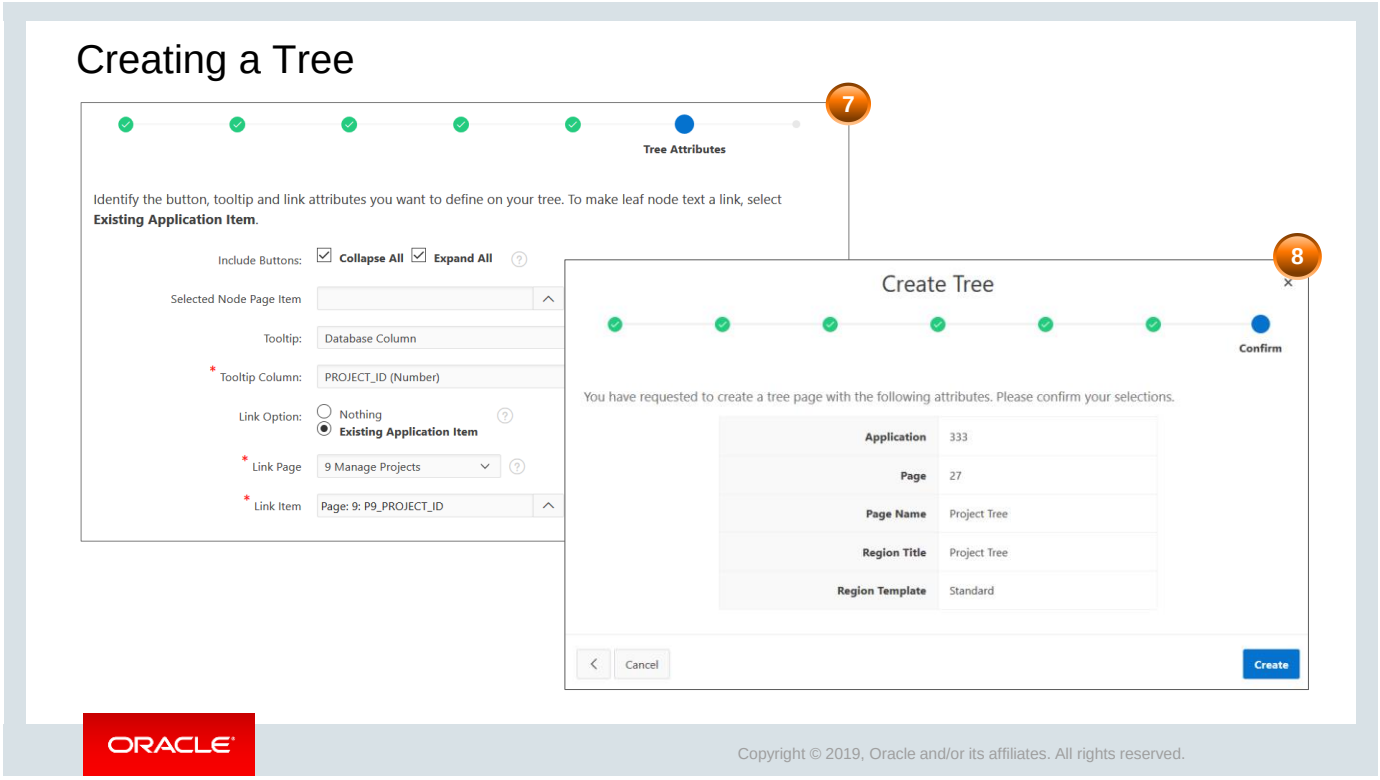
Creating a Tree



3. For Navigation Preference option, click **Create a new navigation menu entry.** Note that the **New Navigation Menu Entry** field is automatically updated with the entry *Project Tree.*
4. Select *Home* for **Parent Navigation Menu Entry**, as shown in screenshot 4. This creates the tree under the PTS application home page. Click **Next.**
5. Select the table on which you want to create the Tree, as shown in screenshot 4, and click **Next.**

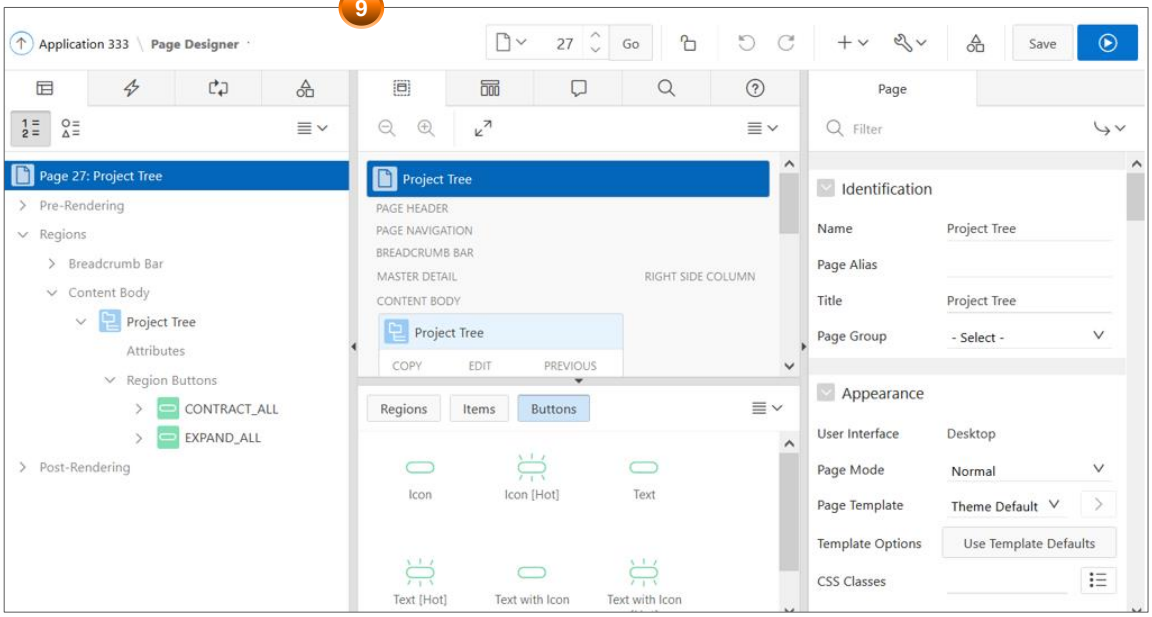
6. For Query, select the columns for the following to include in the tree and click **Next** (screenshot 5):
   - **ID:** Select the column to base the tree on, in this case, `PROJECT_ID`.
   - **Parent ID:** Select the column to use as the parent ID, in this case, `PROJECT_UPGRADE_OF`.
   - **Node Text:** Select the text to appear on the tree nodes, in this case, `PROJECT_NAME`.
   - **Start With:** Select the column to be used to specify the root of the hierarchical tree query; in this case, select `PROJECT_UPGRADE_OF`.
   - **Start Tree:** Choose how to start your query; in this case, select Value is null.
7. You can specify a **`Where` Clause** and **`Order Siblings By`** clause and click **Next.** In addition, you can see the query that was generated by clicking the expand icon for Current Query. The **Where Clause** is used to filter the items to be displayed in the tree, and the value for **Order Siblings By** determines the display order of the items in the tree hierarchy. In this example, in the **Order Siblings By** field, select `PROJECT_NAME (Varchar2)` (screenshot 6).

   Note that `connect_by_isleaf` is a pseudocolumn, and `connect by prior` specifies a condition that identifies the relationship between parent rows and child rows in the hierarchy. The `START WITH` clause identifies the row or rows to be considered for the starting point of the hierarchy.



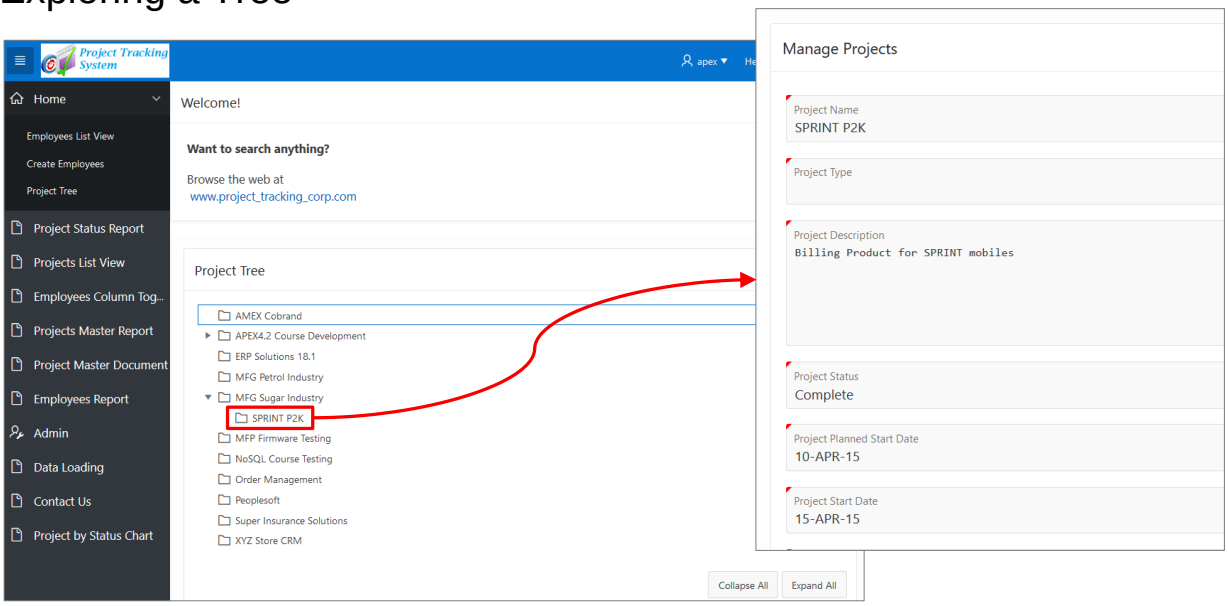
8. When defining the Tree attributes, you can specify the following (screenshot 7):
   - **Include Buttons:** Specify whether you want to include buttons for **Collapse All** and **Expand All** by selecting these options.
   - **Selected Node Page Item:** Select the page or application item to hold the selected node value. This item can be used to save the tree state, by holding the value of the last selected node. When the tree is reloaded, the tree opens to the last selected tree node. In this example, no page item is selected for this option.
   - **Tooltip Column:** Select whether you want to display tooltips. If yes, select the source for the tooltip.
   - **Link Options:** Defines the link to be executed when a node is clicked. In this example, select **Existing Application Item.**
   - **Link Page:** Select the specific page in the application to which you want to link the tree. In this example, Page 9 – Manage Projects is selected.
   - **Link Item:** This is the specific item in the page. In this example, `P9_PROJECT_ID` is selected.
   - Click **Next.**
9. Click **Create.** The Project Tree page now opens in Page Designer view.

Creating a Tree

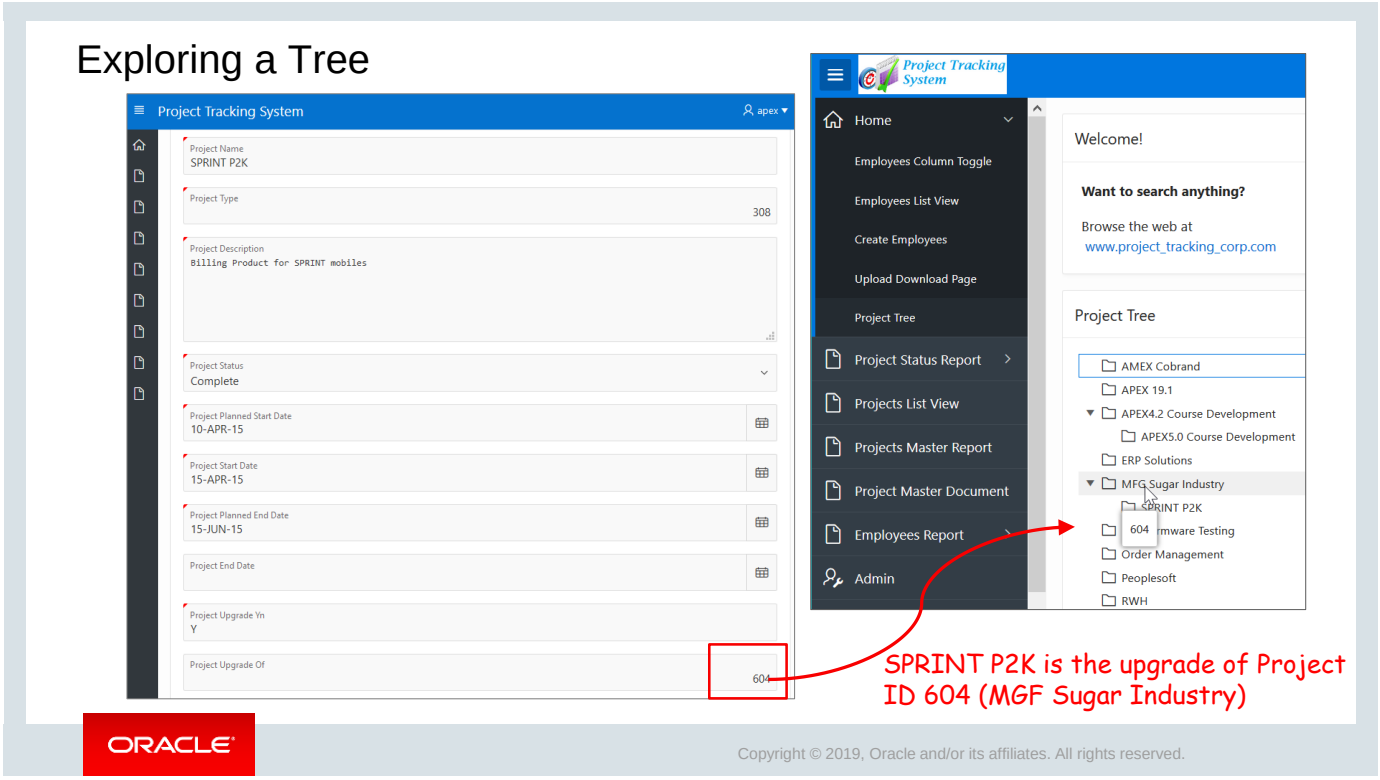


This slide shows the Project Tree page opened in Page Designer view. Click **Save** and **Run.**

Exploring a Tree



The tree displays all the projects listed in the PTS application and their hierarchy. In the example on the slide, you see that the parent project for SPRINT P2K is MGF Sugar Industry project. When you click SPRINT P2K, the Manage Projects form opens.



In the Manage Forms page, note the **Project Upgrade Of** number. In the Project Tree, hover your cursor on MGF Sugar Industry, which is the parent project of SPRINTP2K. When you hover it over MGF Sugar Industry, it displays the Project ID 604, which is the same value that is entered in the Project Upgrade of field in the Manage Projects form for SPRINTP2K. The project SPRINTP2K (project ID 608) is the upgrade of MGF Sugar Industry project (project ID 604).

Practice 20-2 Overview: Creating a Tree Whose Nodes Link to a Different Page

This practice covers the following topics:

- Creating a new page with a tree region and linking it to another page
- Adding a button on a page and navigating back to the tree page using the button

Summary

In this lesson, you should have learned how to create:

- A calendar – You learned how to create the *Project Timeline* calendar that is based on the PROJECTS table used in the PTS application.
- A tree – You learned how to create Project Tree that displays the hierarchical view of the projects that are included in the PTS application.

ORACLE

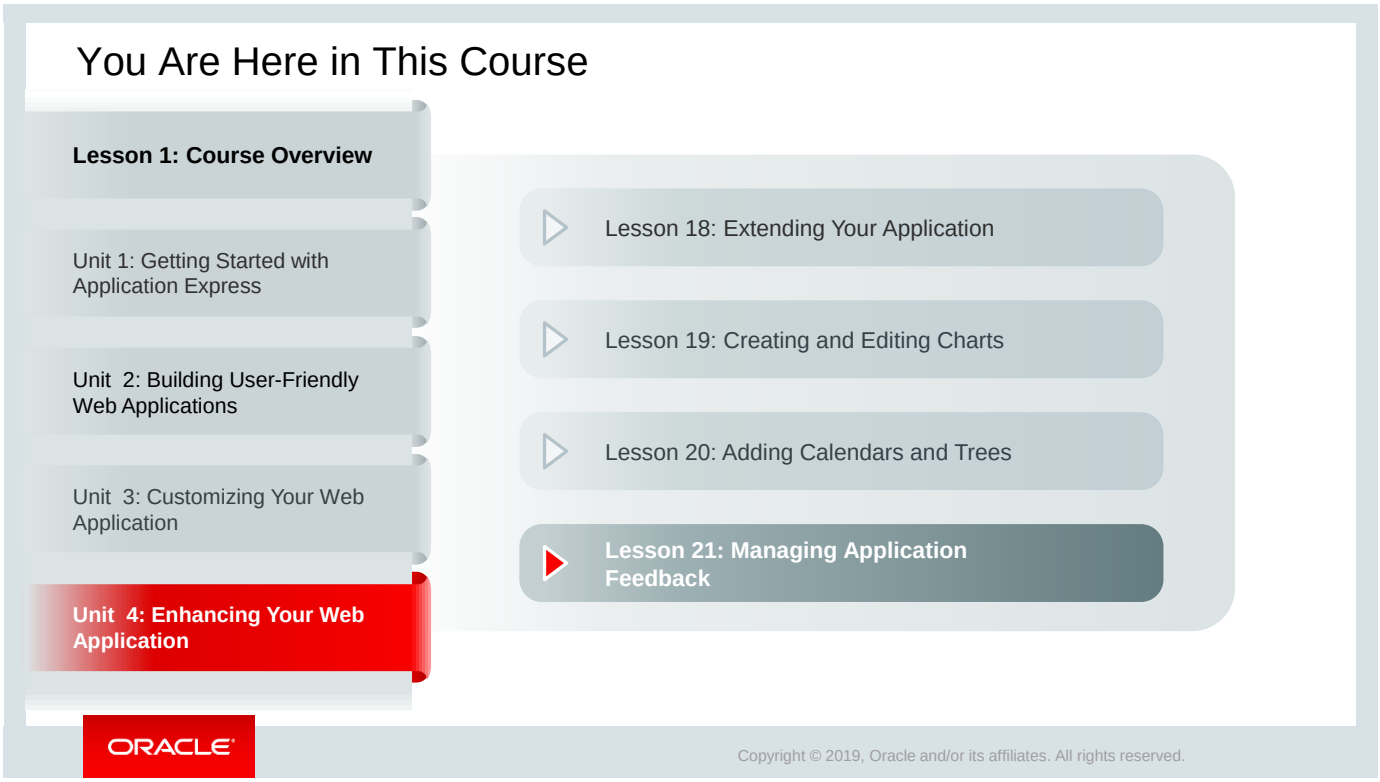
The lesson showed you how to use dynamic queries to display information in a calendar or tree.

21

Managing Application Feedback

Steve successfully created a complete project management tool called PTS by leveraging all the features provided by Oracle Application Express. PTS covers all aspects of project management requirements. Steve developed the PTS application by keeping a simple yet user-friendly design in mind so that users find it to be effective and useful.

Stella congratulates Steve on this great achievement and asks him to add a feedback form into PTS so that the PTS users can provide their appreciation or comments, if any, directly into the application. Steve finds this thought really useful and begins working on it. He feels that user feedback can serve as inputs for future enhancements to the application as well. Steve considers exploring the *Team Development* component of Oracle Application Express to develop the feedback module in PTS.



This slide shows a graphical representation of the entire course, highlighting lesson 21 in particular, which is dealt with in these slides.

Objectives

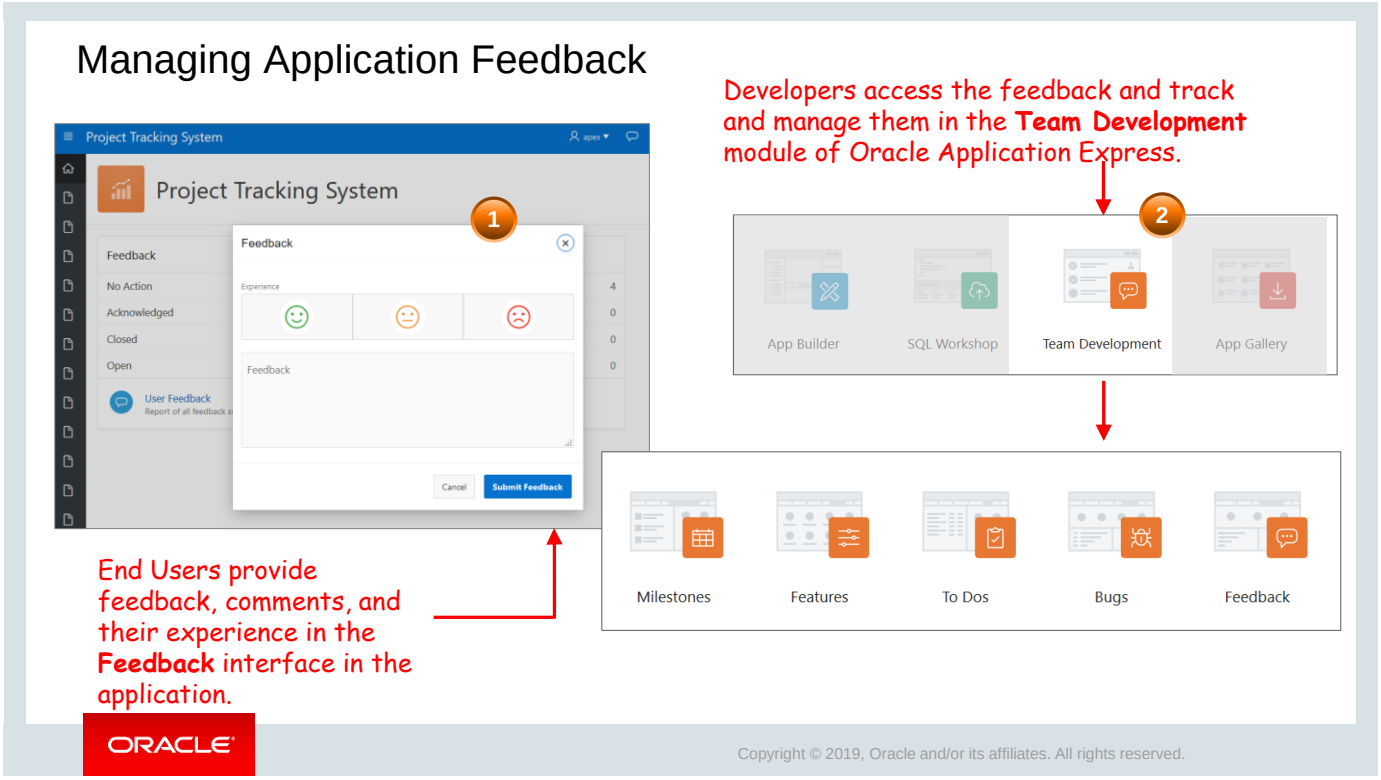
After completing this lesson, you should be able to:

- Describe what is Team Development
- Create a feedback page in your application
- Manage feedback

This lesson explains how to use the **Team Development** component of Oracle Application Express. You learn to track features, milestones, bugs, and to dos. You also learn to manage the feedback received.

Lesson Agenda

- Managing Feedback
- Understanding Team Development
- Reviewing the Progress of Your Milestones and Features
- Enabling Feedback in your application



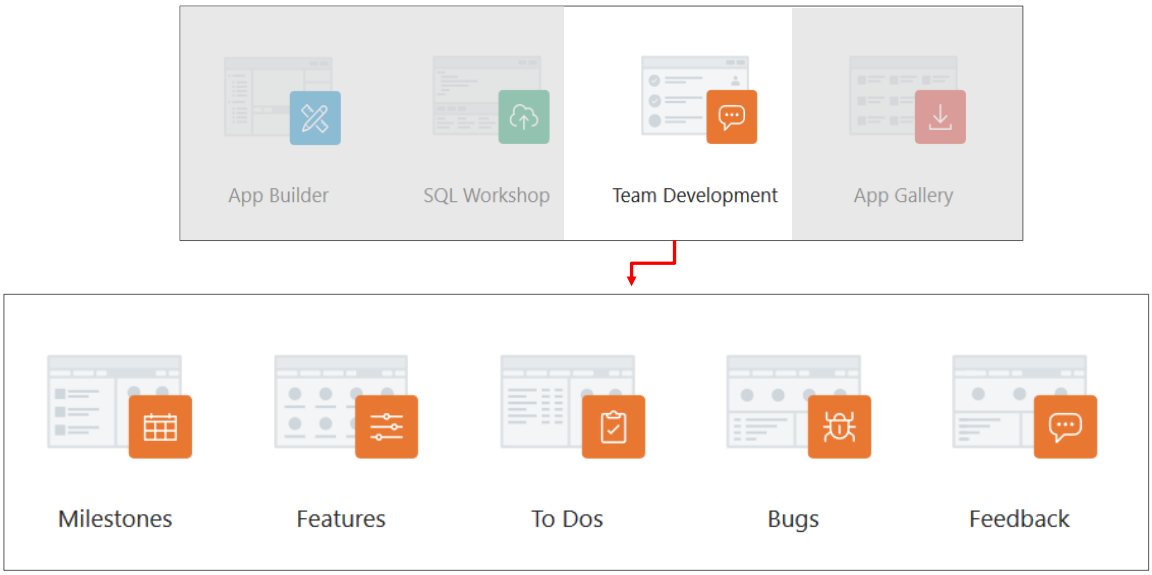
Managing feedback is a two-way mechanism – first, the end users provide their feedback through the front end or the application UI. Steve adds a feedback page to the PTS application, as depicted in screenshot 1. The second mechanism is to track and manage feedback that the developer does using the Team Development module in Oracle Application Express, depicted by screenshot 2. The developer, after analyzing the feedback, categorizes them into features, bugs, or to dos and tracks and manages them by assigning milestones, assigning developers to the bugs or to dos.

In this lesson, you will learn about the Team Development module of Oracle Application Express, how to track and manage feedback, and also how to create a feedback page in your application.

Lesson Agenda

- Managing Feedback
- Understanding Team Development
  - Creating and Updating Features
  - Creating and Updating Milestones
  - Creating Bugs
  - Creating and Updating To Dos
- Reviewing the Progress of Your Milestones and Features
- Enabling Feedback in your application

What Is Team Development?



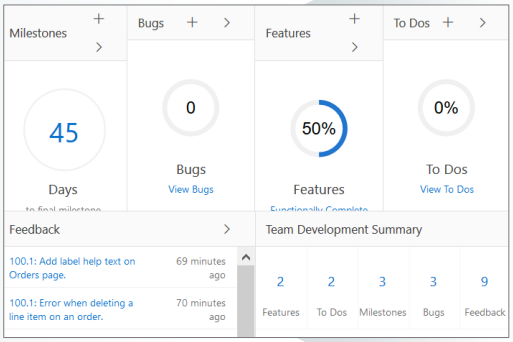
Team Development is a built-in development management module that enables you to manage the application development process by tracking new features, non-feature-related tasks, or to dos, bugs, and milestones. The end users provide real-time feedback, which can then be categorized into to dos, bugs, or features.

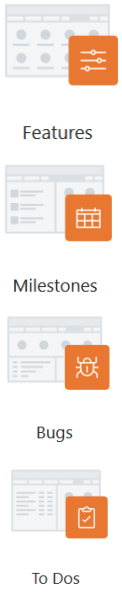
The Workspace Administrator has the privilege to access Team Development by default.

When creating a developer or a user, you have to set the Team Development module access to **Yes** or **No.**

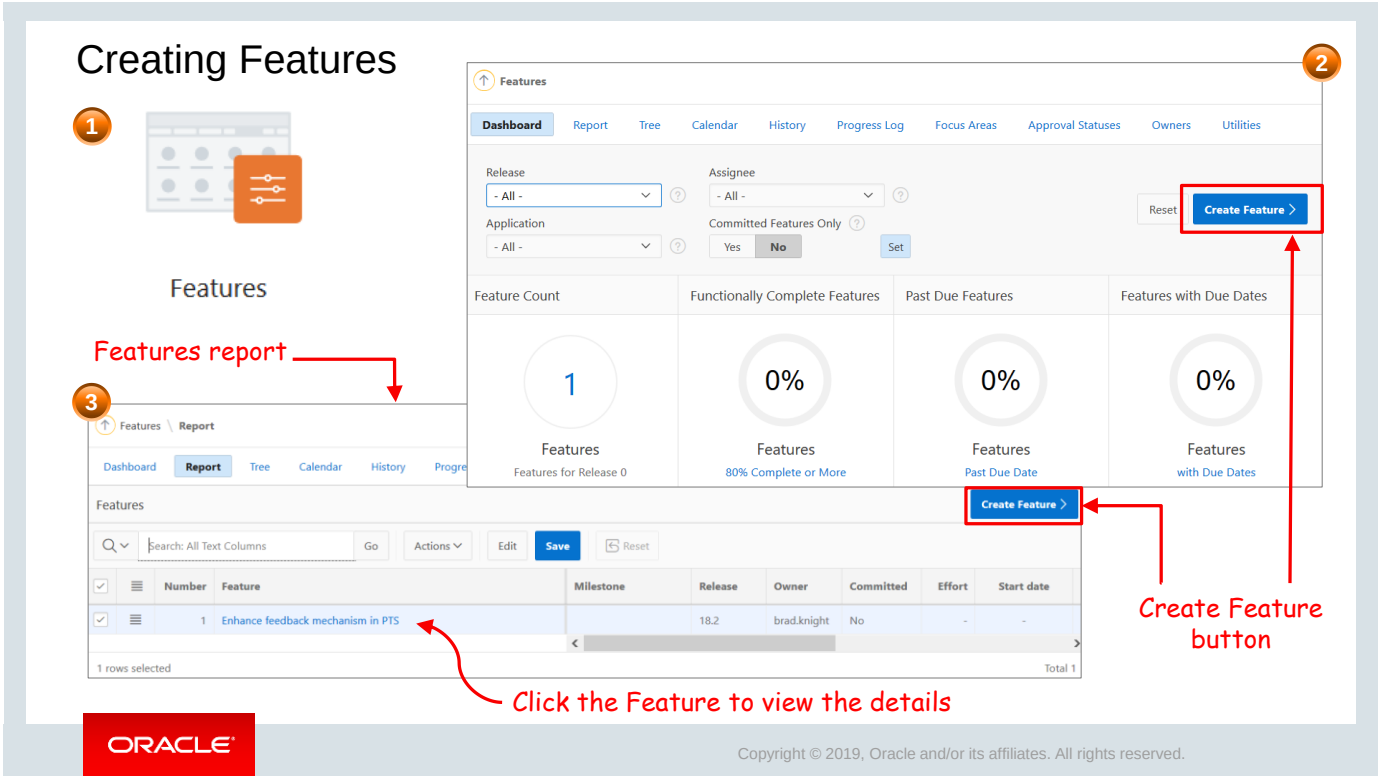
Tracking the Progress of Your Application Development Project

- Create and update features.
- Create and update milestones.
- Create bugs.
- Create and update to dos.
- Review the progress of your milestones and features (dashboards).





The slide lists the tasks that you perform to track the progress of your application. The tasks need not be performed in the order in which they are listed in the slide. However, the order used in the slide is the logical flow of when to do the tasks.



Let's quickly familiarize with the components of the Team Development module and how to create features, milestones, bugs, and to dos.

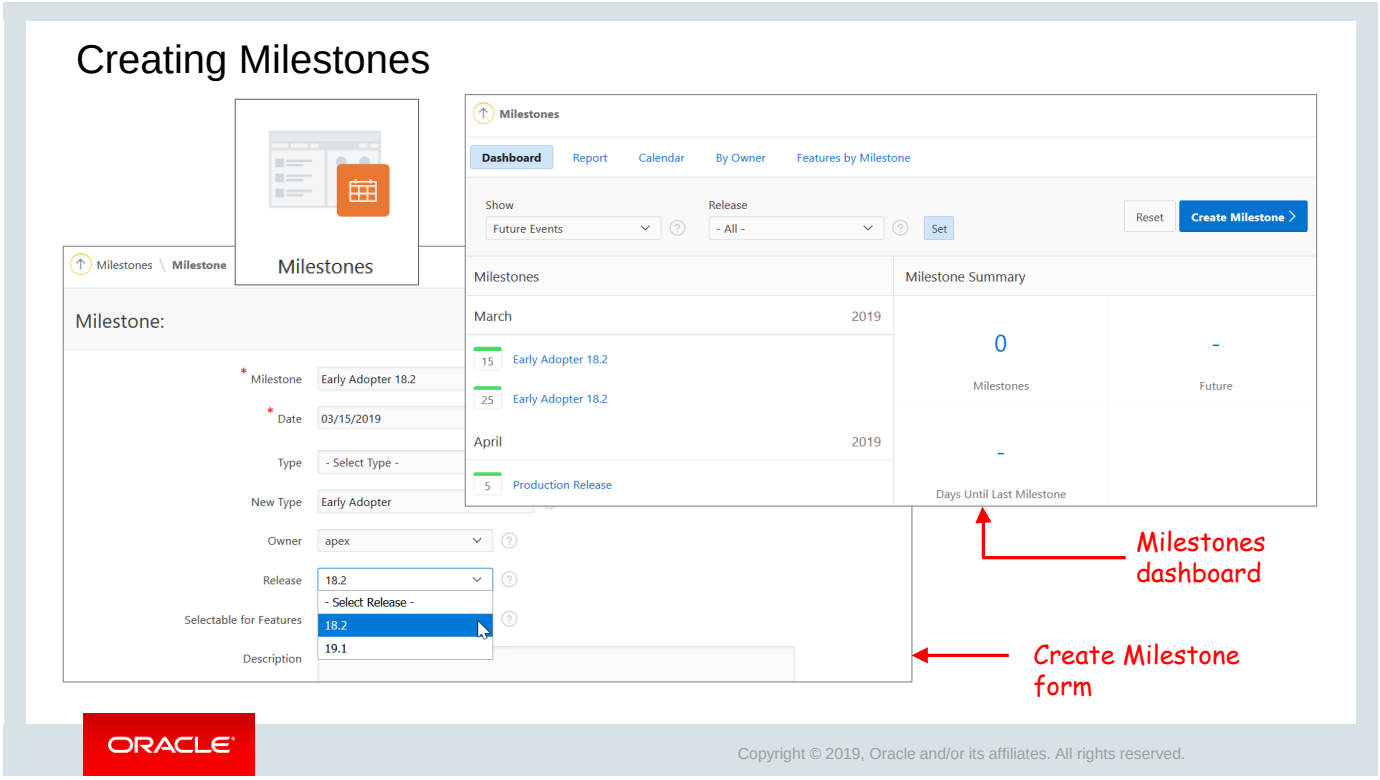
The **Features** page to track the features from initial concept through implementation. The end user can organize features by release, assignee, tags, or associated milestones.

Click the Features icon (screenshot 1) to view the features created. You can see various badges with the count of features based on various categorizations in the Features Dashboard (Screenshot 2) such as:

- Feature Count
- Functionally Complete Features
- Past Due Features
- Features with Due Dates
- Features by Status (Open, Completed)
- Feature Owners
- Without Owners

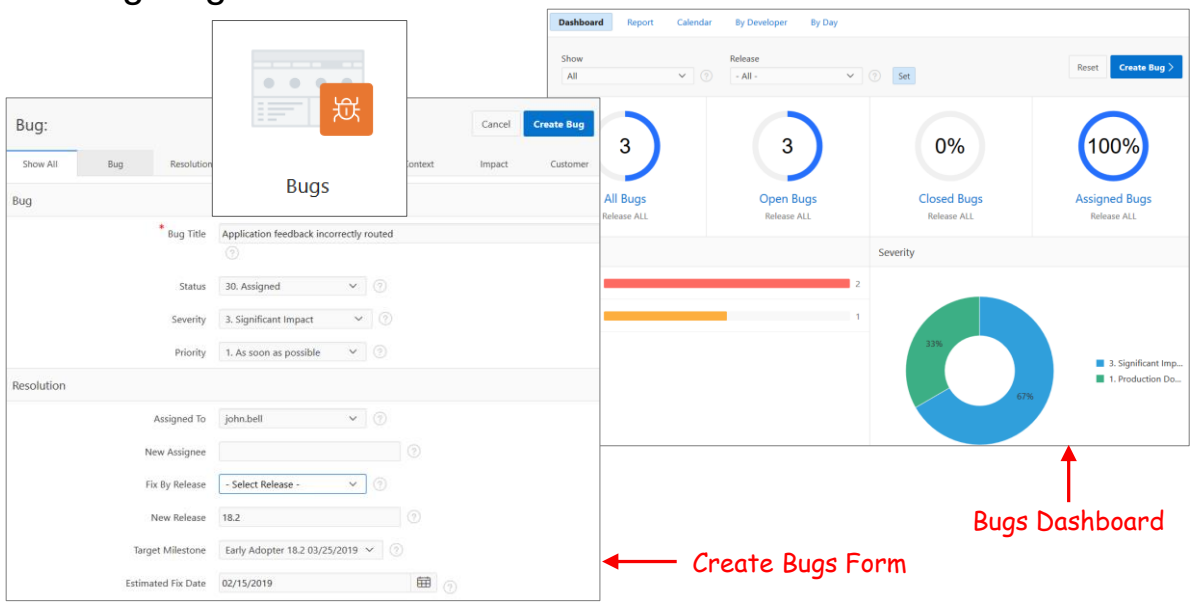
To view details of the features, click any of the badges that are displayed. In this view, additional information about each feature that you have created and its progress is available. There are several tabs that you can select for additional information, such as Report (screenshot 3), Calendar, which shows a calendar and the date on which the task is due and so on.

You can create a feature by clicking **Create Feature** in the Dashboard or Report and filling out the feature details.



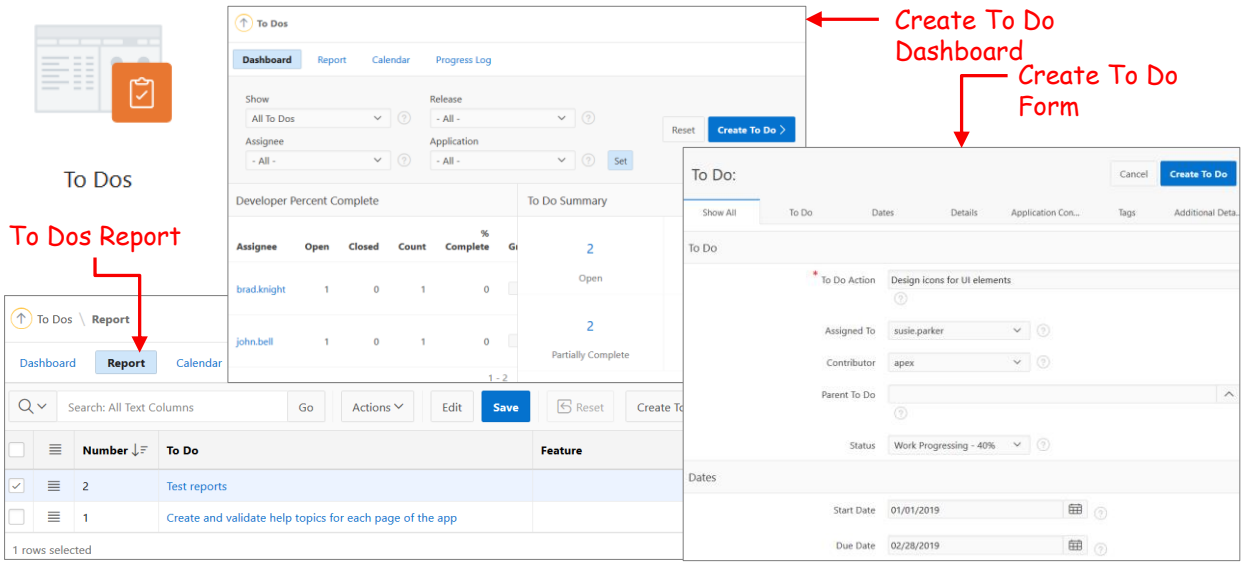
The **Milestones** page allows you to manage important milestones. Milestones track events. You can associate milestones with features, bugs, and to dos. In the example in the slide, you see milestones for the phases of the development life cycle: Early Adopter and Production. You can track how many features, to dos, and bugs are associated with each milestone. Other tabs provide additional information, such as features by milestone, which displays the features that have been assigned to a milestone. It is a good practice to organize milestones by release.

Creating Bugs



Bugs track software defects. Bugs can be assigned; associated with milestones; and tracked by due date, status, and other attributes. Tabs provide additional information, such as viewing all the bugs assigned to a particular developer or bugs opened and closed on a particular day.

Creating To Dos



To dos are action items that can be assigned, prioritized, tagged, and tracked. To dos can also have related parent tasks. To dos may or may not be associated with a feature or milestone. Tabs provide you additional information, such as a view of a to do progress log.

Clicking a assignee name will give a report of all the to dos assigned to him/her. To view the “to dos” assigned to yourself, you can click the **My To Dos** button in the report.

Quiz

Which Team Development component would you create to add feedback to application?

a. Feature
b. To do
c. Milestone
d. Bug

Answer: a

Quiz

Which Team Development component would you create to allow an employee to enter status report information?

a. Feature
b. To do
c. Milestone
d. Bug

Answer: a

Quiz

Which Team Development component would you create to correct the packing list report error when using IE?

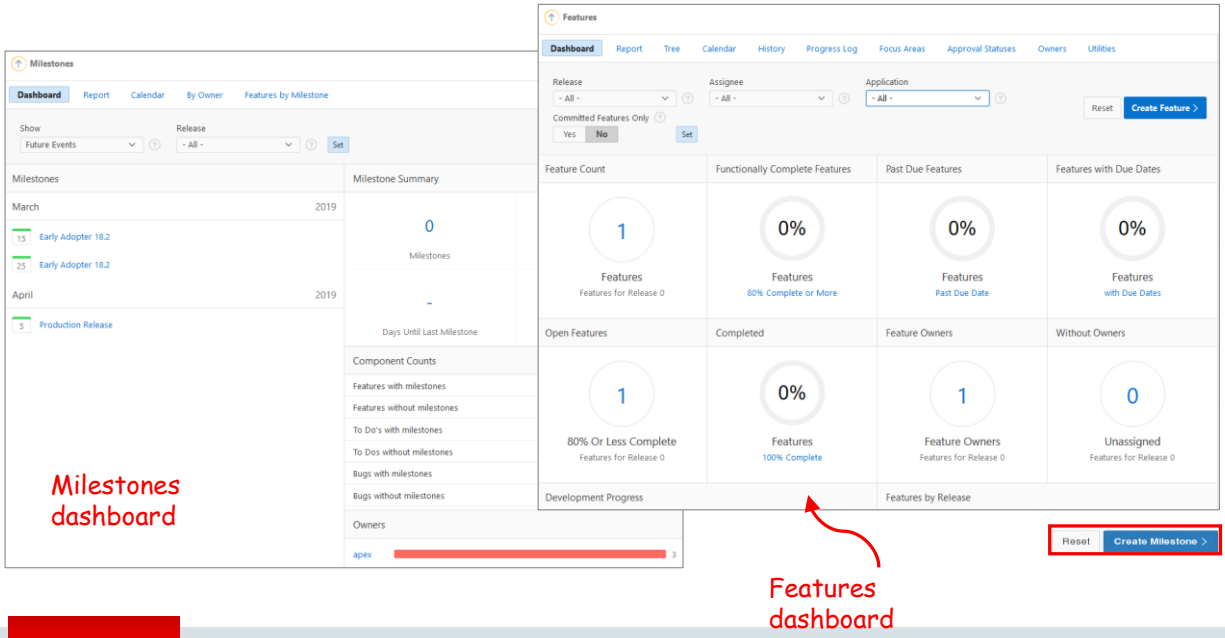
a. Feature
b. To do
c. Milestone
d. Bug

Answer: d

Lesson Agenda

- Managing Feedback
- Understanding Team Development
- Reviewing the Progress of Your Milestones and Features
- Enabling Feedback in your application

Review the Progress of Your Milestones and Features



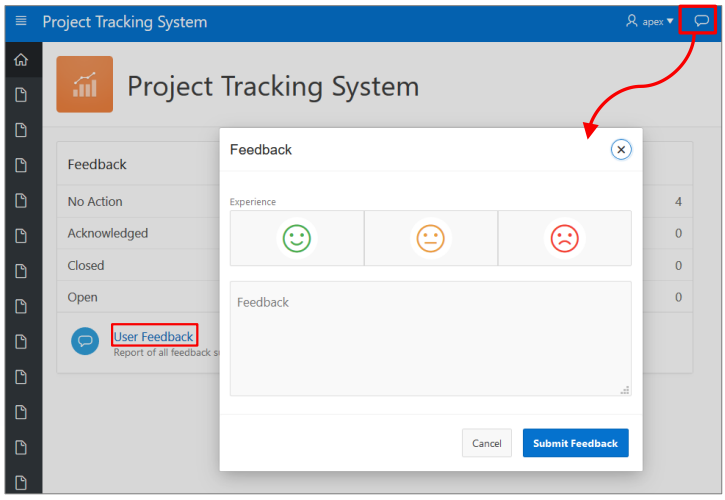
There is a dashboard for every Team Development component. In the example on the slide, the milestone dashboard and features dashboard are displayed. Milestones dashboard provides useful information, such as a summary of the upcoming milestones and the number of days that are left before the due date. Features dashboard provides information of all the features categorized under various badges such as feature count, functional completeness, past due features, open features, completed features, and features by owners.

Lesson Agenda

- Managing Feedback
- Understanding Team Development
- Reviewing the Progress of Your Milestones and Features
- Enabling Feedback in Your Application

Enabling Feedback in your Application

1. Enable feedback in application properties.
2. Create a feedback page.
3. End user submits feedback.
4. Access the submitted feedback in Team Development.



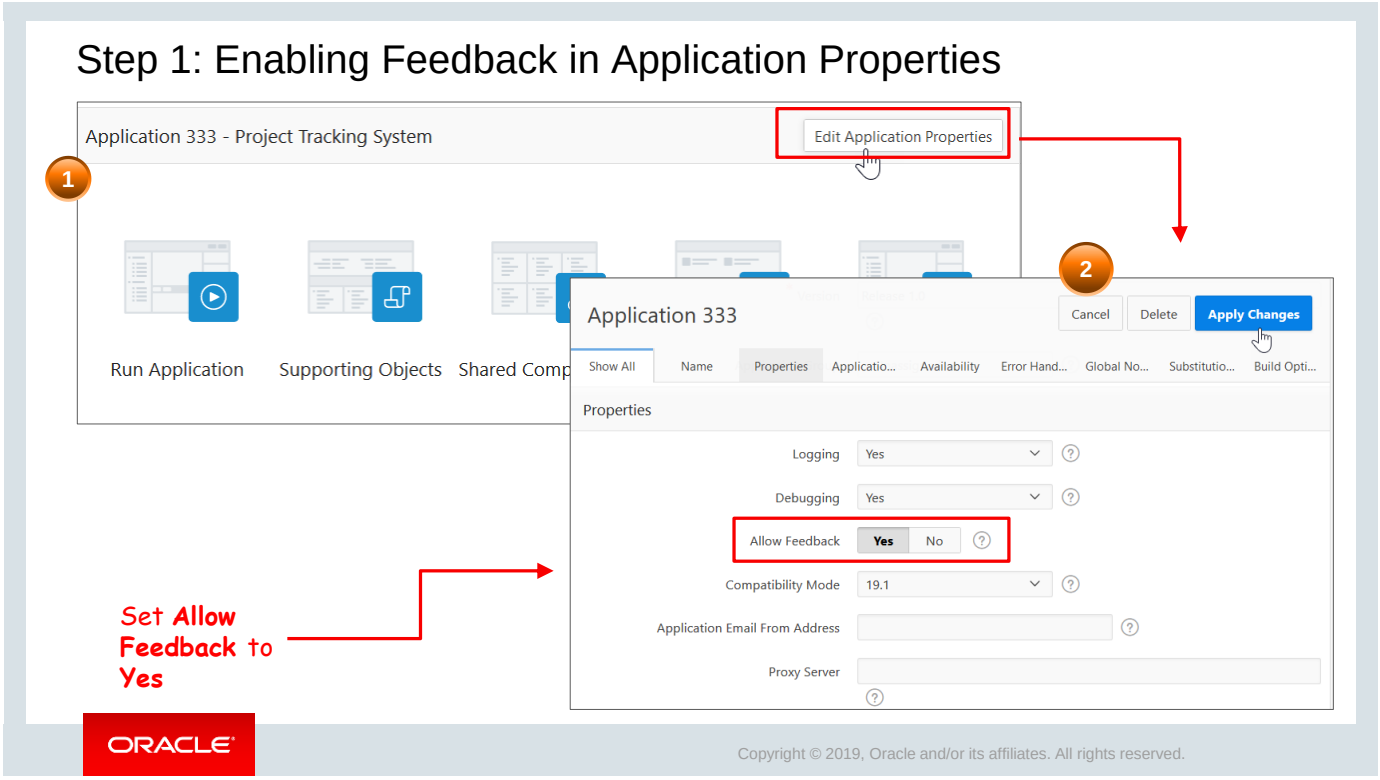
Steve adds a feedback page to the PTS application. Feedback is the process of providing and gathering real-time comments, enhancement requests, and bugs from your application users.

As you can see in the screenshot, the PTS home page now contains the **Feedback** icon at the top-right corner on the navigation bar. Clicking the icon opens the Feedback form, where the end user can indicate his experience, provide feedback, and submit it. The page also provides a count of the feedback in terms of feedbacks that are Open, Closed, Acknowledged, and those on which No Action is taken. The **User Feedback** link provides a list of all the feedbacks.

Enabling feedback in your application involves the following steps:

1. Enable feedback in application properties.
2. Create a feedback page.
3. Submit feedback.
4. Access the submitted feedback in Team Development.

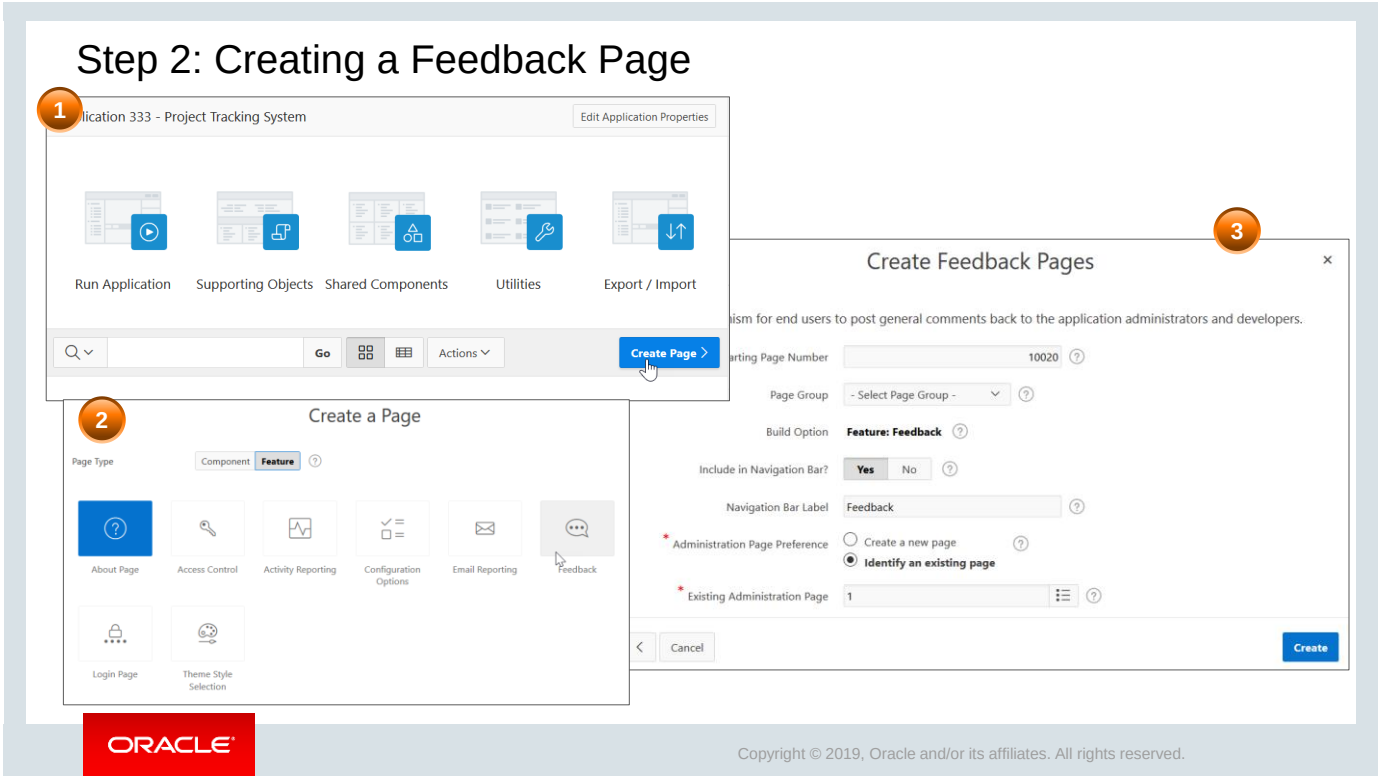
To add a feedback page to an application, perform the steps listed in the following slides.



As the first step to enable feedback, Steve sets the **Allow Feedback** option in Application Properties to **Yes.**

1. Go to the PTS application home page and click **Edit Application Properties.**
2. In the Edit Application Properties page, under Properties, click **Yes** for **Allow Feedback.**

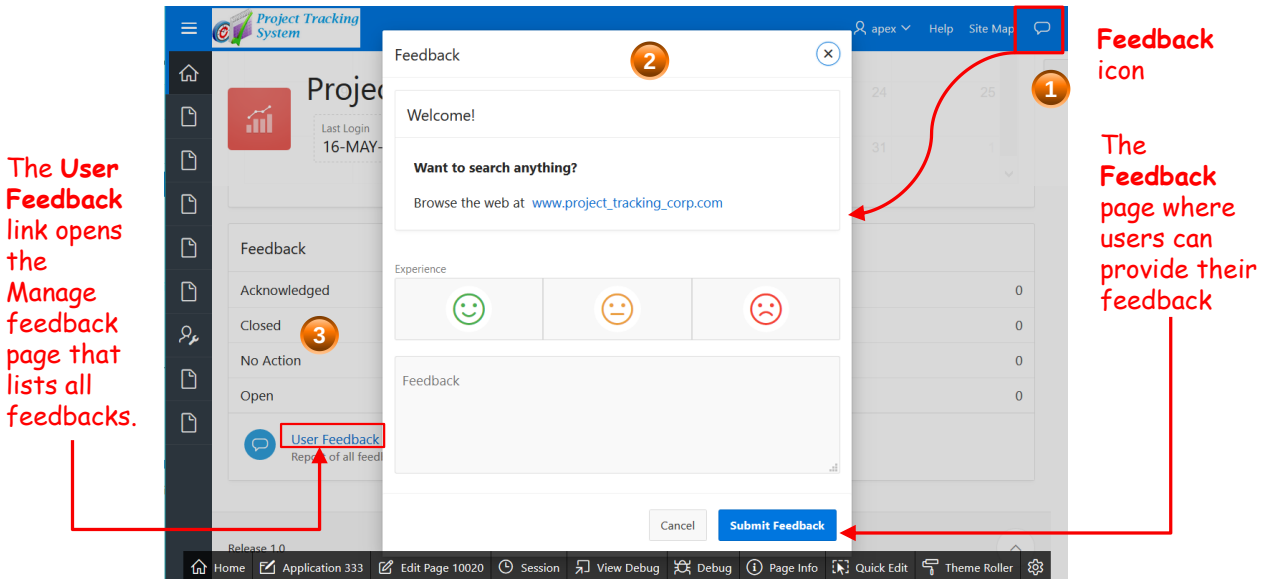
This completes the first step toward enabling feedback in your application.



Next, Steve creates a feedback page that will be displayed when the end user clicks the **Feedback** link on the navigation bar in the PTS application. To create a feedback page:

1. Go to the PTS application home page and click **Create Page** (screenshot 1).
2. In the Create a Page dialog, for Page Type, select **Feature,** click Feedback, and click **Next** (screenshot 2).
3. In the Create Feedback Pages, define the following (screenshot 3).
   - **Include in Navigation Bar?:** Click **Yes.**
   - **Navigation Bar Label:** Enter **Feedback**
   - **Administration Page Preference:** Click **Identify an existing page**
   - **Existing Administration Page:** Enter **1**
4. Click **Create.** The page opens in Page Designer.
5. Click **Save** and **Run.**

Step 3: Submitting Feedback

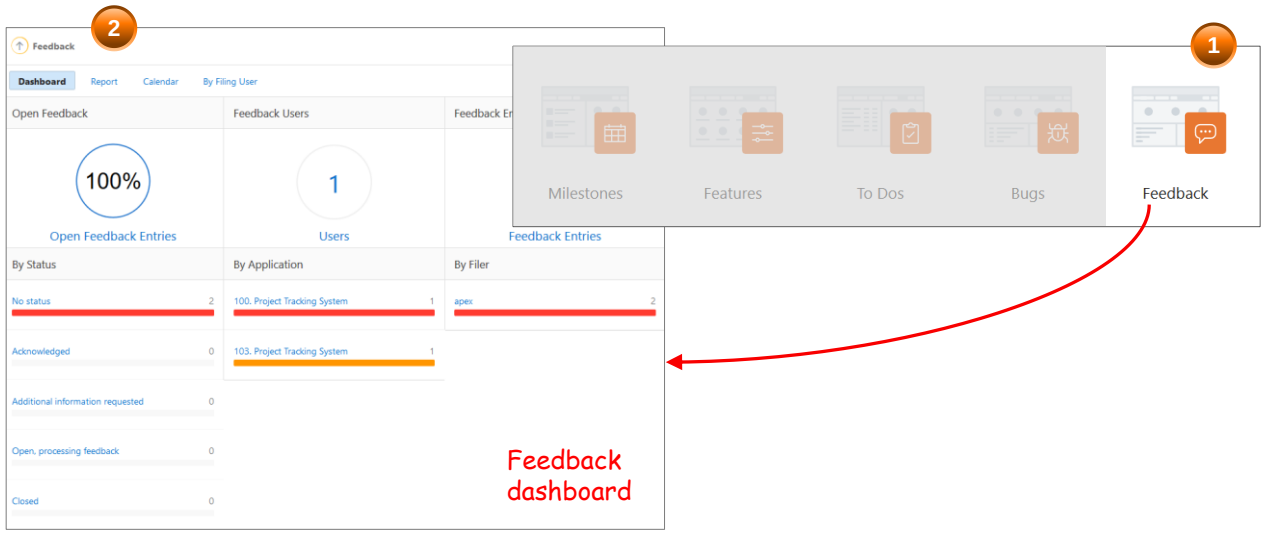


After you run the page, the PTS application opens. The navigation bar now contains the **Feedback** icon. Clicking the Navigation icon, indicated by 1 in the screenshot, opens the Feedback page (indicated by 2).

Enter your feedback in the Feedback text area, click any one of the smileys as applicable to indicate your experience, and click **Submit Feedback.**

The submitted feedback can be accessed on the Feedback page in Team Development and also in the Manage Feedback page that opens when you click the User Feedback link on the application page, as indicated by 3 in the screenshot. We will go into details on accessing submitted feedback in the next two slides.

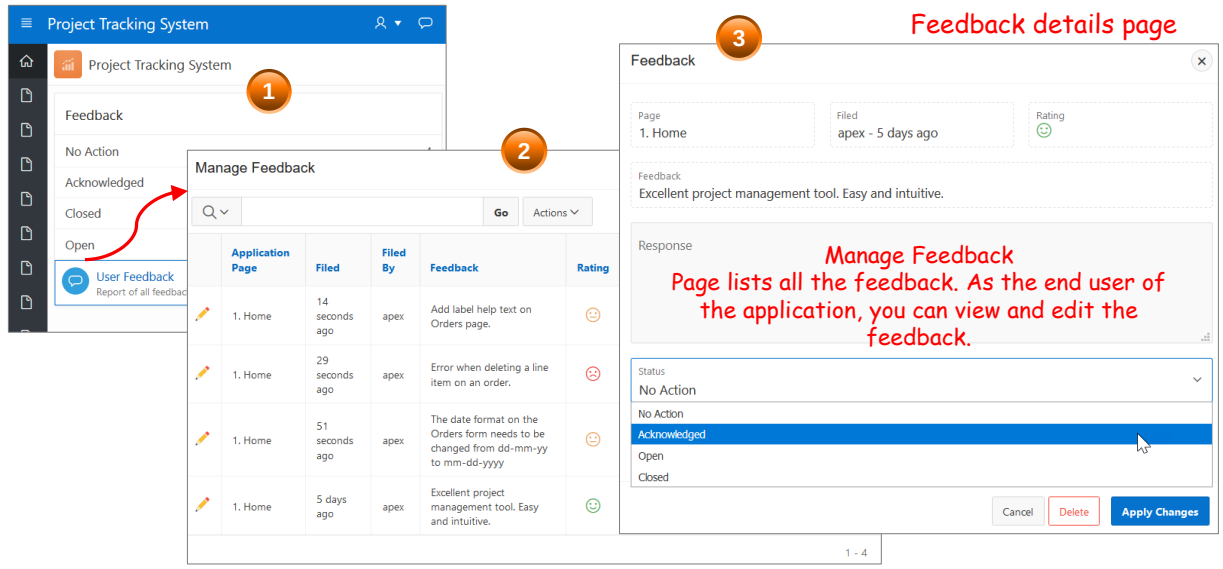
Step 4: Accessing Feedback in Team Development



When your feedback has been submitted, you as a developer can access it in the **Feedback** component of **Team Development**, as shown in screenshot 1. Click **Feedback**, and you can view the feedback listed in the Feedback Dashboard, as shown in screenshot 2. The dashboard lists the feedback grouped by status, application, and filter for all the applications in your workspace, to which you have access as a developer.

You can click each feedback and edit it to change the type to a bug, to do, or feature and assign it to someone. This is covered in Practice 21-1 in the corresponding activity guide for this lesson.

Step 4.1: Accessing Feedback as an End User



On the other hand, you as an end user can access the feedback by clicking the **User Feedback** link in the feedback page in your application, as shown in screenshot 1.

Clicking the **User Feedback** link opens the Manage Feedback page that lists all the feedback specific to your application. This is depicted in Screenshot 2. The Manage Feedback page also contains the edit icon, by clicking which you can view the feedback details, as shown in screenshot 3. You can edit your feedback, provide additional information for the developer, and even change the status to Closed, Acknowledged, or No Action.

Note: In the Manage Feedback page, you can only edit or modify the feedback as an *end user* and change the status to Open, Closed, Acknowledged, or No Action. The user with *Developer* role can mark the feedbacks as bugs, to dos, or enhancements and assign milestones to it by accessing the feedback from **Team Development.**

Quiz

Feedback is enabled for an application automatically.

a. True
b. False

Answer: b

Quiz

You can mark a Feedback as a Feature and assign it to a developer in the Manage Feedback page.

a. True

b. False

Answer: b

Practice 21 Overview: Adding and Monitoring Feedback in Your Application

This practice covers the following topics:

- Creating a feedback form
- Reviewing and editing the feedback

Summary

In this lesson, you should have learned how to:

- Track Team Development components related to your application development process
- Add feedback capabilities to your application
- Manage feedback

In this lesson, you learned how to track Team Development components.